



POD-(H)DG Method for Incompressible Flow Simulations

Guosheng Fu¹ · Zhu Wang²

Received: 17 April 2020 / Revised: 12 August 2020 / Accepted: 30 September 2020 /

Published online: 18 October 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

We present a reduced order method based on proper orthogonal decomposition for the viscous Burgers' equation and the incompressible Navier–Stokes equations discretized using an implicit-explicit hybrid discontinuous Galerkin/discontinuous Galerkin (IMEX HDG/DG) scheme. A novel closure model, which can be easily computed offline, is introduced. Numerical results are presented to test the proposed POD model and the closure model.

Keywords HDG · DG · POD · Burgers' equation · Navier–Stokes equations

Mathematics Subject Classification 65N30 · 65N12 · 76S05 · 76D07

1 Introduction

Reduced order modeling has been widely used in flow control and optimization problems to alleviate the huge computational cost needed in many-query solutions of the large-scale dynamical systems associated to these problems [2, 13, 14, 19]. To achieve the high computational efficiency, model reduction methods construct from data a numerical surrogate model with the dimension greatly reduced from the original system. To build such a low-dimensional model, one can use non-intrusive approaches such as operator learning [3, 18], or intrusive approaches such as projection-based methods [7]. The method to be used in this paper falls into the second category. In particular, we consider the proper orthogonal decomposition (POD) method—one of the most popular snapshot-based model reduction techniques. The

Guosheng Fu gratefully acknowledges the partial support of this work from U.S. National Science Foundation through Grant DMS-2012031. Zhu Wang gratefully acknowledges the partial support of this work from U.S. National Science Foundation through Grant DMS-1913073 and Office of the Vice President for Research at the University of South Carolina through an ASPIRE grant.

✉ Guosheng Fu
gfu@nd.edu

Zhu Wang
wangzhu@math.sc.edu

¹ Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN, USA

² Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA

general POD model reduction methodology splits the overall calculation into offline and online stages. At the offline stage, a handful of reduced basis vectors are determined and a low-dimensional, reduced order model (ROM) is constructed by learning algorithms or by projecting equations to the space spanned by the reduced basis. At the online stage, the ROM is used alternative to the original system for simulations that can be finished in short time or even real time. When the system contains non-polynomial nonlinearities, hyper-reduction has to be used in order to guarantee the online computational complexity to be independent of the dimension of the original system [9,10].

The ROM can be discretized by any conventional numerical method. In particular, when continuous Galerkin finite elements are used, each nodal value will be shared by several elements. If an interpolation type of hyper-reduction methods is applied, such as discrete empirical interpolation method (DEIM) or its variants [10], although the nonlinear functions need only to evaluate at few selected points, many elements that share these nodes have to be looped. This would cause expensive online computations. Thus, the finite element with interpolated coefficients method was developed in [26], in which the nonlinear functions in the ROM are replaced with their finite element interpolants so that the DEIM can be applied directly on the finite element coefficients. However, if a discontinuous Galerkin (DG) method is applied, there is no such issue thanks to the local nature of the DG method. There has been several work that uses POD in the context of DG. In [23] hybridizable discontinuous Galerkin (HDG) POD model has been developed for heat equation. It is shown that highly accurate flux approximation can be recovered in the HDG-POD approximation at a low cost. In [25], POD is applied in the context of symmetric interior penalty DG for solving Allen-Cahn equation. For parametric problems, DG has also been applied together with reduced basis method for elliptic problems [1] and with empirical quadrature procedure for nonlinear conservation laws in [27].

As a first step for investigating reduced order modeling on flow control and optimization applications, we focus on the computational fluid dynamics of incompressible fluid flows in this work. When the POD approximation is sought for such problems, there are two common ways to deal with the incompressibility constraint. One only keeps velocity in the reduced system, which is based on the argument that the POD basis is weakly divergence-free since it is a combination of snapshots and snapshots are weakly divergence-free, thus the pressure term would vanish after projection; the other keeps both velocity and pressure in the reduced system, because either the application at hand needs pressure information or numerical methods for computing snapshots may not provide pointwise divergence free flow fields. Indeed, the discretely divergence-free property does not hold for many popular discretization of the Navier–Stokes equations. A new velocity and pressure ROM is proposed in [5] while introducing a supremizer stabilization to fulfill an equivalent inf-sup condition. In [8], one velocity ROM and two velocity-pressure ROMs are compared that shows the accuracy of snapshots does have a big impact on the performance of velocity ROM. Therefore, in this work, we use the divergence-free HDG method developed in [15] for the full order model (FOM), which ensures the velocity snapshots are exactly pointwise divergence free. As a consequence, we can use the velocity ROM since the POD basis generated from these snapshots would have the same divergence-free property. Furthermore, because the convective term involves upwinding numerical flux in the FOM that can not be precomputed offline, we replace it with an (offline-computable) linear central flux in the ROM. We then add a *linear* convective stabilization term in the ROM which mimics the upwinding stabilization in the FOM. This yields an efficient implementation while keeping stabilization effects of the numerical flux. However, the introduced stabilization might not be enough for convection-dominated problems, because the jumps across elements of the POD basis are small when

snapshots are obtained from high fidelity simulations. Therefore, we include extra dissipation following the closure model developed in [20] to diminish the numerical oscillations.

The rest of paper is organized as follows. In Sect. 2, the full order model is presented. In Sect. 3, the POD reduced order model is derived. Several numerical experiments including the Burgers' equation, the Navier–Stokes equations and the incompressible Euler equations are discussed in Sect. 4. We conclude in Sect. 5 with some future work.

2 Full Order Model via IMEX HDG/DG

Next, we first describe the FOM that is employed for generating snapshots and provides the benchmark solutions in our numerical experiments.

2.1 Notation

Let \mathcal{T}_h be a conforming simplicial triangulation of the domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$. For any element $K \in \mathcal{T}_h$, we denote by h_K its diameter and by $h : \mathcal{T}_h \rightarrow \mathbb{R}$ the mesh size function with $h|_K = h_K$. The collection of element boundaries is $\partial\mathcal{T}_h := \{\partial K : K \in \mathcal{T}_h\}$. Denote by \mathcal{E}_h the set of facets of \mathcal{T}_h (vertices in 1D, edges in 2D, faces in 3D), and by $\mathcal{E}_h^i = \mathcal{E}_h \setminus \partial\Omega$ the set of interior facets. For any element K , denote $\mathbf{n}_K : \partial K \rightarrow \mathbb{R}^d$ to be the unit outward normal direction on ∂K from the element K . Let $\mathbf{n} : \partial\mathcal{T}_h \rightarrow \mathbb{R}^d$ be the unit normal direction on the collection of element boundaries $\partial\mathcal{T}_h$ with $\mathbf{n}|_{\partial K} = \mathbf{n}_K$.

We collect the following set of finite element spaces:

$$\mathbf{V}_h^k := \{\mathbf{v} \in H(\text{div}, \Omega) : \mathbf{v}|_K \in [\mathbb{P}^k(K)]^d, \forall K \in \mathcal{T}_h, (\mathbf{v} \cdot \mathbf{n})|_{\partial\Omega} = 0\}, \quad (1a)$$

$$\widehat{\mathbf{V}}_h^k := \{\widehat{\mathbf{v}} \in [L^2(\mathcal{E}_h)]^d : \widehat{\mathbf{v}}|_F \in [\mathbb{P}^k(F)]^d, (\widehat{\mathbf{v}} \cdot \mathbf{n})|_F = 0, \forall F \in \mathcal{E}_h, \widehat{\mathbf{v}}|_{\partial\Omega} = 0\}, \quad (1b)$$

$$\mathbf{W}_h^k := \{w \in L^2(\Omega) : w|_T \in \mathbb{P}^k(K), \forall K \in \mathcal{T}_h\}, \quad (1c)$$

$$\widehat{\mathbf{W}}_h^k := \{\widehat{w} \in L^2(\mathcal{E}_h) : \widehat{w}|_F \in \mathbb{P}^k(F), \widehat{w}|_{\partial\Omega} = 0\}, \quad (1d)$$

where \mathbb{P}^k is the space of polynomials up to degree $k \geq 0$. In 1D, $\mathbb{P}^k(F)$ is simply point evaluation for the vertex F . Note that functions in $\widehat{\mathbf{V}}_h^k$ and $\widehat{\mathbf{W}}_h^k$ are defined only on the mesh skeleton \mathcal{E}_h .

2.2 The Model Problems

Two mathematical models are considered in this work, namely the 1D viscous Burgers' equation

$$\frac{\partial u}{\partial t} + u \cdot \nabla u - \nu \Delta u = 0 \quad \text{in } \Omega \subset \mathbb{R}, \quad (2)$$

and the 2D incompressible Navier–Stokes equations (3):

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nu \Delta \mathbf{u} = 0 \quad \text{in } \Omega \subset \mathbb{R}^2, \quad (3a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad (3b)$$

where $\nu > 0$ is a positive viscosity parameter. It becomes the incompressible Euler equations when $\nu = 0$. For simplicity of presentation, we use homogeneous Dirichlet boundary

conditions for both problems to derive the FOM IMEX HDG/DG schemes. Other standard boundary conditions will be applied in the numerical experiments presented in Sect. 4.

We remark that the proposed scheme for 2D incompressible Navier–Stokes equations can be directly applied to the 3D case, although we focus on the former in this paper. Detailed numerical study of our POD-(H)DG scheme for 3D incompressible Navier–Stokes is a topic of our on-going work.

