Weierstraß-Institut
für Angewandte Analysis und Stochastik

Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

Error analysis of a SUPG-stabilized POD-ROM method for convection-diffusion-reaction equations

Volker John[1,2], Baptiste Moreau[2], Julia Novo[3]

submitted: September 17, 2021

[1] Freie Universität Berlin
Department of Mathematics and Computer Science
Arnimallee 6, 14195 Berlin
Germany

[2] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: baptiste.moreau@wias-berlin.de
volker.john@wias-berlin.de

[3] Departamento de Matemáticas
Universidad Autónoma de Madrid
Instituto de Ciencias Matemáticas
CSIC-UAM-UC3M-UCM
Spain
E-Mail: julia.novo@uam.es

No. 2874
Berlin 2021

2020 *Mathematics Subject Classification.* 65M60.

Key words and phrases. Convection-diffusion-reaction equations, convection-dominated regime, SUPG-ROM method, error analysis.

The research of Julia Novo was supported by Spanish MINECO under grants PID2019-104141GB-I00 and VA169P20 (Junta de Castilla y Leon, ES).

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Error analysis of a SUPG-stabilized POD-ROM method for convection-diffusion-reaction equations

Volker John, Baptiste Moreau, Julia Novo

Abstract

A reduced order model (ROM) method based on proper orthogonal decomposition (POD) is analyzed for convection-diffusion-reaction equations. The streamline-upwind Petrov–Galerkin (SUPG) stabilization is used in the practically interesting case of dominant convection, both for the full order method (FOM) and the ROM simulations. The asymptotic choice of the stabilization parameter for the SUPG-ROM is done as proposed in the literature. This paper presents a finite element convergence analysis of the SUPG-ROM method for errors in different norms. The constants in the error bounds are uniform with respect to small diffusion coefficients. Numerical studies illustrate the performance of the SUPG-ROM method.

1 Introduction

Reduced order modeling (ROM) is a popular technique for performing very efficient simulations for time-dependent problems with a reasonable accuracy of the results. To this end, one simulation of a fine grid is performed, a so-called full order method (FOM) simulation, and the numerical solution, sometimes together with derived quantities, at certain time instants is stored, forming the set of so-called snapshots. In the context of the finite element method, the snapshots are utilized to construct a basis of a subspace of the linear space spanned by the snapshots. This ROM basis, ideally only a dozen to a few dozen functions, 'knows' important features of the solution of the problem. Then, the underlying problem, or problems which are in some sense close to this problem, are discretized with the ROM basis such that only systems with very small dimension have to be solved. The error committed with the ROM solution is usually somewhat larger than the error obtained with the FOM, but the ROM simulations are much faster.

A question of interest from the analytic point of view is the size of the error between the ROM solution and the solution of the continuous problem in order to quantify the loss of accuracy between the FOM and the ROM solutions. This question will be studied here for linear time-dependent convection-diffusion-reaction equations

$$\begin{array}{rcll} \partial_t u - \varepsilon \Delta u + \boldsymbol{b} \cdot \nabla u + cu & = & f & \text{in } (0,T] \times \Omega, \\ u & = & 0 & \text{on } [0,T] \times \partial\Omega, \\ u(0,\boldsymbol{x}) & = & u_0(\boldsymbol{x}) & \text{in } \Omega, \end{array} \qquad (1)$$

DOI 10.20347/WIAS.PREPRINT.2874 Berlin 2021

where Ω is a bounded open domain in $\mathbb{R}^d$, $d \in \{1,2,3\}$, with polyhedral Lipschitz boundary $\partial\Omega$, $\boldsymbol{b}(t,\boldsymbol{x})$ and $c(t,\boldsymbol{x})$ are given functions, $\varepsilon > 0$ is a constant diffusion coefficient, $u_0(\boldsymbol{x})$ is a given initial condition, and T is a given final time. In the following, it is assumed that there is a constant $\mu_0 > 0$ such that[1]

$$0 < \mu_0 \le \mu(t,\boldsymbol{x}) = \left(c - \frac{1}{2}\nabla\cdot\boldsymbol{b}\right)(t,\boldsymbol{x}), \quad \forall\,(t,\boldsymbol{x}) \in [0,T]\times\Omega. \tag{2}$$

The numerical analysis will even require that $\boldsymbol{b}$ is divergence-free, such that (2) reduces to a condition for the reaction field.

Equations of type (1) model the transport of energy (temperature) or concentrations. In practice, the convective transport with the flow field $\boldsymbol{b}$ is often much stronger than the transport via molecular diffusion. In this situation, one speaks of the convection-dominated regime. Mathematically, this regime is given if $\varepsilon \ll \|\boldsymbol{b}\|_{L^\infty(L^\infty)}L$, where L is a characteristic length scale of the problem and $\|\cdot\|_{L^\infty(L^\infty)}$ is the essential supremum norm in the time-space domain. A characteristic feature of solutions of (1) in the convection-dominated regime is the appearance of layers. These are structures of width $\mathcal{O}(\sqrt{\varepsilon})$ to $\mathcal{O}(\varepsilon)$ where the norm of the gradient of the solution is very large. A major consequence is that for small diffusion coefficients, layers cannot be resolved by affordable meshes.

The standard Galerkin finite element method (FEM) tries to resolve all important features of the solution. It turns out that it fails if layers are present, i.e., the numerical solution is, from the beginning, polluted with spurious oscillations, which increase and usually lead to a blow-up of the simulations. A well known remedy is the use of so-called stabilized discretizations. The most popular one is probably the streamline-upwind Petrov–Galerkin (SUPG) method, which was proposed in [13, 3]. With the SUPG method, the spurious oscillations are greatly reduced and localized to neighborhoods of layers. A finite element error analysis of this method for problems of type (1) is presented in [15]. In this paper, the SUPG FEM will be utilized as FOM. For applying the ROM methodology, the proper orthogonal decomposition (POD) technique will be applied, which is probably the most popular technique. The main goal of POD consists in finding a low dimensional basis that approximates the snapshots, e.g., see [24] for details. The SUPG stabilization has been already used in the context of ROM simulations. In [10], the choice of the stabilization parameter is studied and in [11], the choice of the initial condition with the goal of reducing spurious oscillations. In [1], see also [22], stabilized POD-ROM methods to simulate convection-dominated convection-diffusion-reaction equations are considered. In these works, a local projection stabilization (LPS) streamline-diffusion stabilization term is added to the ROM model. To the best of our knowledge, until now, no bounds with constants independent of inverse powers of the diffusion coefficient, so-called robust estimates, have been proved for any POD-ROM approach for convection-diffusion-reaction equations.

[1] Note that (2) does not pose a loss of generality from the analytic point of view, since the transform $u \mapsto \exp(\alpha t)\tilde{u}$ leads to an equivalent problem for $\tilde{u}$ with reaction term $c + \alpha\exp(\alpha t)$, such that (2) is satisfied whenever α is sufficiently large. In practice, however, such a transform is usually not applied. In addition, due to round-off errors coming from floating point arithmetics, numerical solutions obtained with the original and transformed discrete problem are probably not longer equivalent.

In the literature, one can find meanwhile several works on the numerical analysis of ROM methods for the Navier–Stokes equations. In [23], a ROM method with LPS stabilization is introduced and analyzed. The error bounds in [23] are not independent of inverse powers of the viscosity coefficient, or equivalently, not independent of the Reynolds number. The same method as in [23] is analyzed in [19], but avoiding the penalty term for the pressure included in [23] and adding grad-div stabilization. Both methods in [23] and [19] are based on a FOM with non inf-sup stable pairs of finite element spaces. In [19], a ROM method with snapshots based on inf-sup stable elements with grad-div stabilization both for the FOM and the ROM is also analyzed. For the second method, following [16], a supremizer [2, 21] pressure recovery method is applied to get a ROM pressure approximation. The bounds in [19] are independent of inverse powers of the viscosity, and, to the best of our knowledge, it is the only reference with this property for the error bounds so far. Recently, a POD reduced order Variational Multiscale (VMS) approach for moderately high Reynolds numbers has been proposed in [25]. There are other types of ROM methods with stabilizations, some of which have been utilized for the numerical simulation of turbulent flows, e.g., see [4]. In this reference, a reduced order closure modeling of Smagorinsky (LES) type has been proposed.

This paper presents an error analysis of the SUPG-ROM for equations of type (1). An essential feature of the method, for its analysis, compare Remark 3.2 for an explanation, is that the set of snapshots does not only contain the functions at the time instants but also approximations of the temporal derivative. Then, robust estimates for the error of the SUPG-ROM solution to the weak solution of (1) are derived, i.e., the constants in the error bounds do not blow up as $\varepsilon \to 0$. To the best of our knowledge, this is the first time in the literature that this kind of bounds is proved for POD models. Thus, this paper can be considered as an analytic support of [10], where choosing the stabilization parameter in the same way as in the corresponding SUPG FEM was found to be the advisable choice for the SUPG ROM simulations. The analysis is based on an appropriate decomposition of the error. We compare the POD approximation with the projection of the full order approximation on the POD space. In many references in the literature, the POD approximation is compared with the projection of the corresponding weak solution of (1) instead. Comparing with the projection of the FOM SUPG approximation, one can simplify the error equation so that it is possible to reproduce essentially the analysis in [15] to bound the error in the POD approximation. The final error bounds are a sum of the error from the FOM and additional terms in which the sum of the neglected eigenvalues from the POD technique appears as a factor.

The paper is organized as follows. Section 2 introduces the SUPG-ROM method. The analysis of this method is presented in Section 3. Then, Section 4 contains numerical studies with this method and finally, Section 5 provides a summary and an outlook.

2 The SUPG-ROM method

Throughout this paper, standard notations are used for Lebesgue and Sobolev spaces. Generic constant that do not depend on the mesh width or the length of the time step are denoted by C.

We will denote by $V_{h,r}$ the finite element space where h indicates the fineness of the underlying triangulation $\mathcal{T}_h$ and r the degree of the local finite element polynomials. Assuming that the meshes are quasi-uniform, the following inverse inequality holds for each $v_h \in V_{h,r}$, e.g., see [5, Theorem 3.2.6],

$$\|v_h\|_{W^{m,q}(K)} \le c_{\mathrm{inv}} h_K^{l-m-d\left(\frac{1}{q'}-\frac{1}{q}\right)} \|v_h\|_{W^{l,q'}(K)}, \tag{3}$$

where $0 \le l \le m \le 1$, $1 \le q' \le q \le \infty$, h_K is the size (diameter) of the mesh cell $K \in \mathcal{T}_h$, and $\|\cdot\|_{W^{m,q}(K)}$ is the norm in $W^{m,q}(K)$.

