



Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

A posteriori optimization of parameters in stabilized methods for convection–diffusion problems – Part I

Volker John^{a,b,*}, Petr Knobloch^{c,1}, Simona B. Savescu^{a,2}

^aWeierstrass Institute for Applied Analysis and Stochastics (WIAS), Mohrenstr. 39, 10117 Berlin, Germany

^bFree University of Berlin, Department of Mathematics and Computer Science, Arnimallee 6, 14195 Berlin, Germany

^cCharles University, Faculty of Mathematics and Physics, Department of Numerical Mathematics, Sokolovská 83, 18675 Praha 8, Czech Republic

ARTICLE INFO

Article history:

Received 15 July 2010

Received in revised form 1 April 2011

Accepted 16 April 2011

Available online 27 April 2011

Keywords:

Stabilized finite element methods
Parameter optimization by minimizing
a target functional
SUPG method

ABSTRACT

Stabilized finite element methods for convection-dominated problems require the choice of appropriate stabilization parameters. From numerical analysis, often only their asymptotic values are known. This paper presents a general framework for optimizing stabilization parameters with respect to the minimization of a target functional. Exemplarily, this framework is applied to the SUPG finite element method and the minimization of a residual-based error estimator, an error indicator, and a functional including the crosswind derivative of the computed solution. Benefits of the basic approach are demonstrated by means of numerical results.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The numerical solution of challenging problems in various engineering applications is in general not possible with standard methods that are based, e.g., on central finite differences or the Galerkin finite element method. More sophisticated schemes become necessary that are designed to tackle the special difficulties of the underlying problem.

An example, that will be considered in this paper, are scalar convection-dominated convection–diffusion equations. Solutions of these equations exhibit very fine structures, so-called layers, which cannot be resolved on meshes that are not extremely fine, at least locally. Standard discretizations lead to solutions that are globally polluted by large spurious oscillations. In practice, stabilized methods are used. These methods introduce artificial diffusion. The difficulty consists now in defining the correct amount

of diffusion at the correct positions in the correct directions (anisotropic diffusion) such that numerical solutions with sharp layers and without spurious oscillations are obtained. A method that is optimal with respect to all criteria does not exist yet. Many proposed stabilized methods include so-called stabilization parameters. Often, the asymptotic choice of these parameters is known, e.g., that they should be proportional to the local mesh width. However, in practice, the proportionality factor has to be chosen. There is the experience that different choices of such factors might lead to considerably different numerical solutions. Moreover, the asymptotic choice of the stabilization parameters is based on global stability and convergence analysis. Local features of solutions, like layers, are not taken into account in this analysis.

We would like to mention a second example that demonstrates the difficulties of choosing parameters in numerical simulations – Large Eddy Simulation (LES) of turbulent flows. Turbulent flow simulations require the use of some turbulence model. An often used, so-called eddy viscosity model, is the Smagorinsky model [40]. This model is based on some insight into the physics of turbulent flows and it finally introduces a nonlinear viscosity into the discrete equations. It is rather easy to implement and very well understood from the point of view of mathematical analysis [32]. The derivation of the Smagorinsky model is based on some proportionality relations such that at the end a proportionality factor occurs. Experience shows that the use of a constant for this factor does not lead to good results. Instead, this factor has to be adapted to the local features of the turbulent flow field. An approach in this direction is the dynamic Smagorinsky model [12,33]. Despite all

* Corresponding author at: Weierstrass Institute for Applied Analysis and Stochastics (WIAS), Mohrenstr. 39, 10117 Berlin, Germany.

E-mail addresses: volker.john@wias-berlin.de (V. John), knobloch@karlin.mff.cuni.cz (P. Knobloch), simona.b.savescu@wias-berlin.de (S.B. Savescu).

¹ The work of P. Knobloch was supported in part by the Grant Agency of the Academy of Sciences of the Czech Republic under the Grant No. IAA100190804, by the Grant Agency of the Czech Republic under the Grant No. P201/11/1304, and by the Ministry of Education, Youth and Sports of the Czech Republic in the framework of the research project MSM 0021620839.