2.3 The Semidiscrete HDG/DG Scheme: Burgers' Equation

The semidiscrete HDG/DG scheme for the 1D Burgers' equation (2) reads as follows: Given initial data $u_h(0) \in W_h^k$, for all $t \in (0, T]$, find $(u_h, \hat{u}_h) = (u_h(t), \hat{u}_h(t)) \in W_h^k \times \hat{W}_h^0$ such that

$$\mathcal{M}_h \left(\frac{\partial u_h}{\partial t}, v \right) + \mathcal{C}_h^{dg}(u_h, u_h, v) + v \mathcal{B}_h^{hdg}((u_h, \hat{u}_h), (v, \hat{v})) = 0, \quad \forall (v, \hat{v}) \in W_h^k \times \hat{W}_h^0. \quad (4)$$

Here $\mathcal{M}_h(\cdot, \cdot)$ is the mass operator, $\mathcal{C}_h^{dg}(\cdot, \cdot, \cdot)$ is the nonlinear (DG) convection operator, and $\mathcal{B}_h^{hdg}(\cdot, \cdot)$ is the (HDG) diffusion operator, which are given as follows:

$$\mathcal{M}_h(u, v) = \sum_{K \in \mathcal{T}_h} \int_K u v \, dx, \quad (5a)$$

$$\mathcal{C}_h^{dg}(w, u, v) = -\frac{1}{2} \sum_{K \in \mathcal{T}_h} \left(\int_K (wu) \cdot \nabla v \, dx - \int_{\partial K} \{w\} u^- \cdot \mathbf{n} v \, ds \right), \quad (5b)$$

$$\begin{aligned} \mathcal{B}_h^{hdg}((u, \hat{u}), (v, \hat{v})) &= \sum_{K \in \mathcal{T}_h} \left(\int_K \nabla u \cdot \nabla v \, dx - \int_{\partial K} \nabla u \cdot \mathbf{n} (v - \hat{v}) \, ds \right. \\ &\quad \left. - \int_{\partial K} \nabla v \cdot \mathbf{n} (u - \hat{u}) \, ds + \int_{\partial K} \frac{4k^2}{h} (u - \hat{u})(v - \hat{v}) \, ds \right), \end{aligned} \quad (5c)$$

where $\{w\}$ in (5b) is the standard average operator on element boundaries, and u^- is the upwinding numerical flux, with $u^-|_F = (u|_{K^-})|_F$ for any facet F shared by two elements K^\pm , and K^- is the element such that $(\{w\} \cdot \mathbf{n}_{K^-})|_F \geq 0$.

2.4 The Semidiscrete HDG/DG Scheme: Navier–Stokes Equations

The divergence-free HDG/DG scheme in [15, 16] is used for the Navier–Stokes equations (3). The semidiscrete scheme reads as follows: Given initial data $\mathbf{u}_h(0) \in \mathbf{V}_h^k$, for all $t \in (0, T]$, find $(\mathbf{u}_h, \hat{\mathbf{u}}_h, p_h) = (\mathbf{u}_h(t), \hat{\mathbf{u}}_h(t), p_h) \in \mathbf{V}_h^k \times \hat{\mathbf{V}}_h^k \times W_h^{k-1}$ such that

$$\begin{aligned} \mathcal{M}_h \left(\frac{\partial \mathbf{u}_h}{\partial t}, \mathbf{v} \right) + \mathcal{C}_h^{dg}(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}) + v \mathcal{B}_h^{hdg}((\mathbf{u}_h, \hat{\mathbf{u}}_h), (\mathbf{v}, \hat{\mathbf{v}})) - \mathcal{D}_h(\mathbf{u}_h, q) \\ - \mathcal{D}_h(\mathbf{v}, p_h) = 0, \end{aligned} \quad (6)$$

for all $(\mathbf{v}, \hat{\mathbf{v}}, q) \in \mathbf{V}_h^k \times \hat{\mathbf{V}}_h^k \times W_h^{k-1}$, where the operators are given as follows:

$$\mathcal{M}_h(\mathbf{u}, \mathbf{v}) = \sum_{K \in \mathcal{T}_h} \int_K \mathbf{u} \cdot \mathbf{v} \, dx, \quad (7a)$$

$$\mathcal{C}_h^{dg}(\mathbf{w}, \mathbf{u}, \mathbf{v}) = - \sum_{K \in \mathcal{T}_h} \left(\int_K (\mathbf{w} \otimes \mathbf{u}) : \nabla \mathbf{v} \, dx - \int_{\partial K} (\mathbf{w} \cdot \mathbf{n})(\mathbf{u}^- \cdot \mathbf{v}) \, ds \right), \quad (7b)$$

$$\mathcal{D}_h(\mathbf{u}, q) = \sum_{K \in \mathcal{T}_h} \int_K (\nabla \cdot \mathbf{u}) q \, dx, \quad (7c)$$

$$\begin{aligned} \mathcal{B}_h^{hdg}((\mathbf{u}, \widehat{\mathbf{u}}), (\mathbf{v}, \widehat{\mathbf{v}})) &= \sum_{K \in \mathcal{T}_h} \left(\int_K \nabla \mathbf{u} : \nabla \mathbf{v} \, dx - \int_{\partial K} (\nabla \mathbf{u} \mathbf{n}) \cdot \text{tang}(\mathbf{v} - \widehat{\mathbf{v}}) \, ds \right. \\ &\quad \left. - \int_{\partial K} (\nabla \mathbf{v} \mathbf{n}) \cdot \text{tang}(\mathbf{u} - \widehat{\mathbf{u}}) \, ds \right. \\ &\quad \left. + \int_{\partial K} \frac{4k^2}{h} \text{tang}(\mathbf{u} - \widehat{\mathbf{u}}) \cdot \text{tang}(\mathbf{v} - \widehat{\mathbf{v}}) \, ds \right), \end{aligned} \quad (7d)$$

where \mathbf{u}^- in (7b) is the *upwinding* numerical flux, with $\mathbf{u}^-|_F = (\mathbf{u}|_{K^-})|_F$ for any facet F shared by two elements K^\pm , and K^- is the element such that $(\mathbf{w} \cdot \mathbf{n}_{K^-})|_F \geq 0$, and $\text{tang}(\mathbf{v})|_F := \mathbf{v} - (\mathbf{v} \cdot \mathbf{n})\mathbf{n}$ is the tangential component of the vector \mathbf{v} . Notice that the convective operator (7b) introduces numerical dissipation along element boundaries:

$$\begin{aligned} \mathcal{C}_h^{dg}(\mathbf{u}_h, \mathbf{u}_h, \mathbf{u}_h) &= \frac{1}{2} \sum_{K \in \mathcal{T}_h} \int_{\partial K} |\mathbf{u}_h \cdot \mathbf{n}| (\llbracket \mathbf{u}_h \rrbracket)^2 \, ds \geq 0, \\ \forall \mathbf{u}_h &\in \mathbf{V}_h^k \cap H(\text{div}^0, \Omega), \end{aligned} \quad (8)$$

where

$$H(\text{div}^0, \Omega) := \{\mathbf{v} \in H(\text{div}, \Omega) : \nabla \cdot \mathbf{v} = 0\},$$

which is beneficial in the convection-dominated regime.

We remark that the scheme (6) produces an exactly divergence-free velocity approximation, i.e. $\mathbf{u}_h \in \mathbf{V}_h^{k,0} := \mathbf{V}_h^k \cap H(\text{div}^0, \Omega)$, which is a desired property for the POD model we consider in the next section. In particular, the velocity field $(\mathbf{u}_h, \widehat{\mathbf{u}}_h) \in \mathbf{V}_h^{k,0} \times \widehat{\mathbf{V}}_h^k$ can be directly computed without pressure approximation by solving the following equations:

$$\begin{aligned} \mathcal{M}_h \left(\frac{\partial \mathbf{u}_h}{\partial t}, \mathbf{v} \right) + \mathcal{C}_h^{dg}(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}) + \nu \mathcal{B}_h^{hdg}((\mathbf{u}_h, \widehat{\mathbf{u}}_h), (\mathbf{v}, \widehat{\mathbf{v}})) &= 0, \\ \forall (\mathbf{v}, \widehat{\mathbf{v}}) &\in \mathbf{V}_h^{k,0} \times \widehat{\mathbf{V}}_h^k. \end{aligned} \quad (9)$$

2.5 The Fully Discrete HDG Schemes

For the time discretization, we use the second-order Crank–Nicolson–Adams–Bashforth (CNAB) method [4], which treats the nonlinear convective term explicitly, and other terms implicitly. For simplicity, a uniform time partition is applied. Let $0 = t_0 < t_1 < \dots < t_M = T$ be the partition of the interval $[0, T]$ and the time step $\Delta t = \frac{T}{M}$.

The fully discrete scheme for the Burgers' equation (2) is given as follows: Given initial data $(u_h^0, \widehat{u}_h^0) \in W_h^k \times \widehat{W}_h^0$, for each integer $n = 1, \dots, M$, find $(u_h^n, \widehat{u}_h^n) \in W_h^k \times \widehat{W}_h^0$ such that

$$\begin{aligned} \mathcal{M}_h \left(\frac{u_h^n - u_h^{n-1}}{\Delta t}, v \right) + \mathcal{C}_h^{dg}(\widehat{u}_h^{n-1/2}, \widehat{u}_h^{n-1/2}, v) + \nu \mathcal{B}_h^{hdg}((u_h^{n-1/2}, \widehat{u}_h^{n-1/2}), \\ (\mathbf{v}, \widehat{\mathbf{v}})) &= 0, \end{aligned} \quad (10)$$

for all $(v, \widehat{v}) \in W_h^k \times \widehat{W}_h^0$, where

$$u_h^{n-1/2} := \frac{1}{2}(u_h^n + u_h^{n-1}), \quad \widehat{u}_h^{n-1/2} := \frac{1}{2}(\widehat{u}_h^n + \widehat{u}_h^{n-1}), \quad \widetilde{u}_h^{n-1/2} := \frac{3}{2}u_h^{n-1} - \frac{1}{2}u_h^{n-2}.$$

Here in the first step ($n = 1$) we simply take $\widetilde{u}_h^{1/2} = u_h^0$.