The SUPG method has the form (time-continuous case): Find $u_h \ : \ (0,T] \to V_{h,r}$ such that

$$\begin{aligned}(\partial_t u_h, v_h) + a_{\mathrm{SUPG}}(u_h, v_h) + \sum_{K\in\mathcal{T}_h} \delta_K (\partial_t u_h, \boldsymbol{b}\cdot\nabla v_h)_K \\ = \ (f, v_h) + \sum_{K\in\mathcal{T}_h} \delta_K (f, \boldsymbol{b}\cdot\nabla v_h)_K \quad \forall\, v_h \in V_{h,r},\end{aligned}$$

with $u_h(0,\boldsymbol{x})$ being an appropriate approximation of $u_0(\boldsymbol{x})$ and

$$\begin{aligned} a_{\mathrm{SUPG}}(u_h, v_h) &= \varepsilon(\nabla u_h, \nabla v_h) + (\boldsymbol{b}\cdot\nabla u_h, v_h) + (c u_h, v_h) \\ &\quad + \sum_{K\in\mathcal{T}_h} \delta_K(-\varepsilon\Delta u_h + \boldsymbol{b}\cdot\nabla u_h + c u_h, \boldsymbol{b}\cdot\nabla v_h)_K. \end{aligned}$$

Here, $\{K \in \mathcal{T}_h\}$ denotes set of mesh cells of the triangulation, $(\cdot,\cdot)_K$ the inner product in $L^2(K)$, and $\{\delta_K\}$ are local parameters that have to be chosen appropriately.

Let (2) be satisfied. If the SUPG parameters are chosen such that

$$\delta_K \le \frac{\mu_0}{2\|c\|_{K,\infty}^2}, \quad \delta_K \le \frac{h_K^2}{2\varepsilon c_{\mathrm{inv}}^2}, \tag{4}$$

then the bilinear form $a_{\mathrm{SUPG}}(\cdot,\cdot)$ associated with the SUPG method satisfies

$$a_{\mathrm{SUPG}}(v_h, v_h) \ge \frac{1}{2}\|v_h\|_{\mathrm{SUPG}}^2, \quad \forall\, v_h \in V_{h,r},$$

with

$$\|v_h\|_{\mathrm{SUPG}} := \left(\varepsilon\|\nabla v_h\|_0^2 + \sum_{K\in\mathcal{T}_h} \delta_K \|\boldsymbol{b}\cdot\nabla v_h\|_{0,K}^2 + \|\mu^{1/2} v_h\|_0^2\right)^{1/2}, \tag{5}$$

e.g., see [20, Part **III**, Lemma 3.25]. We will denote by $\Pi_h u(t) \in V_{h,r}$ the solution of the steady-state problem

$$a_{\mathrm{SUPG}}(\Pi_h u(t), v_h) = a_{\mathrm{SUPG}}(u(t), v_h) \quad \forall\, v_h \in V_{h,r}. \tag{6}$$

Next, a fully discrete SUPG scheme will be considered, with the backward Euler method as time integrator and fixed time step τ, which is chosen such that $T = M\tau$. The fully discrete approximation at time $t^n = n\tau$ is denoted by u_h^n. With the notation $u_{h,\tau}^n = (u_h^n - u_h^{n-1})/\tau$ for $n \geq 1$, the fully discrete scheme reads as follows: Find $u_h^n \in V_{h,r}$ such that

$$\begin{aligned}(u_{h,\tau}^n, v_h) + a_{\mathrm{SUPG}}(u_h^n, v_h) &+ \sum_{K\in\mathcal{T}_h} \delta_K(u_{h,\tau}^n, \boldsymbol{b}\cdot\nabla v_h)_K \\ = \quad (f^n, v_h) + \sum_{K\in\mathcal{T}_h} &\delta_K(f^n, \boldsymbol{b}\cdot\nabla v_h)_K \quad \forall\, v_h \in V_{h,r}.\end{aligned} \tag{7}$$

Assume (2),

$$\boldsymbol{b}(t,\boldsymbol{x}) = \boldsymbol{b}(\boldsymbol{x}), \quad \nabla\cdot\boldsymbol{b}(\boldsymbol{x}) = 0, \quad c(t,\boldsymbol{x}) = c(\boldsymbol{x}), \tag{8}$$

that the mesh is uniform with mesh width h, and that the stabilization parameters are the same for all mesh cells, i.e., $\delta_K = \delta$. Also, consider only the convection-dominated regime, i.e., ε is sufficiently small in comparison with the mesh width. Let the stabilization parameter defined to be

$$\delta = \min\left\{\frac{h}{4c_{\mathrm{inv}}\|\boldsymbol{b}\|_{L^\infty}} \min\left\{\frac{1}{2}, \frac{\mu_0}{4\|c\|_{L^\infty}}, \frac{\mu_0^{1/2}}{\|c\|_{L^\infty}^{1/2}}, \frac{\|\boldsymbol{b}\|_{L^\infty} h}{4\varepsilon c_{\mathrm{inv}}}\right\}, \frac{1}{\mu_0}, \frac{1}{\|c\|_{L^\infty}}\right\}. \tag{9}$$

Then, the following error estimate was derived in [15, Theorem 5.3] (see also [9, Theorem 3.3].

$$\|u(t_n) - u_h^n\|_0^2 + \tau\sum_{j=1}^{n} \|u(t_j) - u_h^j\|_{\mathrm{SUPG}}^2 \leq C\left(h^{2r+1} + \tau^2\right). \tag{10}$$

The constant C does not depend on inverse powers of the diffusion coefficient ε.

From the finite element solution, a basis for the ROM will be computed via a proper orthogonal decomposition (POD) method. To this end, snapshots of the finite element solution and of the approximation of its temporal derivative are considered. We study the case that the snapshots are taken in each time instant. Thus, consider the following space

$$\mathcal{V} = \text{span}\left\{y^1, \ldots, y^N\right\},$$

with $N = 2M + 1$. $y^j = u_h^j$, $j = 0, \ldots, M$, and $y^{j+M+1} = u_{h,\tau}^j$, $j = 1, \ldots, M$. A construction of the space of snapshots in this form can be found, e.g., in [18]. It is clear that the last M functions belong to the span of the first $M+1$ functions, because the finite difference approximations of the temporal derivative are linear combinations of the finite element solution at two subsequent time instants. However, as pointed out in [18], the derived POD basis differs generally depending on whether the approximations of the temporal derivative are contained in the snapshots or not, compare also Section 4, where it will be shown that one obtains usually different results with the corresponding ROM simulations.

Let $\mathbb{K} = (k_{i,j})_{i,j=1}^N \in \mathbb{R}^{N\times N}$ be the correlation matrix corresponding to the snapshots with

$$k_{i,j} = \frac{1}{N}(y^i, y^j).$$

Following [18], we denote by $\lambda_1 \geq \lambda_2, \ldots \geq \lambda_p > 0$ the positive eigenvalues of $\mathbb{K}$ and by $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p \in \mathbb{R}^N$ the associated eigenvectors. Then, the orthonormal POD basis of $\mathcal{V}$ is given by

$$\psi_k = \frac{1}{\sqrt{N}} \frac{1}{\sqrt{\lambda_k}} \sum_{j=1}^N v_k^j y^j, \quad k = 1, \ldots, p, \tag{11}$$

where v_k^j is the j-th component of the eigenvector $\boldsymbol{v}_k$. The following error formula holds, see [18, Proposition 1],

$$\frac{1}{N}\sum_{j=1}^N \left\| y^j - \sum_{k=1}^l (y^j, \psi_k)\psi_k \right\|_0^2 = \sum_{k=l+1}^p \lambda_k, \quad l \leq p. \tag{12}$$

Denoting by $\mathbb{S} = (s_{ij})_{i,j=1}^p = (\nabla\psi_j, \nabla\psi_i)_{i,j=1}^p \in \mathbb{R}^{p\times p}$ the stiffness matrix for the POD basis, then for any $v \in \mathcal{V}$ the following inverse inequality holds, see [18, Lemma 2, Remark 2],

$$||\nabla v||_0 \leq \sqrt{\|\mathbb{S}\|_2}\|v\|_0, \tag{13}$$

where $\|\cdot\|_2$ denotes the spectral norm of a matrix.

From the inverse inequality (13) and (12), one obtains

$$\begin{aligned} &\frac{1}{N}\sum_{j=1}^N \left\| \nabla y^j - \sum_{k=1}^l (y^j, \psi_k)\nabla\psi_k \right\|_0^2 \\ \leq\ & \frac{\|\mathbb{S}\|_2}{N}\sum_{j=1}^N \left\| y^j - \sum_{k=1}^l (y^j, \psi_k)\psi_k \right\|_0^2 \leq \|\mathbb{S}\|_2 \sum_{k=l+1}^p \lambda_k. \end{aligned} \tag{14}$$

Instead of (14), the following result that is taken from [14, Lemma 3.2] or [10, Lemma 3.2] can be also applied

$$\frac{1}{N}\sum_{j=1}^N \left\| \nabla y^j - \sum_{k=1}^l (y^j, \psi_k)\nabla\psi_k \right\|_0^2 = \sum_{k=l+1}^p \lambda_k \|\nabla\psi_k\|_0^2.$$

Let $\mathcal{V}_l = \text{span}\,\{\psi_1, \psi_2, \ldots, \psi_l\}$ and denote by P_l the L^2-orthogonal projection onto $\mathcal{V}_l$.

3 Analysis of the SUPG-ROM method

The SUPG reduced order model based on orthogonal decomposition approximation reads as follows: Find $u_l \in \mathcal{V}_l$ such that for $n \geq 1$

$$\begin{aligned}(u_{l,\tau}^n, v_l) + a_{\mathrm{SUPG}}(u_l^n, v_l) + \sum_{K\in\mathcal{T}_h} \delta_K(u_{l,\tau}^n, \boldsymbol{b}\cdot\nabla v_l)_K \\ = (f^n, v_l) + \sum_{K\in\mathcal{T}_h} \delta_K(f^n, \boldsymbol{b}\cdot\nabla v_l)_K \quad \forall\, v_l \in \mathcal{V}_l, \end{aligned} \tag{15}$$

with $u_{l,\tau}^n = (u_l^n - u_l^{n-1})/\tau$. As initial condition, $u_l^0 = P_l u^0$ is taken.

Theorem 3.1 (Error estimate: $L^2(\Omega)$ and discrete in $L^2(0,T)$.) *Assume that $u, \partial_t u \in L^\infty((0,T);H^{r+1}(\Omega))$ and let $\Pi_h\partial_{tt}u, \Pi_h\partial_{ttt}u, \partial_{ttt}u \in L^2((0,T);L^2(\Omega))$, where Π_h is the projection defined in* (6)*. Let the conditions* (8) *and* (2) *be satisfied and consider the convection-dominated regime with the assumption*

$$\varepsilon \leq \frac{\|\boldsymbol{b}\|_{L^\infty}}{c_{\mathrm{inv}}} h. \tag{16}$$

Then, there exists a constant C, independent of ε, such that it holds for $n\tau \leq T$

$$\begin{aligned}\sum_{j=1}^n \tau\|u^j - u_l^j\|_0^2 \leq CT\Bigg[& h^{2r+1} + \tau^2 + \|e_l^0\|_0^2 \\ & + \left((\varepsilon + \|\boldsymbol{b}\|_{L^\infty}^2)\|\mathbb{S}\|_2 + \|c\|_{L^\infty}^2 + 1\right)\sum_{k=l+1}^p \lambda_k\Bigg], \end{aligned} \tag{17}$$

where $e_l^0 = u_l^0 - P_l u_h^0$, with u_h^0 being the finite element initial condition.