² The work of S.B. Savescu was supported by the Deutsche Forschungsgemeinschaft (DFG), Grant No. Jo 329/9-1.

drawbacks, e.g., see [24], the dynamic Smagorinsky model is one of the most often used and most successful LES models. Nowadays, there is another approach to control the influence of the Smagorinsky model – Variational Multiscale (VMS) methods. These methods try to select appropriate scales to which this model is applied [20,15,25,26]. Turbulent flow simulations are a typical example where principal forms of models are known but the results obtained with these models depend on the correct setting of parameters. There are many more numerical methods that require the choice of parameters and for which an a posteriori choice would greatly improve the ability to use them in applications. The a posteriori choice of parameters seems to be a widely open and challenging task in scientific computing.

The idea of choosing parameters in numerical methods a posteriori is not new, the dynamic Smagorinsky model was already mentioned. In essence, this method computes two (or more) discrete solutions in different ways and the parameter choice is based on comparing them. This idea was recently carried over to scalar convection–diffusion equations in [1], based on the work from [35]. In this approach, the different solutions are computed on coarser mesh(es). On the coarser meshes, information on the respective stabilization parameters are derived which are used to update the stabilization parameters on the fine mesh. A severe drawback of this approach is that the dimension of the parameter space is not allowed to exceed the dimension of the respective test function space. Therefore, the approach cannot be applied to the optimization of stabilization parameters in discretizations with first order finite elements as considered in this paper. Moreover, the methodology seems to be only simple for a few globally constant parameters, which is explicitly not the goal of our approach. Another method which determines the stabilization parameter on the basis of two solutions was presented in [36]. In this method, the residuals and their derivatives are used to compute a characteristic length scale which enters the formula for the stabilization parameter. The computations of the stabilization parameters in [36] are restricted to convection–diffusion equations in one dimension and a generalization to more dimensions is not obvious. A method for hyperbolic conservation laws in one dimension can be found in [10]. In this paper, the streamline-diffusion stabilization parameter and an adaptively refined grid are computed a posteriori. The adaptive algorithm uses the Dual Weighted Residual (DWR) approach [2,3] with a backward-in-time dual problem. An iterative procedure based on equilibrating components of the error estimator is used to compute the stabilization parameters and the grids. This method was extended to one-dimensional nonlinear convection–diffusion–reaction equations in [18].

The present paper considers the Streamline-Upwind Petrov–Galerkin (SUPG) finite element method for scalar convection-dominated convection–diffusion equations introduced in [21,4]. Although a number of other stabilized finite element methods have been developed in the past decades, the SUPG method is still the standard approach. In essence, this method adds numerical diffusion in streamline direction. The amount of diffusion depends on local stabilization parameters. There are different formulae for these parameters whose asymptotics are the same, see [27] for a discussion of parameter choices. The properties of solutions obtained with the SUPG method are well known: sharp layers at the correct positions are computed, but non-negligible spurious oscillations occur in a vicinity of layers. These oscillations make the use of the SUPG method in applications difficult as they correspond in general to unphysical situations, like negative concentrations. There have been a large number of attempts to improve the SUPG method in order to get rid of these oscillations while preserving its good properties. However, none of these so-called Spurious Oscillations at Layers Diminishing (SOLD) methods turned out to be entirely successful [27,28].

To improve the solutions obtained with the SUPG method, the present paper pursues a different approach than the SOLD methods. It relies on the optimization of the stabilization parameter, however, in contrast to [1,10,18,36,35], the parameter optimization is formulated as minimization of some functional. This is a nonlinear constrained optimization problem that has to be solved iteratively. A key component of this approach consists in the efficient computation of the Fréchet derivative of the functional with respect to the stabilization parameter. This is achieved by utilizing an adjoint problem with an appropriate right-hand side. The aim of the present paper is to provide a new general framework for the optimization of parameters in stabilized methods for convection–diffusion equations and to demonstrate exemplarily the benefits of this approach. A comprehensive discussion of the choice of appropriate target functionals is postponed to the second part of this paper.