Similarly, the fully discrete scheme for the Navier–Stokes equations (3) is given as follows: Given initial data $(u_h^0, \widehat{u}_h^0) \in V_h^k \times \widehat{V}_h^k$, for each integer $n = 1, \dots, M$, find $(u_h^n, \widehat{u}_h^n, p_h^{n-1/2}) \in V_h^k \times \widehat{V}_h^k \times W_h^{k-1}$ such that

$$\begin{aligned} \mathcal{M}_h \left(\frac{u_h^n - u_h^{n-1}}{\Delta t}, v \right) + \mathcal{C}_h^{dg}(\widetilde{u}_h^{n-1/2}, \widehat{u}_h^{n-1/2}, v) + \nu \mathcal{B}_h^{hdg}((u_h^{n-1/2}, \widehat{u}_h^{n-1/2}), (v, \widehat{v})) \\ - \mathcal{D}_h(v, p_h^{n-1/2}) - \mathcal{D}_h(u_h^{n-1/2}, q) = 0, \end{aligned} \quad (11)$$

for all $(v, \widehat{v}, q) \in V_h^k \times \widehat{V}_h^k \times W_h^{k-1}$.

Efficient implementation of the HDG linear systems (10) and (11) via static condensation were discussed, for example, in [11, 15].

3 The POD Model

In this section, we present the POD model based on the FOM IMEX HDG/DG schemes presented in Sect. 2 using the method of snapshots [24]. We focus on the discussion for the Navier–Stokes equations as the results for Burgers' equation are identical. Since the generated POD basis functions are *global*, we do not see any advantage of formulating a POD-HDG

ROM constructed using both variables u_h and \widehat{u}_h . Hence, we only use the field variable u_h to construct the POD model, and the resulting ROM is a DG scheme.

3.1 Computing POD Basis Functions

The method of snapshots is used to construct the POD bases. To this end, let $\{u_h^i\}_{i=0}^{S-1}$ be snapshots obtained from a full order model simulation (11). The POD bases are obtained by the following steps:

- (i) Decompose the data u_h^i into the mean part (\bar{u}_h) and the fluctuating part (u_h^i):

$$u_h^i = \bar{u}_h + u_h^i, \quad \bar{u}_h = \frac{1}{S} \sum_{i=0}^{S-1} u_h^i.$$

- (ii) Build the (symmetric positive definite) correlation matrix $C \in \mathbb{R}^{S \times S}$ with $C_{ij} = \mathcal{M}_h(u_h^i, u_h^j)$.

- (iii) Solve the eigenvalue problem:

$$CW = W\Lambda,$$

where $\Lambda = \text{diag}[\lambda_1, \dots, \lambda_S]$, $W = [w^1, \dots, w^S]$, λ_i is the i th eigenvalue and w^i is the corresponding normalized i th eigenvector.

(iv) Given an integer $r \ll S$, return the first r POD basis functions $\{\phi_j\}_{j=1}^r$, where

$$\phi_j = \frac{1}{\sqrt{\lambda_j}} \sum_{i=0}^{r-1} w_i^j u_h^i, \quad j = 1, \dots, r.$$

Denote the space $S_h^r = \text{span}\{\phi_1, \dots, \phi_r\}$. Since ϕ_j 's are L^2 -orthonormal, the mass matrix associated with the space S_h^r is the identity matrix.

3.2 The Plain POD-DG Scheme

To construct the POD-DG scheme, we first replace the HDG diffusion operator (7d) by a DG operator:

$$\begin{aligned} \mathcal{B}_h^{dg}(\mathbf{u}, \mathbf{v}) = & \sum_{K \in \mathcal{T}_h} \left(\int_K \nabla \mathbf{u} : \nabla \mathbf{v} \, dx - \int_{\partial K} (\llbracket \nabla \mathbf{u} \rrbracket \mathbf{n}) \cdot \text{tang}(\mathbf{v}) \, ds \right. \\ & \left. - \int_{\partial K} (\llbracket \nabla \mathbf{v} \rrbracket \mathbf{n}) \cdot \text{tang}(\mathbf{u}) \, ds + \int_{\partial K} \frac{4k^2}{h} \text{tang}(\llbracket \mathbf{u} \rrbracket) \cdot \text{tang}(\llbracket \mathbf{v} \rrbracket) \, ds \right), \end{aligned} \quad (12)$$

where on each internal facet $F \in \mathcal{E}_h^i$, $\llbracket \mathbf{v} \rrbracket|_F = (\mathbf{v}^+ - \mathbf{v}^-)$ is the standard jump operator, and $\llbracket \mathbf{u} \rrbracket|_{\partial \Omega} = 0$. Next, we notice that the upwinding convection operator (7b) is linear in the first and third arguments, but nonlinear in the second argument, due to the upwinding numerical flux \mathbf{u}^- . This nonlinearity is quite troublesome for ROM in the sense that it can not be computed using an offline procedure. We mention that it is precisely this nonlinear term that provides the upwinding mechanism for the DG operator (7b), which produces extra numerical dissipation to stabilize the scheme (11) in the under-resolved convection-dominated regime. To seek for an efficient implementation, we replace the (*nonlinear*) upwinding flux by the (*linear*) central flux:

$$\tilde{\mathcal{C}}_h^{dg}(\mathbf{w}, \mathbf{u}, \mathbf{v}) = -\frac{1}{2} \sum_{K \in \mathcal{T}_h} \left(\int_K (\mathbf{w} \otimes \mathbf{u}) : \nabla \mathbf{v} \, dx - \int_{\partial K} (\mathbf{w} \cdot \mathbf{n}) (\llbracket \mathbf{u} \rrbracket \cdot \mathbf{v}) \, ds \right). \quad (13)$$

This creates a *trilinear* operator that satisfies the following energy conservation property:

$$\tilde{\mathcal{C}}_h^{dg}(\mathbf{u}_h, \mathbf{u}_h, \mathbf{u}_h) = 0, \quad \forall \mathbf{u}_h \in \mathbf{V}_h^k \cap H(\text{div}^0, \Omega). \quad (14)$$

Finally, the pressure field can be directly eliminated from the POD scheme because all POD basis functions are globally divergence-free, inherited from the snapshots. The semidiscrete plain POD-DG scheme reads as follows: Given initial data $\mathbf{u}_h(0) = \bar{\mathbf{u}}_h + \mathbf{u}_h^0$ with $\mathbf{u}_h^0 \in S_h^r$, for all $t \in (0, T]$, find $\mathbf{u}_h = \bar{\mathbf{u}}_h + \mathbf{u}_h(t)$ with $\mathbf{u}_h(t) \in S_h^r$ such that

$$\mathcal{M}_h \left(\frac{\partial \mathbf{u}_h}{\partial t}, \mathbf{v} \right) + \tilde{\mathcal{C}}_h^{dg}(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}) + \nu \mathcal{B}_h^{dg}(\mathbf{u}_h, \mathbf{v}) = 0, \quad \forall \mathbf{v} \in S_h^r. \quad (15)$$

We again use the CNAB time discretization, and the fully discrete plain POD-DG scheme reads as follows: Given initial data $\mathbf{u}_h^0 = \bar{\mathbf{u}}_h + \mathbf{u}_h^0$ with $\mathbf{u}_h^0 \in S_h^r$, for each integer $n = 1, \dots, M$, find $\mathbf{u}_h^n = \bar{\mathbf{u}}_h + \mathbf{u}_h^n$ with $\mathbf{u}_h^n \in S_h^r$ such that

$$\mathcal{M}_h \left(\frac{\mathbf{u}_h^n - \mathbf{u}_h^{n-1}}{\Delta t}, \mathbf{v} \right) + \tilde{\mathcal{C}}_h^{dg}(\tilde{\mathbf{u}}_h^{n-1/2}, \tilde{\mathbf{u}}_h^{n-1/2}, \mathbf{v}) + \nu \mathcal{B}_h^{dg}(\mathbf{u}_h^{n-1/2}, \mathbf{v}) = 0, \quad \forall \mathbf{v} \in S_h^r. \quad (16)$$

Remark 3.1 (*Offline-online decomposition*) The POD-DG schemes (15) and (16) can be efficiently implemented via a standard offline-online decomposition. Thus, we introduce the offline-precomputable vectors $\mathbf{C0}, \mathbf{B0} \in \mathbb{R}^r$, matrices $\mathbf{C1}, \mathbf{B} \in \mathbb{R}^{r \times r}$, and third order tensor $\mathbf{C} \in \mathbb{R}^{r \times r \times r}$:

$$\begin{aligned} \mathbf{C0}_j &= \tilde{\mathbf{C}}_h^{dg}(\bar{\mathbf{u}}_h, \bar{\mathbf{u}}_h, \phi_j), \\ \mathbf{B0}_j &= \mathbf{B}_h^{dg}(\bar{\mathbf{u}}_h, \phi_j), \\ \mathbf{C1}_{i,j} &= \tilde{\mathbf{C}}_h^{dg}(\bar{\mathbf{u}}_h, \phi_i, \phi_j) + \tilde{\mathbf{C}}_h^{dg}(\phi_i, \bar{\mathbf{u}}_h, \phi_j), \\ \mathbf{B}_{i,j} &= \mathbf{B}_h^{dg}(\phi_i, \phi_j), \\ \mathbf{C}_{i,j,k} &= \tilde{\mathbf{C}}_h^{dg}(\phi_i, \phi_j, \phi_k). \end{aligned}$$

Denote $\mathbf{u}_h = \sum_{j=1}^r a_j(t) \phi_j$, then the semi-discrete scheme (15) is given in the following form:

$$\frac{\partial a_j}{\partial t} + \mathbf{C0}_j + \mathbf{C1}_{ij} a_i + \mathbf{C}_{ikj} a_i a_k + \nu \mathbf{B0}_j + \nu \mathbf{B}_{ij} a_i = 0 \quad (17)$$

Denoting $\mathbf{a}^n = [a_1^n, \dots, a_r^n] \in \mathbb{R}^r$, the fully discrete scheme (16) is then given in the following form, which can be computed efficiently online,

$$\left(\frac{Id}{\Delta t} + \frac{1}{2} \nu \mathbf{B} \right) \mathbf{a}^n = \frac{\mathbf{a}^{n-1}}{\Delta t} - \left(\mathbf{C0} + \mathbf{C1} \tilde{\mathbf{a}}^{n-1/2} + \tilde{\mathbf{a}}^{n-1/2} \mathbf{C} \tilde{\mathbf{a}}^{n-1/2} + \nu \mathbf{B0} + \frac{1}{2} \nu \mathbf{B} \mathbf{a}^{n-1} \right), \quad (18)$$

where $\tilde{\mathbf{a}}^{n-1/2} = \frac{3}{2} \mathbf{a}^{n-1} - \frac{1}{2} \mathbf{a}^{n-2}$.