Proof In the error analysis, the difference of the SUPG-ROM solution u_l^n and the projection of the finite element solution into the POD space $P_l u_h^n$ is estimated. To this end, the definition of the L^2 projection and (7) yields

$$\begin{aligned}(P_l u_{h,\tau}^n, v_l) + a_{\mathrm{SUPG}}(P_l u_h^n, v_l) + \sum_{K\in\mathcal{T}_h}\delta_K(P_l u_{h,\tau}^n, \boldsymbol{b}\cdot\nabla v_l)_K \\ = (f^n, v_l) + \sum_{K\in\mathcal{T}_h}\delta_K(f^n, \boldsymbol{b}\cdot\nabla v_l)_K + a_{\mathrm{SUPG}}(P_l u_h^n - u_h^n, v_l) \\ + \sum_{K\in\mathcal{T}_h}\delta_K(P_l u_{h,\tau}^n - u_{h,\tau}^n, \boldsymbol{b}\cdot\nabla v_l)_K. \end{aligned} \tag{18}$$

Let us use the following notations

$$e_l^n = u_l^n - P_l u_h^n, \quad \eta_h^n = u_h^n - P_l u_h^n.$$

Subtracting (18) from (15) gives

$$
\begin{aligned}
& (e_{l,\tau}^n, v_l) + a_{\mathrm{SUPG}}(e_l^n, v_l) + \sum_{K\in\mathcal{T}_h} \delta_K(e_{l,\tau}^n, \boldsymbol{b}\cdot\nabla v_l)_K \\
= \; & a_{\mathrm{SUPG}}(\eta_h^n, v_l) + \sum_{K\in\mathcal{T}_h} \delta_K(\eta_{h,\tau}^n, \boldsymbol{b}\cdot\nabla v_l)_K. \qquad (19)
\end{aligned}
$$

To bound the error, we follow at the beginning the steps of the proof of (10) that can be found in [15, Section 5.2]. Now, the assumption of a constant stabilization parameter is used and we denote

$$\|v_h\|_{\mathrm{mat}} = \delta^{1/2}\|v_{h,\tau} + \boldsymbol{b}\cdot\nabla v_h\|_0.$$

Taking in (19) first $v_l = e_l^n$ and then $v_l = \delta e_{l,\tau}^n$ and finally adding these equations leads to

$$
\begin{aligned}
& (e_{l,\tau}^n, e_l^n) + \delta^2(\boldsymbol{b}\cdot\nabla e_l^n, \boldsymbol{b}\cdot\nabla e_{l,\tau}^n) + \varepsilon\|\nabla e_l^n\|_0^2 + \|\mu^{1/2} e_l^n\|_0^2 + \|e_l^n\|_{\mathrm{mat}}^2 \\
& \quad + \varepsilon\delta(\nabla e_l^n, \nabla e_{l,\tau}^n) + \delta(c e_l^n, e_{l,\tau}^n) \\
= \; & a_{\mathrm{SUPG}}(\eta_h^n, e_l^n + \delta e_{l,\tau}^n) + \delta(\eta_{h,\tau}^n, \boldsymbol{b}\cdot\nabla(e_l^n + \delta e_{l,\tau}^n)) \\
& -\delta(c e_l^n, \boldsymbol{b}\cdot\nabla(e_l + \delta e_{l,\tau}^n)) + \sum_{K\in\mathcal{T}_h} \delta\varepsilon\left(\Delta e_l^n, \boldsymbol{b}\cdot\nabla(e_l^n + \delta e_{l,\tau}^n)\right)_K, \qquad (20)
\end{aligned}
$$

where we have taken into account that due to the condition $\nabla\cdot\boldsymbol{b}(\boldsymbol{x}) = 0$ the term $\delta^2(e_{l,\tau}^n, \boldsymbol{b}\cdot\nabla e_{l,\tau}^n)$ vanishes. In the analysis of [15], the same analysis as for the continuous-in-time case is applied at this stage and then truncation errors with respect to time have to be bounded.

The four terms on the right-hand side of (20) have to be bounded. For this purpose, the following estimate is used, which is derived by using the definition of the material derivative and the definition of the stabilization parameter (9)

$$
\begin{aligned}
\|\delta e_{l,\tau}^n\|_0 & \le \|\delta(e_{l,\tau}^n + \boldsymbol{b}\cdot\nabla e_l^n)\|_0 + \|\delta\boldsymbol{b}\cdot\nabla e_l^n\|_0 \\
& \le \delta^{1/2}\|e_l^n\|_{\mathrm{mat}} + \delta\|\boldsymbol{b}\|_{L^\infty} c_{\mathrm{inv}} h^{-1}\mu_0^{-1/2}\|\mu^{1/2} e_l^n\|_0 \\
& \le \delta^{1/2}\|e_l^n\|_{\mathrm{mat}} + \frac{\mu_0^{-1/2}}{8}\|\mu^{1/2} e_l^n\|_0. \qquad (21)
\end{aligned}
$$

For the first term on the right-hand side of (20), we obtain in a first step, by using the Cauchy–Schwarz inequality, Hölder's inequality, the inverse inequality (3), the first line of estimate (21), and also the last line of the same estimate

$$\begin{aligned}
&a_{\mathrm{SUPG}}(\eta_h^n, e_l^n + \delta e_{l,\tau}^n)\\
&\quad\le\ \varepsilon^{1/2}\|\nabla\eta_h^n\|_0\left(\varepsilon^{1/2}\|\nabla e_l^n\|_0 + c_{\mathrm{inv}}h^{-1}\delta^{1/2}\varepsilon^{1/2}\|e_l^n\|_{\mathrm{mat}} + c_{\mathrm{inv}}h^{-1}\delta\|\boldsymbol{b}\|_{L^\infty}\varepsilon^{1/2}\|\nabla e_l^n\|_0\right)\\
&\qquad + (\|\boldsymbol{b}\|_{L^\infty}\|\nabla\eta_h^n\|_0 + \|c\|_{L^\infty}\|\eta_h^n\|_0))\\
&\qquad\quad \times\left(\mu_0^{-1/2}\|\mu^{1/2}e_l^n\|_0 + \delta^{1/2}\|e_l^n\|_{\mathrm{mat}} + \frac{\mu_0^{-1/2}}{8}\|\mu^{1/2}e_l^n\|_0\right)\\
&\qquad +\delta\left[\left(\sum_{K\in\mathcal{T}_h}\varepsilon^2\|\Delta\eta_h^n\|_{0,K}^2\right)^{1/2} + \left(\sum_{K\in\mathcal{T}_h}\|\boldsymbol{b}\|_{L^\infty}^2\|\nabla\eta_h^n\|_{0,K}^2\right)^{1/2} + \left(\sum_{K\in\mathcal{T}_h}\|c\|_{L^\infty}^2\|\eta_h^n\|_{0,K}^2\right)^{1/2}\right]\\
&\qquad\quad \times\|\boldsymbol{b}\|_{L^\infty}c_{\mathrm{inv}}h^{-1}\left(\mu_0^{-1/2}\|\mu^{1/2}e_l^n\|_0 + \delta^{1/2}\|e_l^n\|_{\mathrm{mat}} + \frac{\mu_0^{-1/2}}{8}\|\mu^{1/2}e_l^n\|_0\right).
\end{aligned}$$

Using now the inverse inequality (3) and the definition (9) of the stabilization parameter leads to

$$\begin{aligned}
\delta\left(\sum_{K\in\mathcal{T}_h}\varepsilon^2\|\Delta\eta_h^n\|_{0,K}^2\right)^{1/2}\|\boldsymbol{b}\|_{L^\infty}c_{\mathrm{inv}}h^{-1} &\le \delta\varepsilon h^{-1}c_{\mathrm{inv}}\|\nabla\eta_h^n\|_0\|\boldsymbol{b}\|_{L^\infty}c_{\mathrm{inv}}h^{-1}\\
&\le \frac{1}{16}\|\boldsymbol{b}\|_{L^\infty}\|\nabla\eta_h^n\|_0,\\
\delta\left(\sum_{K\in\mathcal{T}_h}\|\boldsymbol{b}\|_{L^\infty}^2\|\nabla\eta_h^n\|_{0,K}^2\right)^{1/2}\|\boldsymbol{b}\|_{L^\infty}c_{\mathrm{inv}}h^{-1} &\le \frac{1}{8}\|\boldsymbol{b}\|_{L^\infty}\|\nabla\eta_h^n\|_0,\\
\delta\left(\sum_{K\in\mathcal{T}_h}\|c\|_{L^\infty}^2\|\eta_h^n\|_{0,K}^2\right)^{1/2}\|\boldsymbol{b}\|_{L^\infty}c_{\mathrm{inv}}h^{-1} &\le \frac{1}{8}\|c\|_{L^\infty}\|\eta_h^n\|_0.
\end{aligned}$$

Now, Young's inequality and the definition (9) of the stabilization parameter are applied, which gives

$$\begin{aligned}
&a_{\mathrm{SUPG}}(\eta_h^n, e_l^n + \delta e_{l,\tau}^n)\\
&\quad\le\ C\left((\varepsilon + \|\boldsymbol{b}\|_{L^\infty}^2)\|\nabla\eta_h^n\|_0^2 + \|c\|_{L^\infty}^2\|\eta_h^n\|_0^2\right)\\
&\qquad +\frac{1}{8}\left(\varepsilon\|\nabla e_l^n\|_0^2 + \|\mu^{1/2}e_l^n\|_0^2 + \|e_l^n\|_{\mathrm{mat}}^2\right).
\end{aligned}\tag{22}$$

Applying the same analytic tools leads for the second term on the right-hand side of (20) to the estimate

$$\begin{aligned}
&\delta(\eta_{h,\tau}^n, \boldsymbol{b}\cdot\nabla(e_l^n + \delta e_{l,\tau}^n))\\
&\quad\le\ \delta\|\eta_{h,\tau}^n\|_0\|\boldsymbol{b}\|_{L^\infty}c_{\mathrm{inv}}h^{-1}\left(\mu_0^{-1/2}\|\mu^{1/2}e_l^n\|_0 + \delta^{1/2}\|e_l^n\|_{\mathrm{mat}} + \frac{\mu_0^{-1/2}}{8}\|\mu^{1/2}e_l^n\|_0\right)\\
&\quad\le\ C\|\eta_{h,\tau}^n\|_0^2 + \frac{1}{16}\left(\|\mu^{1/2}e_l^n\|_0^2 + \|e_l^n\|_{\mathrm{mat}}^2\right)
\end{aligned}\tag{23}$$

and for the third term on the right-hand side of (20) to

$$\begin{aligned}
&\delta(ce_l^n, \boldsymbol{b}\cdot\nabla(e_l^n+\delta e_{l,\tau}^n))\\
&\quad\le\ \delta\|c\|_\infty\mu_0^{-1/2}\|\mu^{1/2}e_l^n\|_0\|\boldsymbol{b}\|_{L^\infty}c_{\mathrm{inv}}h^{-1}\\
&\qquad\times\left(\mu_0^{-1/2}\|\mu^{1/2}e_l^n\|_0+\delta^{1/2}\|e_l^n\|_{\mathrm{mat}}+\frac{\mu_0^{-1/2}}{8}\|\mu^{1/2}e_l^n\|_0\right)\\
&\quad\le\ \frac{9}{128}\|\mu^{1/2}e_l^n\|_0^2+\frac{1}{16}\|\mu^{1/2}e_l^n\|_0\|e_l^n\|_{\mathrm{mat}}\\
&\quad\le\ \frac{13}{128}\|\mu^{1/2}e_l^n\|_0^2+\frac{1}{32}\|e_l^n\|_{\mathrm{mat}}^2.
\end{aligned}\tag{24}$$