The paper is organized as follows. Section 2 presents the equation and the SUPG method. A general approach for computing the Fréchet derivative of a functional that depends on the numerical solution with respect to parameters in the numerical method is presented in Section 3. This approach is applied to the SUPG method in Section 4. Section 5 contains a proof of concept. It is demonstrated that errors to known solutions can be reduced by using as functional the error in some norm. For problems with unknown solutions, Section 6 illustrates the application of the a posteriori parameter choice based on the minimization of a residual-based a posteriori error estimator, an error indicator, and a functional that includes the crosswind derivative of the computed solution. The most important conclusions, open problems, and an outlook are presented in Section 7. Throughout the paper, standard notations are used for usual function spaces and norms, see, e.g., [6]. The notation $(\cdot, \cdot)_G$ with a set $G \subset \mathbb{R}^d$, $d = 1, 2, 3$, is used for the inner product in the space $L^2(G)$ or $L^2(G)^d$, with $(\cdot, \cdot) = (\cdot, \cdot)_\Omega$.

2. The convection–diffusion problem and its SUPG stabilization

Consider the scalar convection–diffusion problem

$$-\varepsilon \Delta u + \mathbf{b} \cdot \nabla u + cu = f \text{ in } \Omega, \quad u = u_b \text{ on } \Gamma^D, \quad \varepsilon \frac{\partial u}{\partial \mathbf{n}} = g \text{ on } \Gamma^N. \quad (1)$$

Here, $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, is a bounded domain with a polyhedral Lipschitz-continuous boundary $\partial\Omega$ and Γ^D , Γ^N are disjoint and relatively open subsets of $\partial\Omega$ satisfying $\text{meas}_{d-1}(\Gamma^D) > 0$ and $\Gamma^D \cup \Gamma^N = \partial\Omega$. Furthermore, \mathbf{n} is the outward unit normal vector to $\partial\Omega$, $\varepsilon > 0$ is a constant diffusivity, $\mathbf{b} \in W^{1,\infty}(\Omega)^d$ is the flow velocity, $c \in L^\infty(\Omega)$ is the reaction coefficient, $f \in L^2(\Omega)$ is a given outer source of the unknown scalar quantity u , and $u_b \in H^{1/2}(\Gamma^D)$, $g \in L^2(\Gamma^N)$ are given functions specifying the boundary conditions. The usual assumption that

$$c - \frac{1}{2} \text{div} \mathbf{b} \geq c_0 \geq 0 \quad (2)$$

with a constant c_0 is made. Moreover, it is assumed that

$$\{\mathbf{x} \in \partial\Omega; (\mathbf{b} \cdot \mathbf{n})(\mathbf{x}) < 0\} \subset \Gamma^D, \quad (3)$$

i.e., the inflow boundary is a part of the Dirichlet boundary Γ^D .

This paper studies finite element methods for the numerical solution of (1). To this end, (1) is transformed into a variational formulation. Let $\tilde{u}_b \in H^1(\Omega)$ be an extension of u_b (i.e., the trace of \tilde{u}_b equals u_b on Γ^D) and let

$$V = \{v \in H^1(\Omega); \quad v = 0 \text{ on } \Gamma^D\}.$$

Then, a weak formulation of (1) reads: Find $u \in H^1(\Omega)$ such that $u - \tilde{u}_b \in V$ and

$$a(u, v) = (f, v) + (g, v)_{\Gamma^N} \quad \forall v \in V, \quad (4)$$

where

$$a(u, v) = \varepsilon(\nabla u, \nabla v) + (\mathbf{b} \cdot \nabla u, v) + (cu, v).$$

In view of (2) and (3), the weak formulation (4) has a unique solution.