3.3 The Closure Model

Due to the use of *linear* central numerical flux for the convection operator, the plain POD-DG scheme (16) does not inherit the extra (*upwinding*) convective stabilization property of the original HDG/DG scheme (11) that, however, is the key for the stability of the scheme in the under-resolved convection dominated regime. Hence, it is natural to introduce a *linear* stabilization term that mimics such upwinding mechanism in the POD setting. We further include a standard eddy viscosity closure model originally proposed in [20], in order to improve accuracy/stability of the POD-DG scheme. To this end, we denote the following non-negative matrices $\mathbf{CX}, \mathbf{BX} \in \mathbb{R}^{r \times r}$:

$$\mathbf{CX}_{ik} = \frac{1}{2} \sum_{K \in \mathcal{T}_h} \int_{\partial K} \llbracket \phi_i \rrbracket \cdot \llbracket \phi_k \rrbracket \, ds, \quad (19a)$$

$$\mathbf{BX}_{ik} = \mathbf{B}_h^{dg}(\phi_i, (k/r)^2 \phi_k), \quad (19b)$$

and define the POD-DG closure model as follows:

$$\left(\frac{Id}{\Delta t} + \frac{1}{2} \tilde{\mathbf{B}} \right) \mathbf{a}^n = \frac{\mathbf{a}^{n-1}}{\Delta t} - \left(\mathbf{C0} + \mathbf{C1} \tilde{\mathbf{a}}^{n-1/2} + \tilde{\mathbf{a}}^{n-1/2} \mathbf{C} \tilde{\mathbf{a}}^{n-1/2} + \nu \mathbf{B0} + \frac{1}{2} \tilde{\mathbf{B}} \mathbf{a}^{n-1} \right), \quad (20)$$

where $\tilde{\mathbf{B}} = \nu \mathbf{B} + c_1 \mathbf{CX} + c_2 \mathbf{BX}$, with $c_1, c_2 \geq 0$ being two tunable constants that are problem dependent. Here the matrix \mathbf{BX} corresponds to an eddy viscosity model with a quadratic viscosity kernel [20], and the matrix \mathbf{CX} can be interpreted as an upwinding stabilization term (compared with DG upwinding in (8)). We call the term with \mathbf{BX} a diffusive stabilization, and

the term with CX a convective stabilization. We remark that if we take $c_1 = \max |\mathbf{u}_h|$, then the convective stabilization term scales similarly as the full order model case. However, our numerical results in the next section indicates that taking $c_1 = \max |\mathbf{u}_h|$ is too small to make such convective stabilization term effective in the POD setting. Actually, in a case for the Burgers' equation, we need to take $c_1 = 2 \times 10^8$ (see Example 1 in Sect. 4 below) to see the positive impact of this stabilization term. This observation also partially justify our choice of *linear* central numerical flux in the convection operator (13) over the *nonlinear* upwinding numerical flux for the plain POD-DG scheme (16).

Finally, we remark that the two parameters c_1 and c_2 are tuned purely at the online stage, such tuning cost is negligible comparing to the computational cost of the full order model (10).

4 Numerical Results

In this section, we present some numerical examples for the POD-DG closure model (20). The NGSolve software [22] is used for the simulations. In all the numerical examples below, we collect enough snapshots at equidistant time instances to capture the flow dynamics. The number of snapshots are chosen such that further increasing it will not produce better results for the POD models. But other approaches such as the ones based on incremental ideas in [12,17] can be applied for better efficiency.

4.1 Example 1. Burgers' Equation: Discontinuous Initial Condition

We consider the Burgers' equation (2) with $\nu = 10^{-4}$ and the periodic boundary conditions. The initial condition is taken to be a step function

$$u(0) = \begin{cases} 1 & \text{if } x < 0.5, \\ 0 & \text{if } x \geq 0.5. \end{cases},$$

and the final time is $T = 1$. Two cases of the full order model (10) are tested here that associate with different discretization parameters, including mesh size h of the uniform mesh, polynomial degree k , and the uniform time step size Δt .

- (i) Slightly resolved case: $h = 10^{-4}$, $k = 2$, $\Delta t = 0.1h$.
- (ii) Fully resolved case: $h = 10^{-4}$, $k = 6$, $\Delta t = 0.04h$.

To build the POD basis, we collect 501 snapshots in the time interval $[0, 1]$ taken at equidistant time instances. The numerical solutions for the fully resolved case ($k = 6$) at $t = 0.5$ and $t = 1$ are shown in Fig. 1. We observe the sharp gradient is resolved within 2 cells.

The eigenvalues of the correlation matrix C are shown in Fig. 2, where we do not observe any significant difference for both cases.

Figure 3 visualizes three POD basis functions $\phi_1(x)$, $\phi_2(x)$, and $\phi_{10}(x)$ for both cases. Again, we observe no significant difference between these two cases.

To build the POD model, we use $r = 20$ basis functions, which capture about 97.87% of the total energy for both cases, that is, $\frac{\sum_{i=1}^r \lambda_i}{\sum_{i=1}^S \lambda_i} \approx 0.9787$ with λ_i being the i th eigenvalue of the correlation matrix $C \in \mathbb{R}^{S \times S}$. Numerical results for the plain POD-DG scheme (18), along with the computed L^2 - and L^1 -errors for $u_h^{rom} - u_h^{fom}$ at time $t = 0, 0.5, 1$, where u_h^{fom} is the solution to the full order model (10), and u_h^{rom} is the solution to the POD-DG

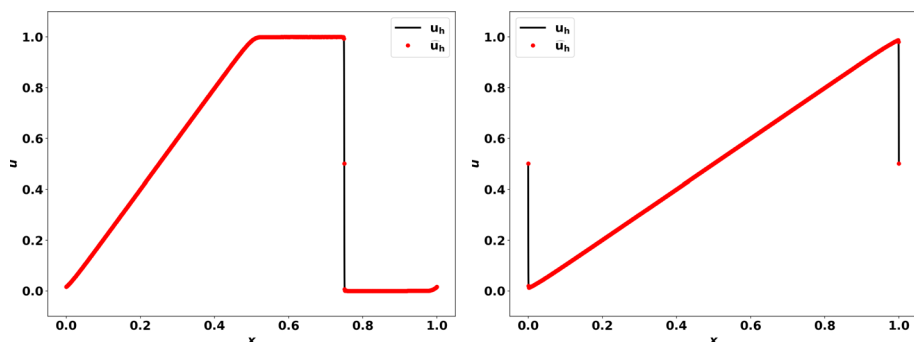


Fig. 1 Example 1. Numerical solution at $t = 0.5$ (left) and $t = 1$ (right). Black line: u_h . Red dots: \hat{u}_h (Color figure online)

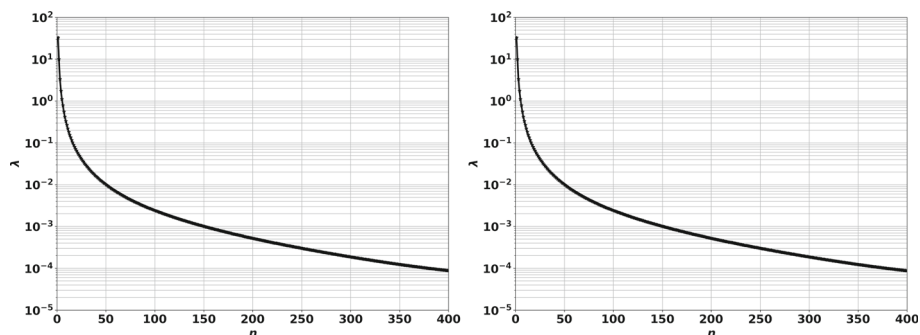


Fig. 2 Example 1. First 400 eigenvalues of the correlation matrix C . Left: slightly resolved case $k = 2$. Right: fully resolved case $k = 6$

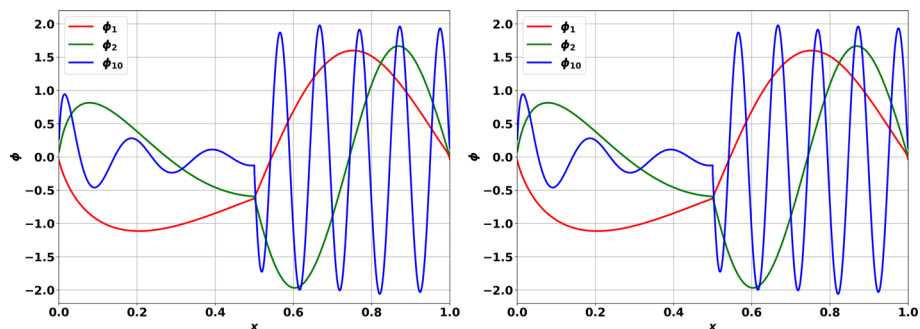


Fig. 3 Example 1. Illustrative examples of POD basis functions. Left: slightly resolved case $k = 2$. Right: fully resolved case $k = 6$

model (18), are shown in Fig. 4. It is clearly seen that the plain POD-DG model produces very oscillatory results, with the associated error for $t = 0.5$ and $t = 1$ being an order of magnitude larger than the initial projection error at $t = 0$.