For bounding the fourth term on the right-hand side of (20), the triangle inequality and also assumption (16) is utilized, which gives

$$\begin{aligned}
&\sum_{K\in\mathcal{T}_h}\delta\varepsilon(\Delta e_l^n, \boldsymbol{b}\cdot\nabla(e_l^n+\delta e_{l,\tau}^n))\\
&\le\ \varepsilon\delta c_{\mathrm{inv}}h^{-1}\|\nabla e_l^n\|_0\|\boldsymbol{b}\|_{L^\infty}\left(\|\nabla e_l^n\|_0+c_{\mathrm{inv}}h^{-1}\left(\delta^{1/2}\|e_l^n\|_{\mathrm{mat}}+\delta\|\boldsymbol{b}\|_{L^\infty}\|\nabla e_l^n\|_0\right)\right)\\
&\le\ \frac{1}{8}\varepsilon\|\nabla e_l^n\|_0^2+\frac{c_{\mathrm{inv}}}{8h}\delta^{1/2}\varepsilon\|\nabla e_l^n\|_0\|e_l^n\|_{\mathrm{mat}}+\frac{1}{64}\varepsilon\|\nabla e_l^n\|_0^2\\
&\le\ \frac{9}{64}\varepsilon\|\nabla e_l^n\|_0^2+\frac{1}{32}\varepsilon^{1/2}\|\nabla e_l^n\|_0\|e_l^n\|_{\mathrm{mat}}\\
&\le\ \frac{5}{32}\varepsilon\|\nabla e_l^n\|_0^2+\frac{1}{64}\|e_l^n\|_{\mathrm{mat}}^2.
\end{aligned}\tag{25}$$

Inserting (22), (23), (24), and (25) in (20) yields

$$\begin{aligned}
&(e_{l,\tau}^n, e_l^n)+\delta^2(\boldsymbol{b}\cdot\nabla e_l^n, \boldsymbol{b}\cdot\nabla e_{l,\tau}^n)+\frac{1}{2}\varepsilon\|\nabla e_l^n\|_0^2+\frac{1}{2}\|\mu^{1/2}e_l^n\|_0^2+\frac{1}{2}\|e_l^n\|_{\mathrm{mat}}^2\\
&\quad+\varepsilon\delta(\nabla e_l^n, \nabla e_{l,\tau}^n)+\delta(ce_l^n, e_{l,\tau}^n)\\
&\le\ C\left(\left(\varepsilon+\|\boldsymbol{b}\|_{L^\infty}^2\right)\|\nabla\eta_h^n\|_0^2+\|c\|_{L^\infty}^2\|\eta_h^n\|_0^2\right)+C\|\eta_{h,\tau}^n\|_0^2,
\end{aligned}\tag{26}$$

with the constants being independent of ε. Summation over the first n time instants, multiplying with τ, and observing that $\mu(\boldsymbol{x})=c(\boldsymbol{x})$ leads to the inequality

$$\begin{aligned}
&\|e_l^n\|_0^2+\delta\left(\varepsilon\|\nabla e_l^n\|_0^2+\|\mu^{1/2}e_l^n\|_0^2+\delta\|\boldsymbol{b}\cdot\nabla e_l^n\|_0^2\right)\\
&\qquad+\sum_{j=1}^n\tau\left(\varepsilon\|\nabla e_l^j\|_0^2+\|\mu^{1/2}e_l^j\|_0^2+\|e_l^j\|_{\mathrm{mat}}^2\right)\\
&\le\ \|e_l^0\|_0^2+\delta^2\|\boldsymbol{b}\cdot\nabla e_l^0\|_0^2+\varepsilon\delta\|\nabla e_l^0\|_0^2+\delta\|\mu^{1/2}e_l^0\|_0^2\\
&\qquad+C\sum_{j=1}^n\tau\left((\varepsilon+\|\boldsymbol{b}\|_{L^\infty}^2)\|\nabla\eta_h^j\|_0^2+\|c\|_{L^\infty}^2\|\eta_h^j\|_0^2\right)+C\sum_{j=1}^n\tau\|\eta_{h,\tau}^j\|_0^2.
\end{aligned}\tag{27}$$

For the terms at the initial time, one obtains with Hölder's inequality, the definition (9) of the stabilization parameter, and assumption (16)

$$\|e_l^0\|_0^2+\delta^2\|\boldsymbol{b}\cdot\nabla e_l^0\|_0^2+\varepsilon\delta\|\nabla e_l^0\|_0^2+\delta\|\mu^{1/2}e_l^0\|_0^2\le C\|e_l^0\|_0^2.$$

Applying now (12), which is possible by the definitions of η_h^j and P_l, and (14), thereby observing that the finite differences approximations of the time derivative are included in the set of snapshots, and using that $\tau \leq C/N$, yields

$$\|e_l^n\|_0^2 \leq C\|e_l^0\|_0^2 + C\left((\varepsilon + \|\boldsymbol{b}\|_{L^\infty}^2)\|\mathbb{S}\|_2 + \|c\|_{L^\infty}^2 + 1\right)\sum_{k=l+1}^{p} \lambda_k. \tag{28}$$

Since

$$u^n - u_l^n = (u^n - u_h^n) + (u_h^n - P_l u_h^n) + (P_l u_h^n - u_l^n), \tag{29}$$

applying the triangle inequality, (12), and (10) leads finally to the estimate given in Theorem 3.1. □

Remark 3.2 (To the proof of Theorem 3.1.)

- *The proof of Theorem 3.1 derives a bound of the error between $\boldsymbol{u}_l^n$ and $P_l\boldsymbol{u}_h^n$. Since P_l is, by definition, the orthogonal projection onto the space $\mathcal{V}_l$, one can write in equation* (18)*, for the first term on the left-hand side $(P_l\boldsymbol{u}_{h,\tau}^n, v_l)$ instead of $(\boldsymbol{u}_{h,\tau}^n, v_l)$, since both terms are equal. However, the last term on the right-hand side of* (18) *does not vanishes. To bound this term, we introduced the subset of snapshots of the finite difference approximations of the time derivative. In case one does not add these snapshots, one can bound the error coming from the last term in* (18) *by a term that contains the factor $(\Delta t)^{-1}$, so that the error bound becomes worse, see analogous comments made in [18, Remark 1].*
- *Observe that the bound of the term $\|e_l^n\|_0^2$ in* (28) *depends on the tail of the eigenvalues but not on the fineness of the temporal and spatial discretizations. This situation is in contrast to the final error bound* (17)*, where the parameters of the discretizations appear with the same powers as in the bound* (10) *for the original method. The result* (28) *can be proved due to the fact that we have compared the POD approximation with the projection of the SUPG approximation instead of the projection of the analytic solution as it is often done in the literature. The same idea was recently applied in [19].*
- *The assumption of using the same time steps in the FOM and the SUPG-ROM is used by proceeding from* (26) *to* (27)*, and then from* (27) *to* (28)*. In the first of these steps, one has to sum over all time instants of the SUPG-ROM method to have the telescoping sum property and in the second step,* (12) *and* (14) *can be applied only for the snapshots, which come from the FOM simulation.*

As a corollary of Theorem 3.4, one can derive a pointwise in time error estimate in the $L^2(\Omega)$ norm. To this end, we use a lemma whose proof can be found in [17, Lemma 3.6] together with the fact that the finite differences approximating the time derivatives are part of the set of snapshots. As it is explained in [17], the possibility of proving pointwise in time error estimates is one of the advantages of increasing the set of snapshots with the temporal difference quotients.

Lemma 3.3 *Let $T > 0$, Z be a normed space, $\{z^n\}_{n=0}^M \subset Z$, and $\tau = T/M$. Then,*

$$\max_{0\le k\le M} \|z^k\|_Z^2 \le C\left(\frac{1}{2M+1}\sum_{n=0}^M \|z^n\|_Z^2 + \frac{1}{2M+1}\sum_{n=1}^M \|z_\tau^n\|_Z^2\right),$$

where $C = 6\max\{1, T^2\}$ and $z_\tau^n = (z^n - z^{n-1})/\tau$ for $n = 1, \ldots, M$.

Theorem 3.4 (Error estimate: $L^2(\Omega)$ and discrete in $L^\infty(0,T)$.) *Let the assumptions of Theorem 3.1 be satisfied. Then, there exists a constant C, independent of ε, such that it holds for $n\tau \le T$*

$$\begin{aligned}\max_{1\le j\le M} \|u^j - u_l^j\|_0^2 \ \le\ & C\Bigg[h^{2r+1} + \tau^2 + \|e_l^0\|_0^2 \\ & + \left((\varepsilon + \|\boldsymbol{b}\|_{L^\infty}^2)\|\mathbb{S}\|_2 + \|c\|_{L^\infty}^2 + 1\right)\sum_{k=l+1}^p \lambda_k\Bigg].\end{aligned} \quad (30)$$

Proof Starting from (29) and utilizing (28) and (10), the only term left to be bounded is

$$\max_{1\le j\le M} \|u_h^j - P_l u_h^j\|_0^2.$$

To this end, we apply Lemma 3.3 with $z^n = u_h^n - P_l u_h^n$ and $Z = L^2(\Omega)$ together with (12) and then the statement of the theorem is reached. □

Corollary 3.5 (Error estimate: SUPG norm in space and discrete in $L^2(0,T)$.) *Let the assumption of Theorem 3.1 be satisfied, then it holds*

$$\begin{aligned}&\sum_{j=1}^n \tau\|u^j - u_l^j\|_{\mathrm{SUPG}}^2 \\ &\le\ C\Bigg[h^{2r+1} + \tau^2 + T\left((\varepsilon + \delta\|\boldsymbol{b}\|_{L^\infty}^2)\|\mathbb{S}\|_2 + \|c\|_{L^\infty}\right)\sum_{k=l+1}^p \lambda_k \\ &\quad + T\left(\varepsilon\|\mathbb{S}\|_2 + \delta\|\boldsymbol{b}\|_{L^\infty}^2\|\mathbb{S}\|_2 + \|c\|_{L^\infty}\right) \\ &\quad \times\left(\|e_l^0\|_0^2 + \left((\varepsilon + \|\boldsymbol{b}\|_{L^\infty}^2)\|\mathbb{S}\|_2 + \|c\|_{L^\infty}^2 + 1\right)\sum_{k=l+1}^p \lambda_k\right)\Bigg],\end{aligned} \quad (31)$$

where the constant is independent of ε.

Proof With the definition (5) of the SUPG norm and (13), one obtains

$$\|e_l^n\|_{\mathrm{SUPG}}^2 \le \left(\varepsilon\|\mathbb{S}\|_2 + \delta\|\boldsymbol{b}\|_{L^\infty}^2\|\mathbb{S}\|_2 + \|c\|_{L^\infty}\right)\|e_l^n\|_0^2.$$

Using (28) and $n\tau \le T$ gives

$$\begin{aligned}\sum_{j=1}^{n} \tau \|e_l^j\|_{\mathrm{SUPG}}^2 \quad \le \quad & CT\left(\varepsilon\|\mathbb{S}\|_2 + \delta\|\boldsymbol{b}\|_{L^\infty}^2\|\mathbb{S}\|_2 + \|c\|_{L^\infty}\right)\\ & \times\left(\|e_l^0\|_0^2 + \left((\varepsilon + \|\boldsymbol{b}\|_{L^\infty}^2)\|\mathbb{S}\|_2 + \|c\|_{L^\infty}^2 + 1\right)\sum_{k=l+1}^{p}\lambda_k\right).\end{aligned}$$

Applying the decomposition (29) and utilizing the triangle inequality, (10), (5), (13), and (12) finishes the proof of this theorem. □

Remark 3.6 *The error analysis for other temporal discretizations, as Crank–Nicolson, can be carried out in a similar way taking into account some technical considerations. On the one hand, one would need a bound analogous to* (10) *for the corresponding temporal discretization. This bound can be proved arguing as in [15, Theorem 5.3]. On the other hand, one can argue as in [18, Theorem 9] to do the corresponding changes to Theorem 3.1. The final error bounds will be of the same principal form as for the backward Euler scheme, with the first order convergence in time replaced by the order of the considered temporal discretization. Since there is no new insight with respect to the ROM contributions in the error bounds and the change with respect to the temporal scheme can be expected, we think that it is not worthwhile to present the this analysis in detail here. But the numerical studies will show some result that were computed with a second order temporal discretization.*

4 Numerical studies

For supporting analytic results, usually an example with prescribed smooth solution (polynomials, sine or cosine functions) is utilized in the literature. However, such an example does not possess layers, which are the most important feature of solutions of convection-diffusion equations from practice. For this reason, we decided to refrain from presenting such an example. Instead, we like to concentrate our numerical studies on an example that models a traveling wave. On the one hand, it has a prescribed analytic solution such that computing errors is easily possible, but on the other hand, the solution has a layer. An example of this form was originally proposed in [12] and a modification was utilized in [10]. This modification will be considered also here.

Let $\Omega = (0,1)^2, T = 1$, and let the coefficients of the convection-diffusion-reaction equation (1) be given by $\varepsilon = 10^{-8}$, $\boldsymbol{b} = (\cos(\pi/3), \sin(\pi/3))^T$, and $c = 1$. The prescribed solution of (1) possesses the analytic form

$$u(t,x,y) = 0.5\,\sin(\pi x)\,\sin(\pi y)\left[\tanh\left(\frac{x+y-t-0.5}{\sqrt{\varepsilon}}\right)+1\right]. \qquad (32)$$

This solution exhibit a moving layer of width $\mathcal{O}\left(\sqrt{\varepsilon}\right)$. The solution at the initial time is presented in Figure 1.

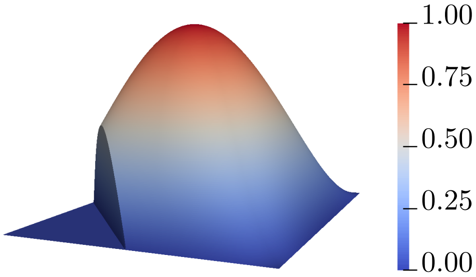

Figure 1: Solution at initial time.

All simulations were performed with the code ParMooN, [8, 26], and all linear systems of equations were solved with the sparse direct solver UMFPACK, [6].

P_1 *finite elements.* First, the case of P_1 finite elements will be studied, i.e., $r = 1$. The simulations were performed on uniform triangular meshes, where the coarsest mesh is constructed by dividing the unit square with a diagonal from $(0,1)$ to $(1,0)$. In the error bounds (17), (30), and (31), which are always for the square of the errors, the impact of the spatial resolution of the FOM appears as the term h^3. Three different meshes were used for the FOM with $h \in \{2.21 \cdot 10^{-2}, 1.10 \cdot 10^{-2}, 5.5 \cdot 10^{-3}\}$ and the corresponding numbers of degrees of freedom (including Dirichlet nodes) are $4\,225$, $16\,641$ and $66\,049$, respectively. All these meshes are too coarse for resolving the layer. The choice of the SUPG stabilization parameter is based on (9). From $\|\boldsymbol{b}\|_{L^\infty} = \max\{\cos(\pi/3), \sin(\pi/3)\} = \sqrt{3}/2$ and $\|c\|_{L^\infty} = \mu_0 = 1$, it follows that

$$\delta = \min\left\{\frac{2h}{4\sqrt{3}c_{\mathrm{inv}}} \min\left\{\frac{1}{4}, \frac{\sqrt{3}h}{8\varepsilon c_{\mathrm{inv}}}\right\}, 1\right\}. \tag{33}$$

An estimate for the parameter c_{inv} for P_1 finite elements was obtained as follows. Using (3) with $l = 0$, $m = 1$ and $d = q = q' = 2$ yields

$$c_{\mathrm{inv}} \geq h \sup_{v_h \in V_h} \frac{\|\nabla v_h\|_0}{\|v_h\|_0}. \tag{34}$$

Then, 10^7 randomly chosen isosceles triangles with right angle and randomly chosen functions v_h were inserted in the right-hand side of (34), from which we found that $c_{\mathrm{inv}} = 8.5$ is an appropriate value. It follows from (33) for the convection-dominated regime that $\delta \approx 8.49 \cdot 10^{-3}h$. This value is much smaller than usually used values, which are of the order $\delta = Ch$ with $C \in [0.1, 1]$. In fact, we could observe large spurious oscillations using this value, although the simulations did not blow up. To be more consistent with the usual practice, and since replacing δ by $C\delta$ with C being a fixed constant has no impact on the numerical analysis as long as C is sufficiently small to respect the upper bounds (4), we used for the simulations 100δ with δ from (33) and c_{inv} as given above, i.e., the SUPG stabilization parameter is approximately $0.849\ h$. Exemplarily, Figure 2 presents a FOM solution at the final time from which one can see that the size of the undershoots is quite small.

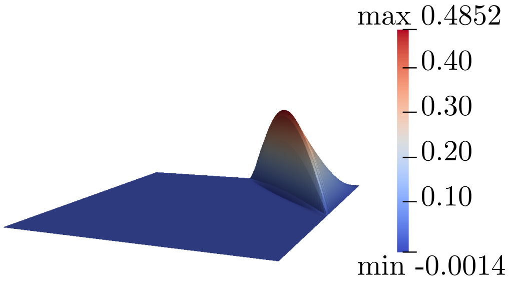

Figure 2: P_1 finite elements. FOM solution at $t = 1.0$ for $h = 1.10 \cdot 10^{-2}$, with SUPG stabilization.

As temporal discretization, both the backward Euler scheme and BDF2 (with the first step to be a backward Euler step) were utilized. For the time step, $\tau = 10^{-4}$ was used. The POD was performed with the snapshots of all time instants. As usual in practice, the POD was applied to fluctuations of the snapshots.

ROM simulations were performed also with the backward Euler and the BDF2 scheme. For $\tau = 10^{-4}$, we found that the results turned out to be practically identical. First, one can conclude that the term with respect to the time step in the error bounds is negligible. This effect was in fact our goal of choosing a very small time step, since the numerical simulations shall study the spatial error of the ROM solutions and not the temporal error. And second, for the sake of brevity, it is sufficient to restrict the presentation of the results to one method. We decided to choose the backward Euler method since it corresponds to the analysis from Section 3. As initial condition in the ROM simulations, the $L^2(\Omega)$ projection of the initial condition of the corresponding FOM simulations was utilized. Hence, $e_l^0 = 0$ and the corresponding term in the error bounds vanishes.

For performing the POD and computing the basis for the ROM simulations, two approaches were pursued. The first one is the standard one, which uses the snapshots of the solution. It will be called *SnapSol*. In the second approach, in addition to the snapshots of the solution, also the snapshots of the approximation of the time derivative were utilized. This approach corresponds to the situation that was analyzed in Section 3 and it will be called *SnapSolTimeDeriv*.

Since the layer is not resolved by the used grid, it appears to the discrete method as a kind of singularity. A consequence is that the order of error reduction with respect to the mesh width for the errors we are interested in is reduced to 0.5. We checked that for larger diffusion coefficients, where the solution does not possess a layer, the optimal orders can be observed.

Results for the norm in $L^2((0,T);L^2(\Omega))$ are presented in Figure 3. These pictures contain the line that corresponds to the error obtained with the FOM solution u_{FOM}, the curve that describes the error between the FOM and the ROM solution $\|u_{\mathrm{FOM}} - u_{\mathrm{ROM}}\|_{L^2((0,T);L^2(\Omega))}$,

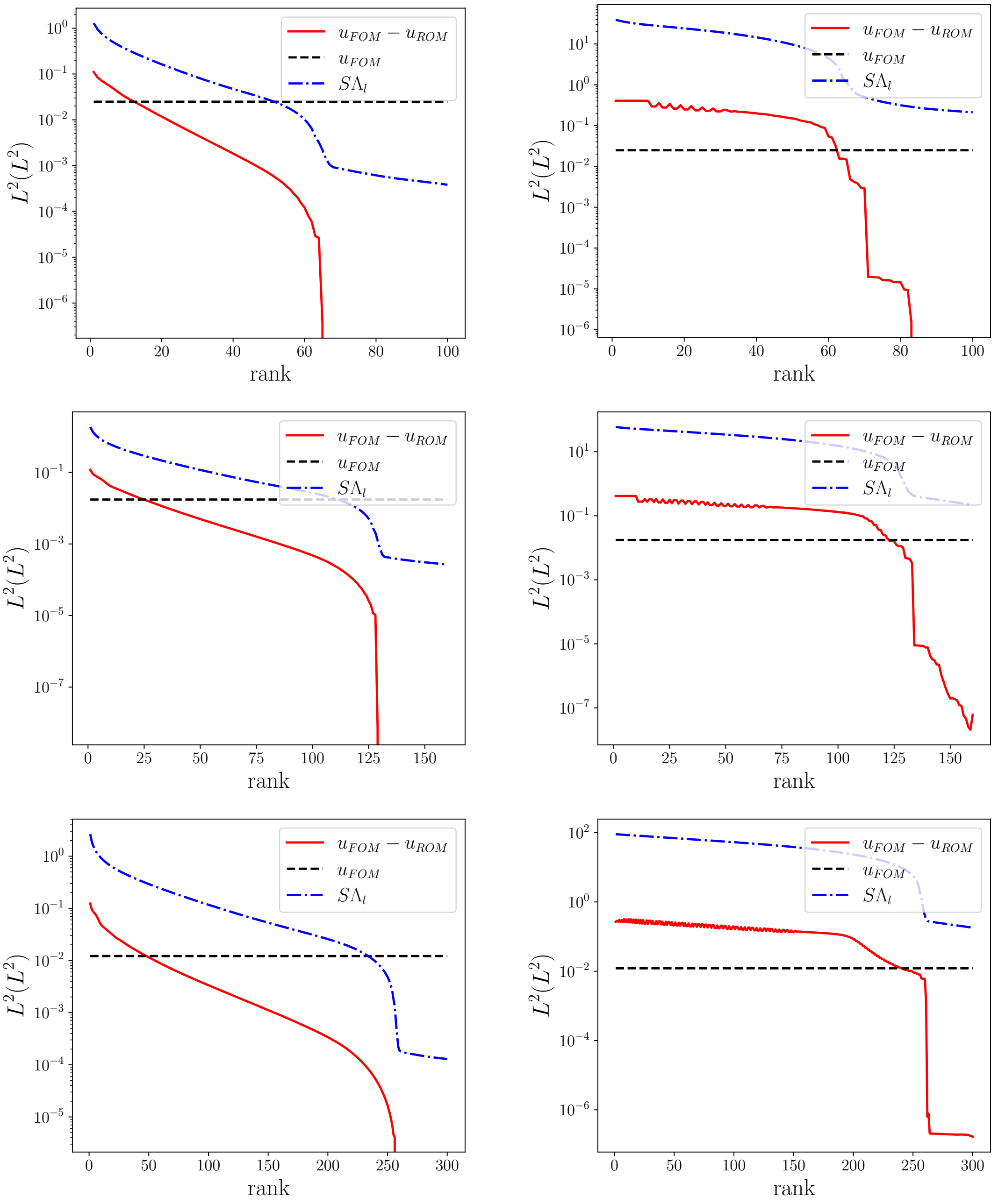

Figure 3: P_1 finite elements. Errors of the ROM solutions in $L^2((0,T);L^2(\Omega))$; top to bottom: meshes with $h \in \{2.21 \cdot 10^{-2}, 1.10 \cdot 10^{-2}, 5.5 \cdot 10^{-3}\}$; left: *SnapSol*; right: *SnapSolTimeDeriv*. Note the different scalings of the abscissas. The notation $u_{\mathrm{FOM}} - u_{\mathrm{ROM}}$ stands for $\|u_{\mathrm{FOM}} - u_{\mathrm{ROM}}\|_{L^2((0,T);L^2(\Omega))}$.