Next, we consider the POD-DG closure model (20) with only convective stabilization ($c_1 > 0, c_2 = 0$). We refer to the resulting model as the POD-DG-C model. We tune the parameter $c_1 = 10^4$ for $k = 2$, and $c_1 = 2 \times 10^8$ for $k = 6$ to produce satisfactory results.

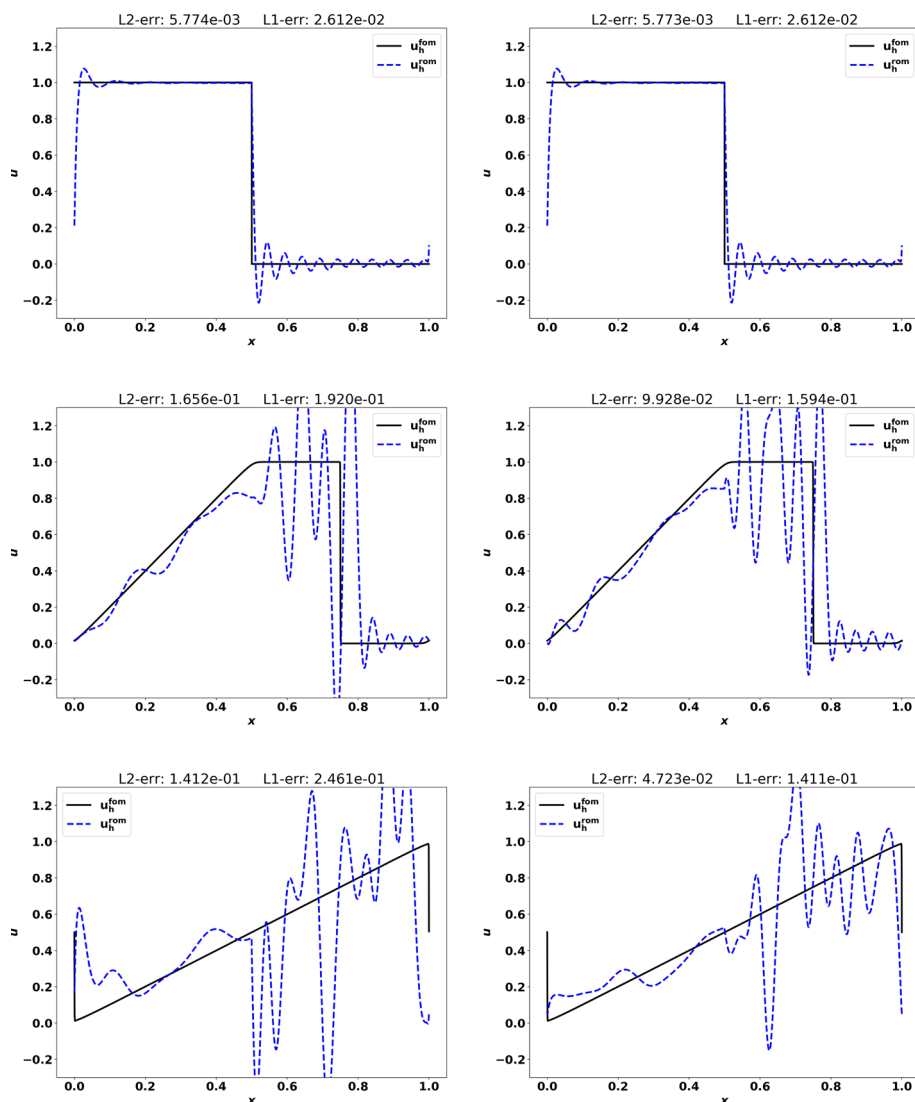


Fig. 4 Example 1. Numerical solution u_h^{rom} for the POD-DG model along with full order model solution u_h^{fom} . Top: $t = 0$. Middle: $t = 0.5$. Bottom: $t = 1$. Left: slightly resolved case $k = 2$. Right: fully resolved case $k = 6$

Note that for the classical upwinding DG scheme (4), the parameter c_1 corresponds to the magnitude of the solution $\|u_h\| \approx 1$, which is too small for the POD-DG model to suppress numerical oscillation. We don't have a physical interpretation for the parameter c_1 , but argue that our global POD DG basis functions are very smooth across element boundaries (which is especially true for the fully resolved case $k = 6$), and one needs to have a large weighting coefficient c_1 to make the convective stabilization term effective. The associated numerical results at $t = 0.5$ and $t = 1$ are shown in Fig. 5. Significant improvement over the results of the plain POD-DG model can be clearly observed. We also found that the errors at $t = 0.5$

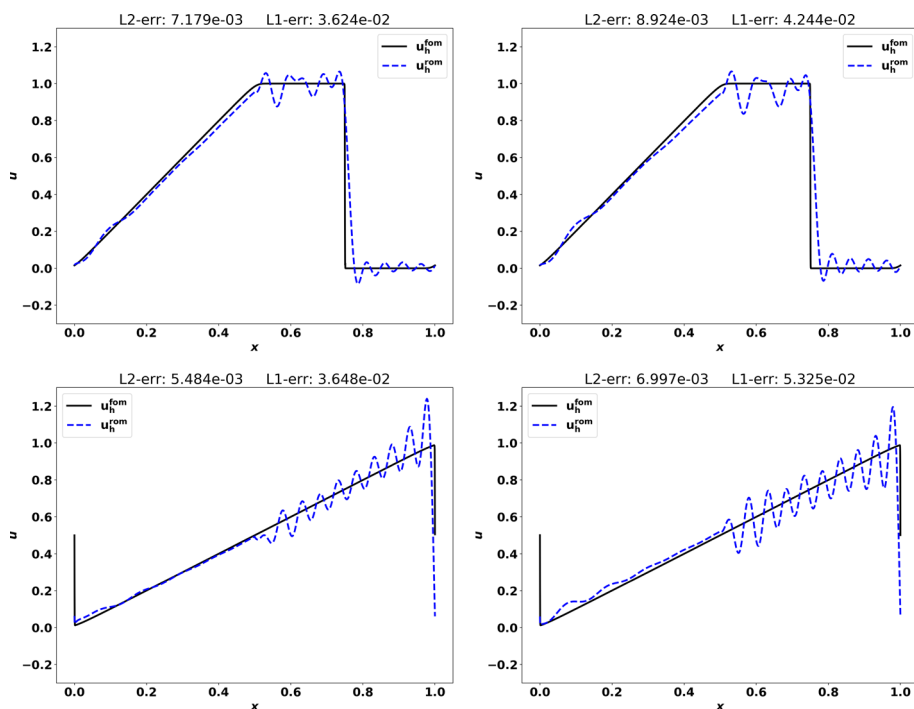


Fig. 5 Example 1. Numerical solution u_h^{rom} for the POD-DG-C model along with full order model solution u_h^{fom} . Top: $t = 0.5$. Bottom: $t = 1$. Left: slightly resolved case $k = 2$, $c_1 = 10^4$. Right: fully resolved case $k = 6$, $c_1 = 2 \times 10^8$

and $t = 1$ for the POD-DG-C closure model are of similar magnitude to the POD projection error at $t = 0$ in Fig. 4. However, the POD solution is still oscillatory behind the shock.

Furthermore, we consider the POD-DG closure model (20) with both convective and diffusive stabilizations, which is referred to the POD-DG-CD model. We use the same parameter c_1 as the POD-DG-C model, i.e. $c_1 = 10^4$ for $k = 2$, and $c_1 = 2 \times 10^8$ for $k = 6$; and set $c_2 = 0.01$. The associated numerical results at $t = 0.5$ and $t = 1$ are shown in Fig. 6. We observe that the errors at $t = 0.5$ and $t = 1$ for the POD-DG-CD model is similar and slightly smaller than those for the POD-DG-C model, and the post-shock oscillations are also diminished.

Finally, the time evolution of the three models along with the full order model are presented in Fig. 7 for $k = 2$. The results for $k = 6$ are similar and are omitted to save space.