and the curve for the term

$$S\Lambda_l = \left(\left((\varepsilon + \|\boldsymbol{b}\|^2_{L^\infty})\|\mathbb{S}\|_2 + \|c\|^2_{L^\infty} + 1 \right) \sum_{k=l+1}^{p} \lambda_k \right)^{1/2}, \tag{35}$$

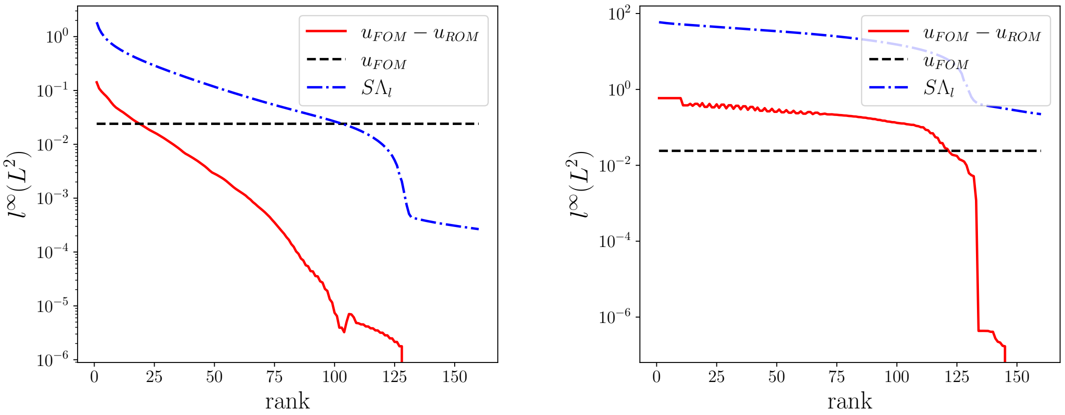

Figure 4: P_1 finite elements. Errors of the ROM solutions in $l^\infty((0,T);L^2(\Omega))$; mesh with $h = 1.10\cdot 10^{-2}$; left: *SnapSol*; right: *SnapSolTimeDeriv*. The notation $u_{\mathrm{FOM}} - u_{\mathrm{ROM}}$ stands for $\|u_{\mathrm{FOM}} - u_{\mathrm{ROM}}\|_{l^\infty((0,T);L^2(\Omega))}$.

which is the last term in the error bound (17). It can be observed that in all cases the ROM solution tends to the FOM solution as the rank of the ROM space increases, since the red lines takes very small values. The finer the FOM simulations, i.e., the more details of the solution are computed, the more basis functions in the ROM are needed to come close to the FOM error, note the different scalings of the abscissas. The convergence is faster for *SnapSol*, i.e., for the standard POD approach, and the errors are smaller. For small numbers of ROM basis functions, an oscillatory behavior of the error obtained with *SnapSolTimeDeriv* can be seen. We think that this behavior is caused by using two subsets of snapshots in the POD that are of different nature: the solution and the temporal derivative. Maybe, the basis functions that are added are determined alternately by these two subsets.

The behavior of the error in $l^\infty((0,T);L^2(\Omega))$ is very similar to the errors in $L^2((0,T);L^2(\Omega))$, compare Figure 4 for an exemplary result.

Figure 5 presents the results for the $L^2((0,T);SUPG)$ error. Again, the principal behavior of the curves is the same as described for the error in $L^2((0,T);L^2(\Omega))$.

A striking feature of all results in Figures 3 – 5 is that the shapes of the red and the blue curves are rather similar. Hence, the term $S\Lambda_l$ from (35) is up to a factor a good approximation of the error $\|u_{\mathrm{FOM}} - u_{\mathrm{ROM},l}\|$ in the respective norm, where the first l POD modes were utilized for computing the ROM solution. A detailed inspection of the results shows that this factor is usually between around 10 and $100 - 300$ as long as the rank of the ROM basis is sufficiently small, which is the standard case in applications. For the error in $L^2((0,T);SUPG)$, this observation allows to obtain an a posteriori error estimate for the ROM solution

$$\begin{aligned}&\|u - u_{\mathrm{ROM,l}}\|_{L^2((0,T);SUPG)}\\ &\quad\leq \quad \|u - u_{\mathrm{FOM}}\|_{L^2((0,T);SUPG)} + \|u_{\mathrm{FOM}} - u_{\mathrm{ROM,l}}\|_{L^2((0,T);SUPG)}. \end{aligned} \tag{36}$$

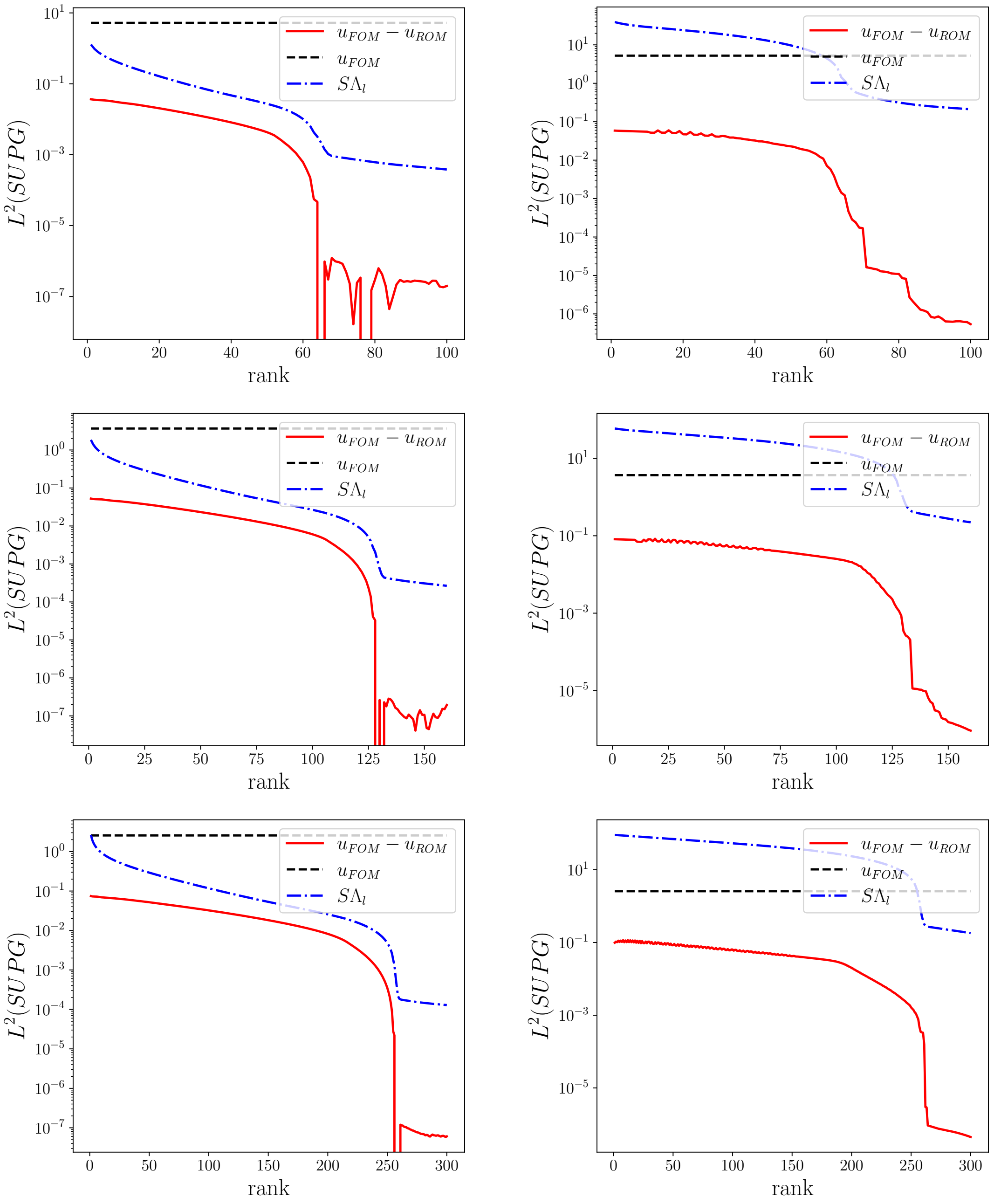

Figure 5: P_1 finite elements. Errors of the ROM solutions in $L^2((0,T);SUPG)$; top to bottom: meshes with $h \in \{2.21 \cdot 10^{-2}, 1.10 \cdot 10^{-2}, 5.5 \cdot 10^{-3}\}$; left: *SnapSol*; right: *SnapSolTimeDeriv*. Note the different scalings of the abscissas. The notation $u_{\mathrm{FOM}} - u_{\mathrm{ROM}}$ stands for $\|u_{\mathrm{FOM}} - u_{\mathrm{ROM}}\|_{L^2((0,T);SUPG)}$.

Concerning the first term on the right-hand side of (36), a robust a posteriori residual-based error estimator for the error in the SUPG norm at each time instant was proposed in [7], under

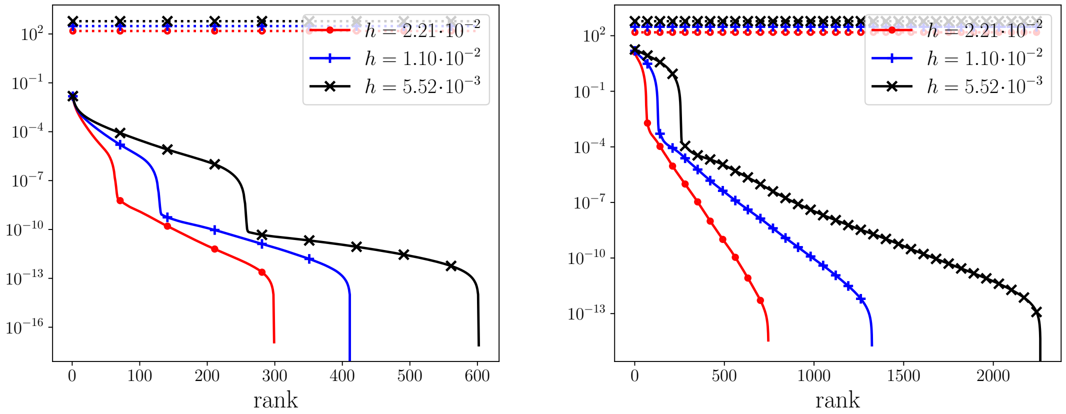

Figure 6: P_1 finite elements. Dotted line with markers: $\|\mathbb{S}\|_2$; solid line with markers: $\sum_{k=l+1}^{p} \lambda_k$; left: *SnapSol*; right: *SnapSolTimeDeriv*.

an assumption which is discussed in detail in this paper. Numerical studies in [7] show that the error is usually overestimated by a factor of magnitude 10, independently of ε since the estimator is robust. In the pre-processing simulation for computing the FOM solution, the first term on the right-hand side of (36) can be approximated by evaluating the error estimator, which is inexpensive. The second term on the right-hand side of (36) can be approximated by $S\Lambda_l$, which gives for the considered example and the common approach *SnapSol* also an overestimation of 10 or even less for a wide range of ranks and independently of the mesh size.

Finally, in order to provide some insight in the term $S\Lambda_l$ from (35), Figure 6 depicts results concerning the terms $\|\mathbb{S}\|_2$ and the tail of the eigenvalues $\sum_{k=l+1}^{p} \lambda_k$, which are contained in $S\Lambda_l$. It can be seen that both terms increase with increasing refinement of the mesh. The term $\|\mathbb{S}\|_2$ is of the order of several hundreds. This order of magnitude corresponds to the orders reported in [19, p. 361] for corresponding matrices in the case of incompressible Navier–Stokes equations. Concerning the tail of the eigenvalues, the qualitative behavior is similar for *SnapSol* and *SnapSolTimeDeriv*, but the quantitative values are much different. The values for *SnapSolTimeDeriv* are larger by at least three orders of magnitude. We could observe that the eigenvalues of *SnapSol* and *SnapSolTimeDeriv* themselves differ by a similar order of magnitude (not presented for brevity). In addition, we observed by performing simulations with $\varepsilon \in \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}\}$ that the tail of the eigenvalues for a fixed rank is usually the higher the smaller the diffusion coefficient is, for both *SnapSol* and *SnapSolTimeDeriv*. Since we think that this behavior can be expected, we like to abstain from a detailed presentation of this result.

P_2 *finite elements.* Numerical studies for P_2 finite elements were performed on the same type of grids as for P_1 finite elements, with $h \in \{2.21 \cdot 10^{-2}, 1.10 \cdot 10^{-2}, 5.5 \cdot 10^{-3}\}$, which led to $16\,641$, $66\,049$ and $263\,169$ degrees of freedom, respectively. BDF2 was used as temporal discretization with the time step $\tau = 10^{-4}$ and a backward Euler step at the beginning. The choice of the SUPG parameter followed the approach that was used for P_1 finite elements. For the constant in the inverse inequality (3), we found that $c_{\text{inv}} = 17.7$ is an appropriate value. With this value, one obtains from (33) a value of around $4 \cdot 10^{-3}h$ and multiplication with 100 leads to $\delta \approx 0.4\,h$, which is of the same order as the parameter used in [10] ($\delta = h$). For P_2 finite elements, the common way of storing the FOM solution only at selected time instants was applied. Concretely, the snapshots were stored after every tenth time step. These approaches will be called *SnapSol10* and *SnapSolTimeDeriv10*.

Results for the ROM error in $L^2((0,T);L^2(\Omega))$ are presented in Figure 7 and examples for some ROM solutions at the final time in Figure 8. Concerning the error, a qualitatively similar behavior can be observed as for P_1 elements, compare Figure 3. In particular, the shapes of the curves for $\|u_{\text{FOM}} - u_{\text{ROM}}\|_{L^2((0,T);L^2(\Omega))}$ and $S\Lambda_l$ are again similar. But there are also a number of differences compared with the P_1 case, which will be discussed next. Whereas for the same number of degrees of freedom, the number of ROM basis functions for obtaining a certain error with respect to the FOM solution is similar for the method that uses only the snapshots, considerably more basis functions are needed for the other method. The curves for $S\Lambda_l$ decrease slower for larger ranks. And finally, the curves for *SnapSolTimeDeriv10* are less oscillatory for small ranks. Figure 8 shows that increasing the rank does not only usually reduces the error, but it decreases also the size of the undershoots. In addition, the smearing of the layer might be reduced, which can be seen by comparing the maximal values for *SnapSolTimeDeriv10* in Figure 8 with the corresponding value in Figure 2.

For brevity, the results for the error in $l^\infty((0,T);L^2(\Omega))$ will be not presented, since they resemble closely the results for the errors in $L^2((0,T);L^2(\Omega))$, like in the case of the P_1 finite element. Also for the error in $L^2((0,T);SUPG)$, the results look similar as for the P_1 finite element. In Figure 9, it can be seen that again the curves for $S\Lambda_l$ overestimate the curves for $\|u_{\text{FOM}} - u_{\text{ROM}}\|_{L^2((0,T);SUPG)}$ by around of factor of 10. Thus, the way for computing an a posteriori error estimate that is described for P_1 finite elements and *SnapSol* can be used also for P_2 finite elements and *SnapSol10*.

5 Summary and outlook

An error analysis for a SUPG-stabilized POD-ROM method for convection-diffusion-reaction equations was presented. The constants in the error bounds do not blow up if the diffusion coefficient becomes small. Numerical simulations illustrate the behavior of the method. An approach for an a posteriori estimation of the error in the norm of $L^2((0,T);SUPG)$ is proposed.

The way of analyzing the errors required to consider a set that contains not only the snapshots of the FOM solution but also approximations of the temporal derivative. In our opinion, the most important open question is the derivation of error estimates with constants that are independent of the diffusion coefficient and of the inverse of the length of the time step for a SUPG-ROM method that uses only the snapshots, compare Remark 3.2. The numerical

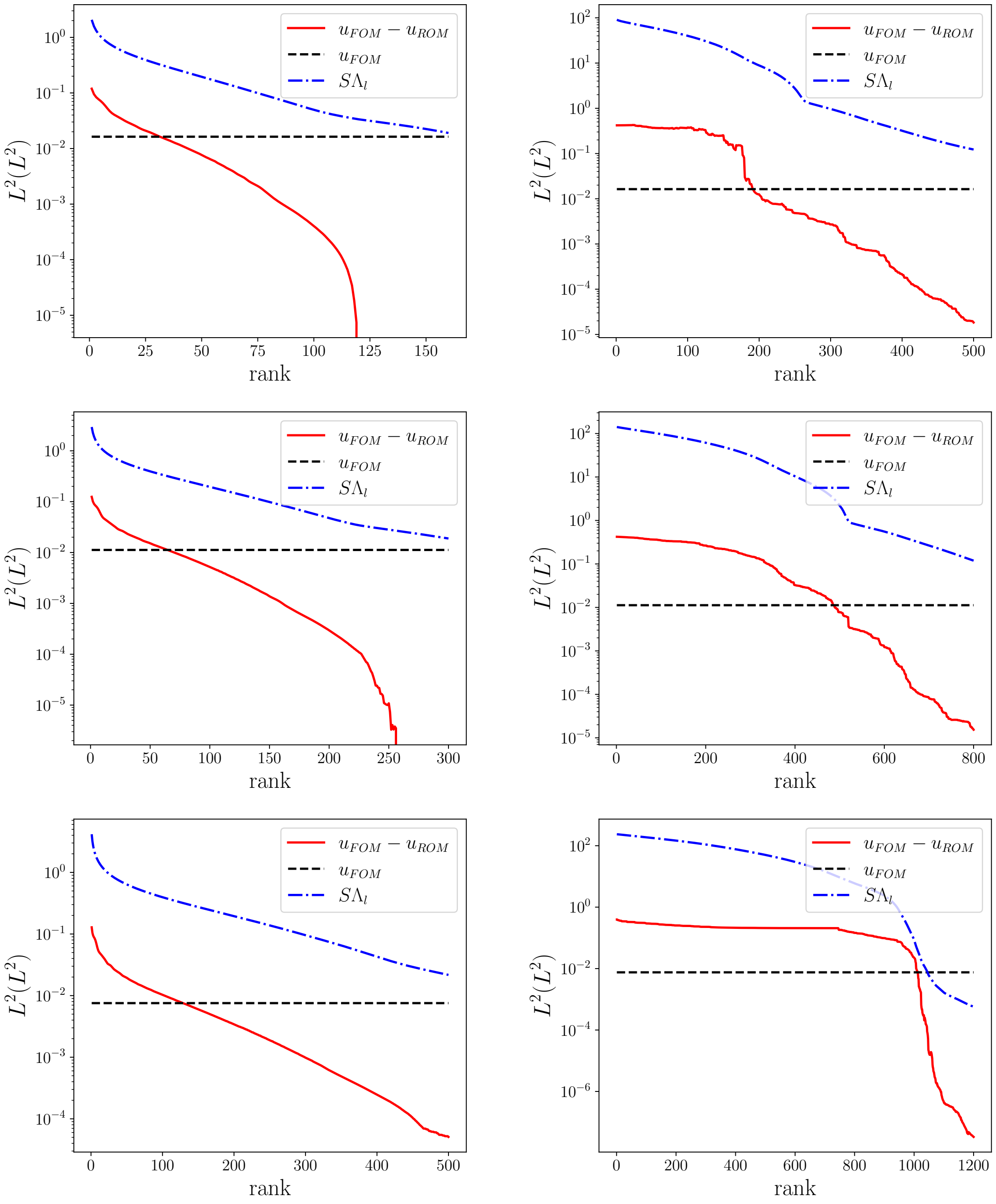

Figure 7: P_2 finite elements. Errors of the ROM solutions in $L^2((0,T);L^2(\Omega))$; top to bottom: meshes with $h \in \{2.21 \cdot 10^{-2}, 1.10 \cdot 10^{-2}, 5.5 \cdot 10^{-3}\}$; left: *SnapSol10*; right: *SnapSolTimeDeriv10*. The notation $u_{\mathrm{FOM}} - u_{\mathrm{ROM}}$ stands for $\|u_{\mathrm{FOM}} - u_{\mathrm{ROM}}\|_{L^2((0,T);L^2(\Omega))}$.

studies presented in this paper are promising that such estimates can be obtained, since they show that this method (*SnapSol*) usually performed better than the analyzed method (*SnapSolTimeDeriv*). Another open question is the analysis of SUPG-ROM methods where

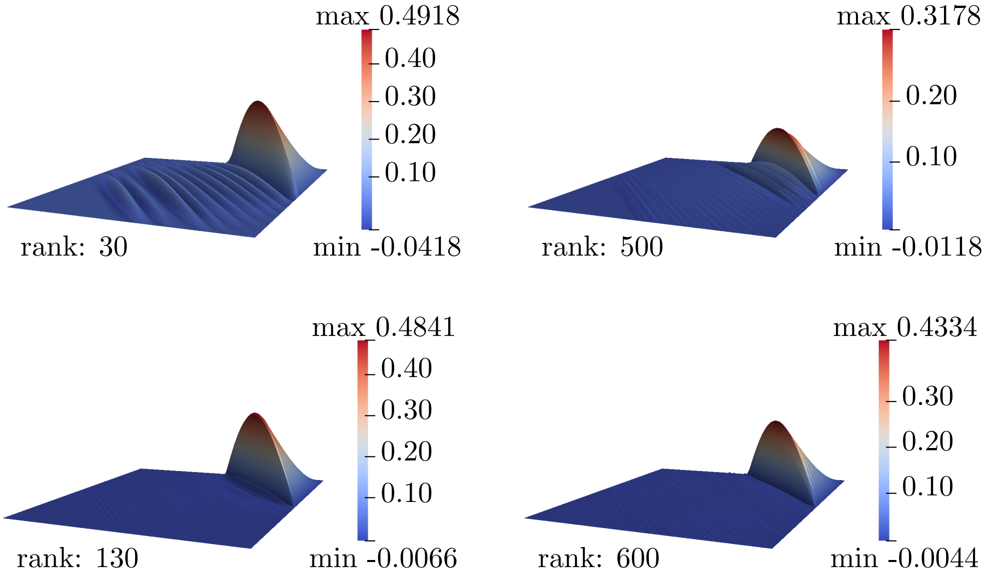

Figure 8: P_2 finite elements. Solution at $t = 1.0$ for SUPG ROM with $h = 1.10 \cdot 10^{-2}$; left: *SnapSol10*; right: *SnapSolTimeDeriv10*.

the snapshots are stored only after m time steps, with $m \in \mathbb{N}$, $m > 1$, i.e., of methods like *SnapSol10* and *SnapSolTimeDeriv10*.