4.2 Example 2. Burger's Equation: Smooth Initial Condition

We consider the same problem as Example 1, but with the following smooth initial condition:

$$u(0) = \exp(-200(x - 0.3)^2).$$

Very similar results as those for Example 1 are observed. In particular, we need to take $c_1 = 10^4$ for the case $k = 2$, and $c_1 = 10^8$ for the case $k = 6$ to make the POD-DG-C model produce satisfactory results, and use the POD-DG-CD model with $c_2 = 0.01$ to further

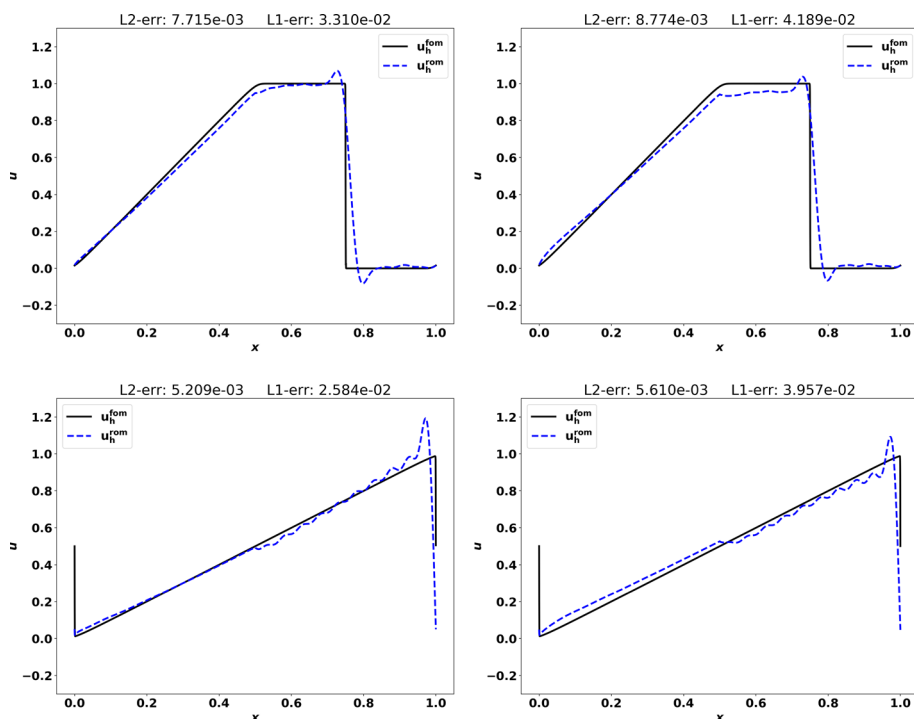


Fig. 6 Example 1. Numerical solution u_h^{rom} for the POD-DG-CD model along with full order model solution u_h^{fom} . Top: $t = 0.5$. Bottom: $t = 1$. Left: slightly resolved case $k = 2$, $c_1 = 10^4$, $c_2 = 0.01$. Right: fully resolved case $k = 6$, $c_1 = 2 \times 10^8$, $c_2 = 0.01$

improve the results. We present in Fig. 8 the numerical solution of different POD-DG models at final time $t = 1$ for the slightly resolved case $k = 2$. It is again clear that the POD-DG-C model produce better results than the plain POD-DG model, and the POD-DG-CD model further improves the results of POD-DG-C model by suppressing post-shock oscillations.

4.3 Example 3. Navier–Stokes: 2D Flow Past a Cylinder, $Re = 100$

We consider the classical flow past a cylinder benchmark problem [21]. The domain is a rectangular channel with an almost vertically centered circular obstacle, c.f. Fig. 9,

$$\Omega := [0, 2.2] \times [0, 0.41] \setminus \{(x, y) : \|(x, y) - (0.2, 0.2)\|_2 \leq 0.05\}.$$

The boundary is decomposed into $\Gamma_{in} := \{x = 0\}$, the inflow boundary, $\Gamma_{out} := \{x = 2.2\}$, the outflow boundary, and $\Gamma_{wall} := \partial\Omega \setminus (\Gamma_{in} \cup \Gamma_{out})$, the wall boundary. On Γ_{out} we prescribe natural boundary conditions $(-\nu \nabla \mathbf{u} + p\mathbf{I})\mathbf{n} = 0$, on Γ_{wall} homogeneous Dirichlet boundary conditions for the velocity (no-slip) and on Γ_{in} the inflow Dirichlet boundary conditions

$$\mathbf{u}(0, y, t) = 6\bar{u} y(0.41 - y)/0.41^2 \cdot (1, 0),$$

with $\bar{u} = 1$ the average inflow velocity. The viscosity is taken to be $\nu = 10^{-3}$, hence Reynolds number $Re = \bar{u}D/\nu = 100$, where $D = 0.1$ is the disc diameter.

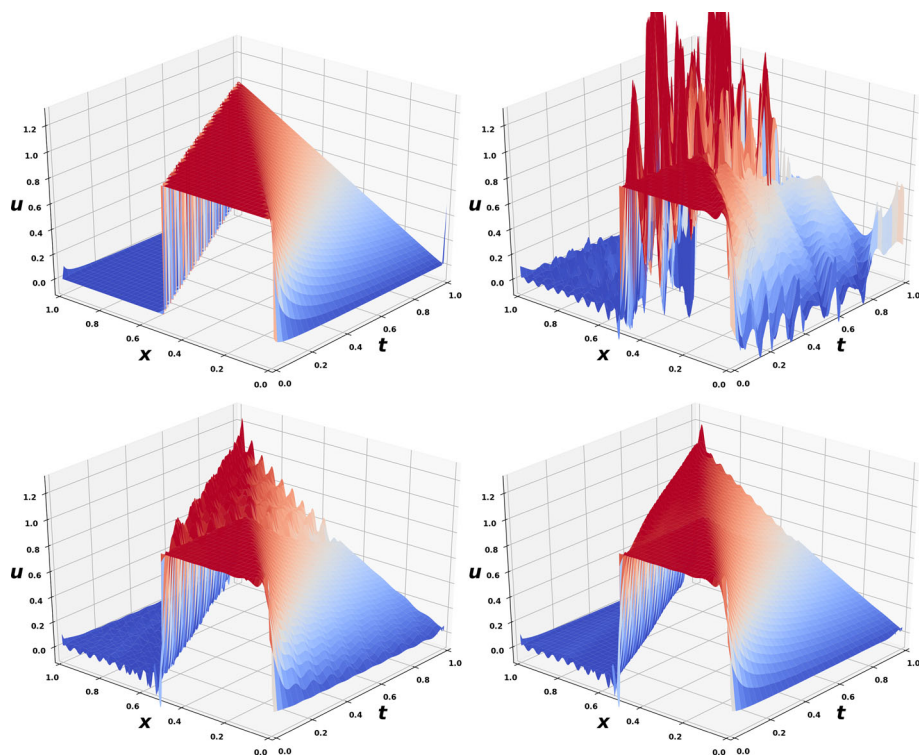


Fig. 7 Example 1. Time evolution of numerical solutions. Top left: full order model. Top right: POD-DG. Bottom left: POD-DG-C. Bottom right: POD-DG-CD. Slightly resolved case $k = 2$

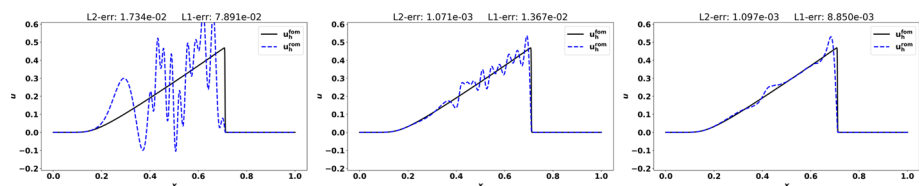


Fig. 8 Example 2. Numerical solution at final time $t = 1$. Left: POD-DG model. Middle: POD-DG-C model with $c_1 = 10^4$, $c_2 = 0$. Right: POD-DG-CD model with $c_1 = 10^4$, $c_2 = 0.01$. 20 POD bases are used. Slightly resolved case $k = 2$, $h = 10^{-4}$

For this Reynolds number, the flow turns into a time-periodic behaviour with a vortex shedding behind the cylinder. For the FOM, we consider the scheme (11) with polynomial degree $k = 3$ on a (curved) unstructured triangular mesh with 292 triangular elements, and take time step size $\Delta t = 0.001$. A precomputed fully developed velocity profile is used for the initial condition; see Fig. 9 for the geometry, the mesh and the initial velocity field.

To build the POD bases, we collect 401 snapshots in the time interval $[0, 2]$ taken at equidistant time instance. To build the POD model, we use 6 POD bases which capture about 99.81% of the total energy and run the simulation up to time $T = 20$. We consider the plain POD-DG model and the POD-DG-C model with $c_1 = 5$. The constant c_1 is tuned to yield relatively the smallest L^2 -error between FOM and ROM solutions at final time for a range of

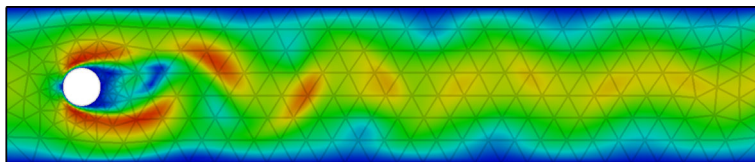


Fig. 9 Example 3: the initial velocity field (color corresponding to velocity magnitude $\|\mathbf{u}_h\|_2$ from 0 to 2.17)

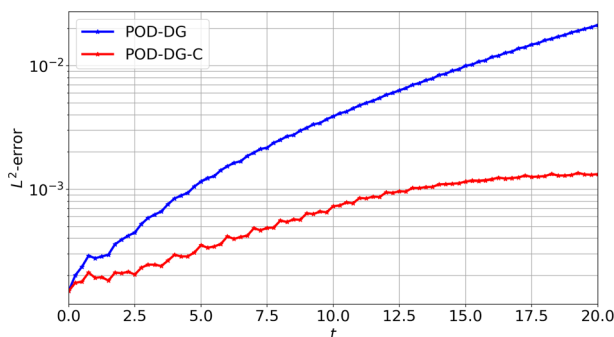


Fig. 10 Example 3: time evolution of the L^2 -velocity error $\|\mathbf{u}_h^{fom} - \mathbf{u}_h^{rom}\|$

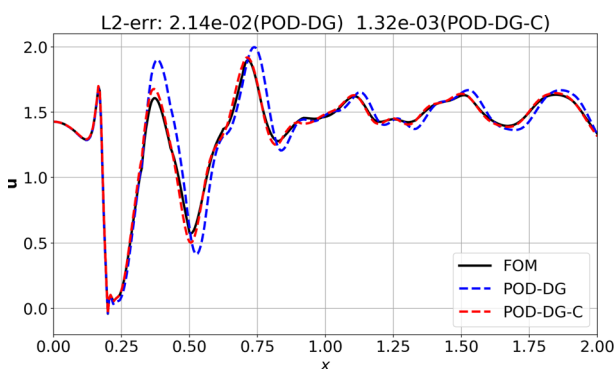


Fig. 11 Example 3: x -component of velocity field along cut line $y = 0.25$ at time $t = 20$

choices. Taking c_1 too big or too small leads to less accurate approximations. It is interesting to observe that this time c_1 is close to the maximum velocity magnitude $v_{\max} \approx 2.17$, which is very different to the scaling in the Burgers' equation cases in Examples 1–2. Here, probably due to the relative small Reynolds number, we find that adding extra diffusive stabilization in (20) does not improve the results. Hence, results for the POD-DG-CD model will not be shown. The time evolution of the L^2 velocity error $\|\mathbf{u}_h^{fom} - \mathbf{u}_h^{rom}\|$ is plotted in Fig. 10. We observe that the error for the POD-DG-C model is an order of magnitude smaller than that for the plain POD-DG model at time $t = 20$.