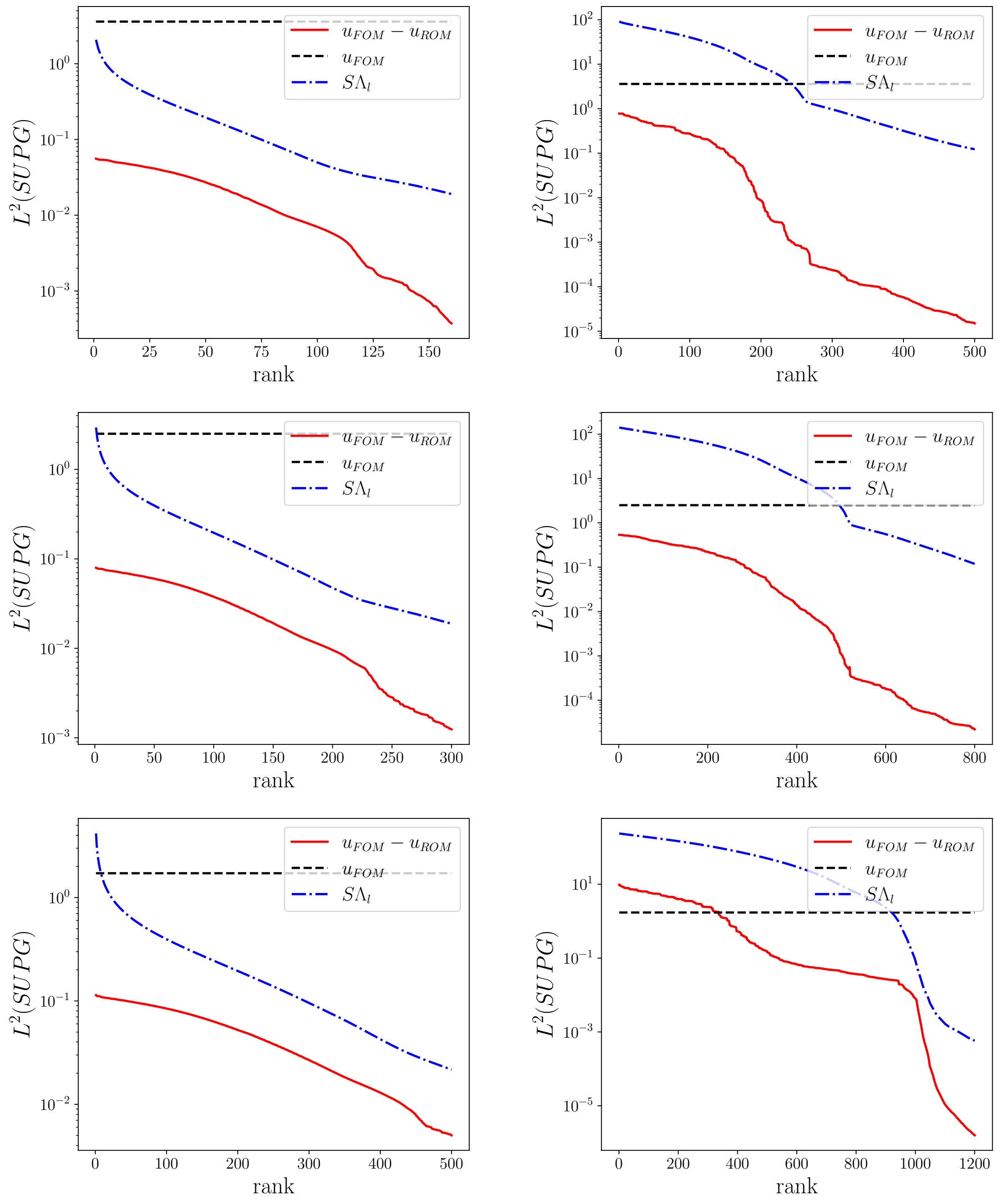

Figure 9: P_2 finite elements. Errors of the ROM solutions in $L^2((0,T);SUPG)$; mesh with $h = 1.10 \cdot 10^{-2}$; left: *SnapSol10*; right: *SnapSolTimeDeriv10*. The notation $u_{\text{FOM}} - u_{\text{ROM}}$ stands for $\|u_{\text{FOM}} - u_{\text{ROM}}\|_{L^2((0,T);SUPG)}$.

References

[1] M. Azaïez, T. Chacón Rebollo, and S. Rubino. A cure for instabilities due to advection-dominance in POD solution to advection-diffusion-reaction equations. *J. Comput. Phys.*, 425:109916, 27, 2021.

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

[3] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32(1-3):199–259, 1982. FENOMECH '81, Part I (Stuttgart, 1981).

[4] T. Chacón Rebollo, E. Delgado Ávila, M. Gómez Mármol, F. Ballarin, and G. Rozza. On a certified Smagorinsky reduced basis turbulence model. *SIAM J. Numer. Anal.*, 55(6):3047–3067, 2017.

[5] P. G. Ciarlet. *The finite element method for elliptic problems*. North-Holland Publishing Co., Amsterdam-New York-Oxford, 1978. Studies in Mathematics and its Applications, Vol. 4.

[6] T. A. Davis. Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Software*, 30(2):196–199, 2004.

[7] J. de Frutos, B. Garcí a Archilla, V. John, and J. Novo. An adaptive SUPG method for evolutionary convection-diffusion equations. *Comput. Methods Appl. Mech. Engrg.*, 273:219–237, 2014.

[8] S. Ganesan, V. John, G. Matthies, R. Meesala, S. Abdus, and U. Wilbrandt. An object oriented parallel finite element scheme for computing pdes: Design and implementation. In *IEEE 23rd International Conference on High Performance Computing Workshops (HiPCW) Hyderabad*, pages 106–115. IEEE, 2016.

[9] B. García-Archilla, V. John, and J. Novo. On the convergence order of the finite element error in the kinetic energy for high reynolds number incompressible flows. *Comput. Methods Appl. Mech. Engrg.*, 385:Article 114032, 2021.

[10] S. Giere, T. Iliescu, V. John, and D. Wells. SUPG reduced order models for convection-dominated convection-diffusion-reaction equations. *Comput. Methods Appl. Mech. Engrg.*, 289:454–474, 2015.

[11] S. Giere and V. John. Towards physically admissible reduced-order solutions for convection-diffusion problems. *Appl. Math. Lett.*, 73:78–83, 2017.

[12] J.-L. Guermond. Stabilization of Galerkin approximations of transport equations by subgrid modeling. *M2AN Math. Model. Numer. Anal.*, 33(6):1293–1316, 1999.

[13] T. J. R. Hughes and A. Brooks. A multidimensional upwind scheme with no crosswind diffusion. In *Finite element methods for convection dominated flows (Papers, Winter Ann. Meeting Amer. Soc. Mech. Engrs., New York, 1979)*, volume 34 of *AMD*, pages 19–35. Amer. Soc. Mech. Engrs. (ASME), New York, 1979.

[14] T. Iliescu and Z. Wang. Variational multiscale proper orthogonal decomposition: Navier-Stokes equations. *Numer. Methods Partial Differential Equations*, 30(2):641–663, 2014.

[15] V. John and J. Novo. Error analysis of the SUPG finite element discretization of evolutionary convection-diffusion-reaction equations. *SIAM J. Numer. Anal.*, 49(3):1149–1176, 2011.

[16] K. Kean and M. Schneier. Error analysis of supremizer pressure recovery for POD based reduced-order models of the time-dependent Navier-Stokes equations. *SIAM J. Numer. Anal.*, 58(4):2235–2264, 2020.

[17] B. Koc, S. Rubino, M. Schneier, J. R. Singler, and T. Iliescu. On optimal pointwise in time error bounds and difference quotiens for the proper orthogonal decomposition. *arXiv:submit/3405903 [math.NA] 8 Oct 2020*, 2021.

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

[19] J. Novo and S. Rubino. Error Analysis of Proper Orthogonal Decomposition Stabilized Methods for Incompressible Flows. *SIAM J. Numer. Anal.*, 59(1):334–369, 2021.

[20] H.-G. Roos, M. Stynes, and L. Tobiska. *Robust numerical methods for singularly perturbed differential equations*, volume 24 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2008. Convection-diffusion-reaction and flow problems.

[21] G. Rozza and K. Veroy. On the stability of the reduced basis method for stokes equations in parametrized domains. *Comput. Methods Appl. Mech. Engrg.*, 196(7):1244–1260, 2007.

[22] S. Rubino. A streamlike derivative POD-ROM for advection-diffusion-reaction equations. In *SMAI 2017—$8^{\rm e}$ Biennale Française des Mathématiques Appliquées et Industrielles*, volume 64 of *ESAIM Proc. Surveys*, pages 121–136. EDP Sci., Les Ulis, 2018.

[23] S. Rubino. Numerical analysis of a projection-based stabilized POD-ROM for incompressible flows. *SIAM J. Numer. Anal.*, 58(4):2019–2058, 2020.

[24] L. Sirovich. Turbulence and the dynamics of coherent structures. Parts I–III. *Quart. Appl. Math.*, 45(3):561–590, 1987.

[25] G. Stabile, F. Ballarin, G. Zuccarino, and G. Rozza. A reduced order variational multiscale approach for turbulent flows. *Adv. Comput. Math.*, 45(5-6):2349–2368, 2019.

[26] U. Wilbrandt, C. Bartsch, N. Ahmed, N. Alia, F. Anker, L. Blank, A. Caiazzo, S. Ganesan, S. Giere, G. Matthies, R. Meesala, A. Shamim, J. Venkatesan, and V. John. ParMooN—A modernized program package based on mapped finite elements. *Comput. Math. Appl.*, 74(1):74–88, 2017.