We plot the x -component of the velocity field along the cut line $y = 0.25$ at time $t = 20$ in Fig. 11. Clearly the result for the POD-DG-C model is closer to FOM than that for the plain POD-DG model, which produces a visible phase shift.

Finally, the velocity magnitude contour lines at time $t = 5$ and $t = 20$ for different models are shown in Fig. 12. Here we observe that at time $t = 5$, both POD-DG and POD-DG-C

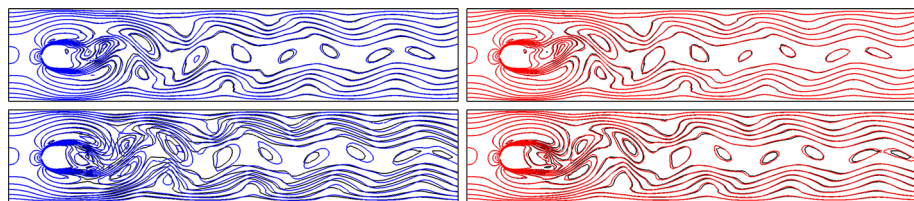


Fig. 12 Example 3 (ROMs are 6-dimensional): velocity contour at time $t = 5$ (top) and $t = 20$ (bottom). 10 equispaced contour lines from 0 to 2.17. Black: FOM. Blue: POD-DG. Red: POD-DG-C with $c_1 = 5$ (Color figure online)

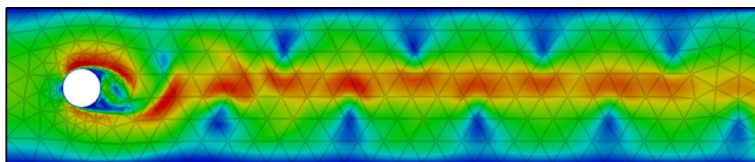


Fig. 13 Example 4: the initial velocity field (color corresponding to velocity magnitude $\|u_h\|_2$ from 0 to 2.4)

models produce similar results as the FOM. On the other hand, visible phase shift, especially behind the cylinder, is observed for the POD-DG model (in blue) at time 20, while the result for POD-DG-C (in red) is still in good agreement with FOM.

4.4 Example 4. Navier–Stokes:2D Flow Past a Cylinder, $Re = 500$

We consider the same problem as Example 3, but with a larger Reynolds number $Re = 500$. For the FOM, we consider the scheme (11) with polynomial degree $k = 6$ on the mesh used in Example 3. The initial (fully developed) velocity field is shown in Fig. 13.

To build the POD bases, we collect 501 snapshots in the time interval $[0, 2]$ taken at equidistant time instance. To construct the POD model, we use 10 POD bases which capture about 99.90% of the total energy and run the simulation up to time $T = 20$. We consider the plain POD-DG model and the POD-DG-C model with $c_1 = 12$, which is tuned to yield relatively smallest L^2 -error between FOM and ROM solutions at final time. Again, we find that adding extra diffusive stabilization in (20) does not improve the results. Hence, results for the POD-DG-CD model will not be shown. The time evolution of the L^2 velocity error $\|u_h^{fom} - u_h^{rom}\|$ is plotted in Fig. 14. We observe that the error for the POD-DG-C model is again an order of magnitude smaller than that for the plain POD-DG model at time $t = 20$.

We plot the x -component of the velocity field along the cut line $y = 0.25$ at time $t = 20$ in Fig. 15, and velocity contour lines at time $t = 5$ and $t = 20$ in Fig. 16. Similar results as those in Example 3 is observed. In particular, while both models produces similar results at time $t = 5$. Significant improvement from POD-DG-C model over the plain POD-DG model is observed for the velocity magnitude contour lines at time $t = 20$. This indicates our POD-DG-C model is more accurate than POD-DG model for long time simulations.

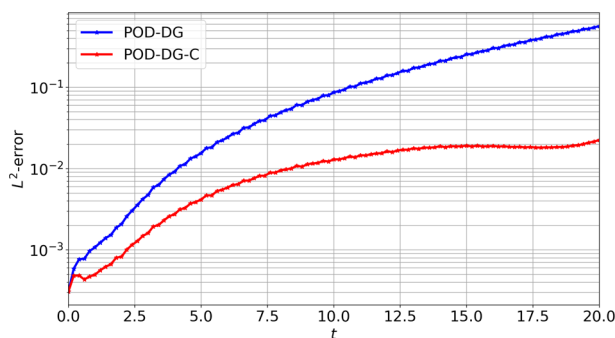


Fig. 14 Example 4: time evolution of the L^2 -velocity error $\|u_h^{fom} - u_h^{rom}\|$

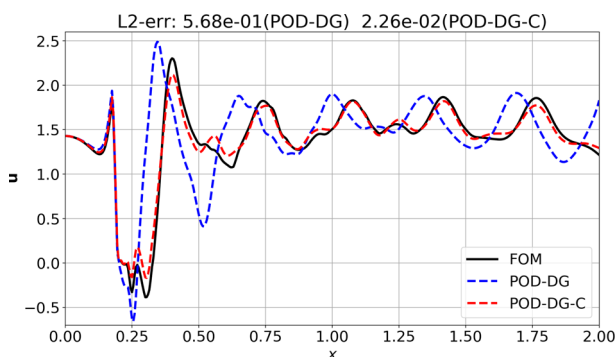


Fig. 15 Example 4: x -component of velocity field along cut line $y = 0.25$ at time $t = 20$

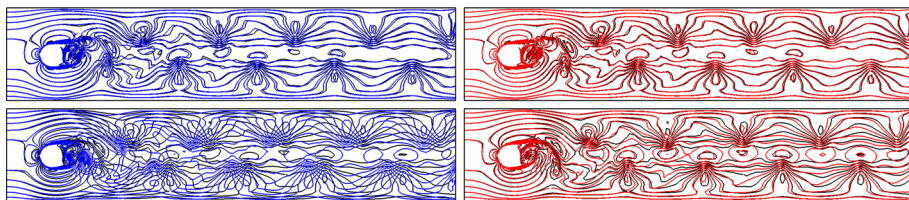


Fig. 16 Example 4 (ROMs are 10-dimensional): velocity contour at time $t = 5$ (top) and $t = 20$ (bottom). 10 equispaced contour lines from 0 to 2.4. Black: FOM. Blue: POD-DG. Red: POD-DG-C with $c_1 = 12$ (Color figure online)

4.5 Example 5. Incompressible Euler: Double Shear Layer Problem

In our last example, we consider the classical double shear layer problem [6]. We solve the Euler equation (3) with $v = 0$ on a periodic domain $[0, 2\pi] \times [0, 2\pi]$ with an initial condition:

$$u_1(x, y, 0) = \begin{cases} \tanh((y - \pi/2)/\rho) & y \leq \pi \\ \tanh((3\pi/2 - y)/\rho) & y > \pi, \end{cases} \quad (21)$$

$$u_2(x, y, 0) = \delta \sin(x), \quad (22)$$

with $\rho = \pi/15$ and $\delta = 0.05$.

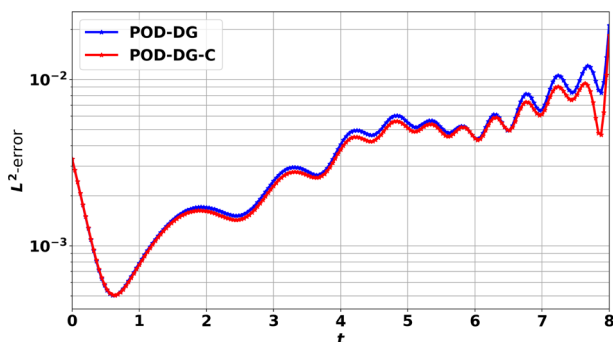


Fig. 17 Example 5: time evolution of the L^2 -velocity error $\|u_h^{fom} - u_h^{rom}\|$

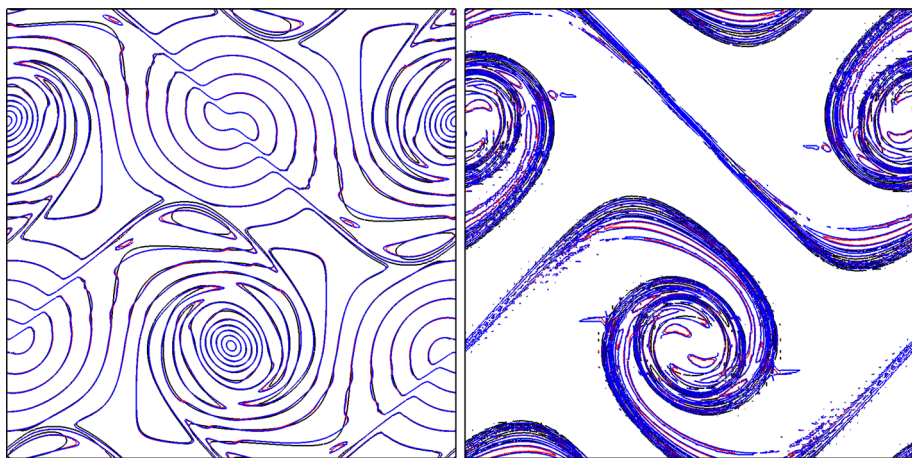


Fig. 18 Example 5 (ROMs are 10-dimensional): Left: 10 equispaced velocity magnitude contour lines from 0 to 1.5. Right: 10 equispaced vorticity contour lines from -4.9 to 4.9 . Black: FOM. Blue: POD-DG. Red: POD-DG-C with $c_1 = 40$ (Color figure online)

For the FOM (11), we use \mathbb{P}^3 approximation on fixed uniform structured triangular meshes with mesh size $2\pi/64$ and run the simulation up to time $t = 8$ with time step size $\Delta t = 0.001$. To build the POD bases, we collect 401 snapshots in the time interval $[0, 8]$ taken at equidistant time instance. To build the POD model, we use 10 POD bases which captures about 99.95% of the total energy and run the simulation up to time $T = 8$. We consider the plain POD-DG model and the POD-DG-C model. The parameter c_1 in the POD-DG-C model is tuned to be $c_1 = 40$. The time evolution of the L^2 velocity error $\|u_h^{fom} - u_h^{rom}\|$ is plotted in Fig. 17. In contrast to Examples 3–4, we observe that the error for both models are very similar, which indicates that our current convective stabilization approach is not effective for the current problem. We further remark that we also observe no accuracy improvement by considering the POD-DG-CD model.

Finally, we plot velocity magnitude ($\|u_h\|$) and vorticity ($\nabla \times u_h$) contour lines for the two models along with the results for the FOM at final time $t = 8$ in Fig. 18. It can be observed that the results for both models are very similar and are close to the FOM results. This is a rather surprising result as the POD-DG model does not introduce any spatial numerical dissipation,

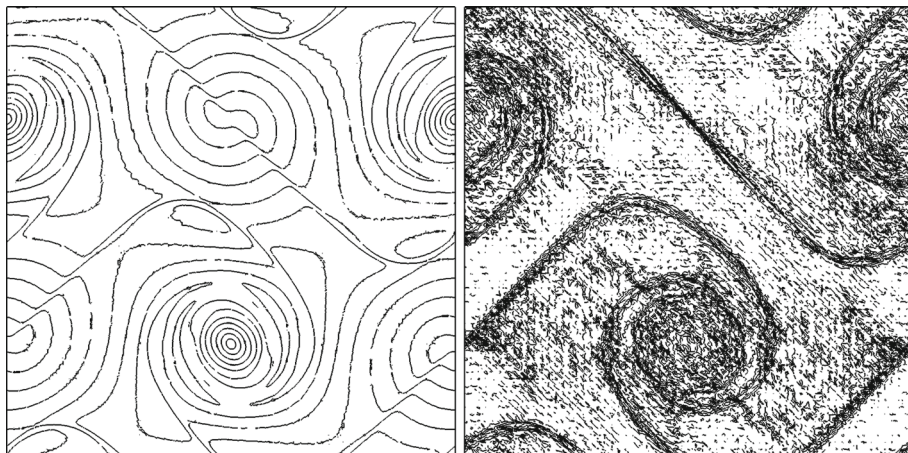


Fig. 19 Example 5: Left: 10 equispaced velocity magnitude contour lines from 0 to 1.5. Right: 10 equispaced vorticity contour lines from -4.9 to 4.9 . Results for (conservative) C-FOM

yet its vorticity approximation is still free from large oscillations. For comparison, we also plot in Fig. 19 the vorticity approximations for the FOM (11) with the upwinding convection operator \mathcal{C}_h^{dg} replaced by the conservative version $\tilde{\mathcal{C}}_h^{dg}$ in (13), which we denote as C-FOM. It is clear that the (conservative) POD-DG model has better stability property than C-FOM. We conjecture the reason for the superior performance of POD-DG model over C-FOM is that the global POD bases obtained from FOM (11) might have some extra built-in stabilization properties.

5 Conclusion

We have presented a POD-DG reduced order model for the viscous Burgers' equation and the incompressible Navier–Stokes equations discretized using an IMEX HDG/DG scheme. A novel offline-computable closure model was introduced for the POD-DG ROM which further improves its stability and accuracy. Numerical results show the superior performance of the proposed closure model compared with a plain POD-DG scheme without the closure model. In future work, we will pursue in the same direction and investigate the proposed model in the parametrized flow problems with applications in flow control and optimization.

Data Availability Statement The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request

References

1. Antonietti, P.F., Pacciarini, P., Quarteroni, A.: A discontinuous Galerkin reduced basis element method for elliptic problems. *ESAIM: Math. Model. Numer. Anal.* **50**, 337–360 (2016)
2. Antoulas, A.: Approximation of large-scale dynamical systems. In: *Advances in Design and Control. Society for Industrial and Applied Mathematics (SIAM)*, vol. 6. Philadelphia, PA (2005)
3. Antoulas, A., Beattie, C., Gügürcin, S.: *Interpolatory methods for model reduction. Society for Industrial and Applied Mathematics (SIAM)*. Philadelphia, PA (2020)

4. Ascher, U.M., Ruuth, S.J., Wetton, B.T.R.: Implicit-explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal.* **32**, 797–823 (1995)
5. Ballarin, F., Manzoni, A., Quarteroni, A., Rozza, G.: Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier–Stokes equations. *Int. J. Numer. Methods Eng.* **102**, 1136–1161 (2015)
6. Bell, J.B., Colella, P., Glaz, H.M.: A second-order projection method for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **85**, 257–283 (1989)
7. Benner, P., Gugercin, S., Willcox, K.: A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.* **57**, 483–531 (2015)
8. Caiazzo, A., Iliescu, T., John, V., Schischlowa, S.: A numerical investigation of velocity-pressure reduced order models for incompressible flows. *J. Comput. Phys.* **259**, 598–616 (2014)
9. Carlberg, K., Farhat, C., Cortial, J., Amsallem, D.: The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comput. Phys.* **242**, 623–647 (2013)
10. Chaturantabut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* **32**, 2737–2764 (2010)
11. Cockburn, B.: Static condensation, hybridization, and the devising of the HDG methods. In: Barrenea, G., Brezzi, F., Cangiani, A., Georgoulis, E. (eds.) *Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations*. Lecture Notes in Computational Science and Engineering, vol. 114, pp. 129–177. Springer, Cham (2016)
12. Fareed, H., Singler, J.R., Zhang, Y., Shen, J.: Incremental proper orthogonal decomposition for PDE simulation data. *Comput. Math. Appl.* **75**, 1942–1960 (2018)
13. Gunzburger, M.D.: Perspectives in flow control and optimization. In: *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), vol. 5. Philadelphia, PA (2003)
14. Hesthaven, J.S., Rozza, G., Stamm, B., et al.: *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, vol. 590. Springer, Cham (2016)
15. Lehrenfeld, C.: Hybrid Discontinuous Galerkin methods for solving incompressible flow problems. Diploma Thesis, MathCCES/IGPM, RWTH Aachen (2010)
16. Lehrenfeld, C., Schöberl, J.: High order exactly divergence-free hybrid discontinuous Galerkin methods for unsteady incompressible flows. *Comput. Methods Appl. Mech. Eng.* **307**, 339–361 (2016)
17. Nigro, P.S.B., Anand, M., Teixeira, Y., Pimenta, P.M., Wriggers, P.: An adaptive model order reduction by proper snapshot selection for nonlinear dynamical problems. *Comput. Mech.* **57**, 537–554 (2016)
18. Peherstorfer, B., Willcox, K.: Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Eng.* **306**, 196–215 (2016)
19. Quarteroni, A., Manzoni, A., Negri, F.: *Reduced Basis Methods for Partial Differential Equations: An Introduction*, vol. 92. Springer, Cham (2015)
20. San, O., Iliescu, T.: Proper orthogonal decomposition closure models for fluid flows: Burgers equation. *Int. J. Numer. Anal. Model. Ser. B* **5**, 217–237 (2014)
21. Schäfer, M., Turek, S., Durst, F., Krause, E., Rannacher, R.: Benchmark computations of laminar flow around a cylinder. In: Hirschel, E.H. (ed.) *Flow Simulation with High-Performance Computers II*. Notes on Numerical Fluid Mechanics (NNFM), vol. 48, pp. 547–566 (1996)
22. Schöberl, J.: C++11 Implementation of finite elements in NGSolve. ASC Report 30/2014, Vienna University of Technology, Institute for Analysis and Scientific Computing (2014)
23. Shen, J., Singler, J.R., Zhang, Y.: HDG-POD reduced order model of the heat equation. *J. Comput. Appl. Math.* **362**, 663–679 (2019)
24. Sirovich, L.: Turbulence and the dynamics of coherent structures. I. Coherent structures. *Q. Appl. Math.* **45**, 561–571 (1987)
25. Uzunca, M., Karasözen, B.: Energy stable model order reduction for the Allen–Cahn equation. In: Benner, P., Ohlberger, M., Patera, A., Rozza, G., Urban, K. (eds.) *Model Reduction of Parametrized Systems*, vol. 17, pp. 403–419. Springer (2017)
26. Wang, Z.: Nonlinear model reduction based on the finite element method with interpolated coefficients: semilinear parabolic equations. *Numer. Methods Part. Differ. Equ.* **31**, 1713–1741 (2015)
27. Yano, M.: Discontinuous Galerkin reduced basis empirical quadrature procedure for model reduction of parametrized nonlinear conservation laws. *Adv. Comput. Math.* **45**, 2287–2320 (2019)