Let $\{\mathcal{T}_h\}_h$ be a family of triangulations of Ω parameterized by positive parameters h whose only accumulation point is zero. The triangulations \mathcal{T}_h are assumed to consist of a finite number of open (mapped) polyhedral subsets K of Ω such that $\bar{\Omega} = \bigcup_{K \in \mathcal{T}_h} \bar{K}$ and the closures of any two different sets in \mathcal{T}_h are either disjoint or possess either a common vertex or a common edge or (if $d = 3$) a common face. Further, it is assumed that any edge (face) of \mathcal{T}_h which lies on $\partial\Omega$ is contained either in Γ^D or in Γ^N .

For each h , a finite element space $W_h \subset H^1(\Omega)$ defined on \mathcal{T}_h and approximating the space $H^1(\Omega)$ in the usual sense is introduced, see, e.g., [6]. Furthermore, for each h , let $\tilde{u}_{bh} \in W_h$ be a function whose trace on Γ^D approximates u_b . Finally, we set $V_h = W_h \cap V$. Then, the Galerkin discretization of (1) reads: Find $u_h \in W_h$ such that $u_h - \tilde{u}_{bh} \in V_h$ and

$$a(u_h, v_h) = (f, v_h) + (g, v_h)_{\Gamma^N} \quad \forall v_h \in V_h. \tag{5}$$

Again, this problem is uniquely solvable. As discussed in the introduction, the Galerkin discretization (5) is inappropriate if convection dominates diffusion since in this case the discrete solution is usually globally polluted by spurious oscillations. An improvement can be achieved by adding a stabilization term to the Galerkin discretization. One of the most efficient procedures of this type is the SUPG method [21,4] that is frequently used because of its stability properties, its higher-order accuracy in appropriate norms, and its easy implementation, see, e.g., [37].

The SUPG stabilization depends on a stabilization parameter that will be denoted by y_h in the following. It is assumed that all admissible stabilization parameters are contained in a finite-dimensional space $Y_h \subset L^\infty(\Omega)$. For example, Y_h can consist of piecewise constant functions with respect to the triangulation \mathcal{T}_h .

The SUPG discretization of (1) reads: Find $u_h \in W_h$ such that $u_h - \tilde{u}_{bh} \in V_h$ and

$$a(u_h, v_h) + s_h(y_h; u_h, v_h) = (f, v_h) + (g, v_h)_{\Gamma^N} + r_h(y_h; v_h) \quad \forall v_h \in V_h, \tag{6}$$

where

$$s_h(y_h; u_h, v_h) = (-\varepsilon \Delta_h u_h + \mathbf{b} \cdot \nabla u_h + cu_h, y_h \mathbf{b} \cdot \nabla v_h),$$

$$r_h(y_h; v_h) = (f, y_h \mathbf{b} \cdot \nabla v_h).$$

The SUPG method requires that the functions from W_h are H^2 on each mesh cell of \mathcal{T}_h , which is satisfied for common finite element spaces. The notation Δ_h denotes the cell-wise defined Laplace operator.

A detailed discussion of ways that are used in practice for choosing the stabilization parameter y_h in the case of first order finite elements can be found in [27]. Modifications for higher order finite elements are discussed, e.g., in [7,9,11]. A common choice is, for any mesh cell $K \in \mathcal{T}_h$,

$$y_h|_K = \frac{h_K}{2p|\mathbf{b}|} \xi(Pe_K) \quad \text{with} \quad \xi(\alpha) = \coth \alpha - \frac{1}{\alpha}, \quad Pe_K = \frac{|\mathbf{b}|h_K}{2p\varepsilon}, \tag{7}$$

where h_K is the cell diameter in the direction of the convection vector \mathbf{b} , p is the polynomial degree of the local finite element space, and Pe_K is the local Péclet number which determines whether the problem is locally (i.e., within a particular mesh cell) convection dominated or diffusion dominated. Note that, generally, the parameters h_K , Pe_K and $y_h|_K$ are functions of the points $\mathbf{x} \in K$. The evaluation of the cell diameter in the direction of the convection is discussed also in [27].

If (2) holds with $c_0 > 0$, a sufficient condition for the ellipticity of the bilinear form on the left-hand side of (6) in a standard SUPG norm is

$$0 \leq y_h(\mathbf{x}) \leq \frac{1}{2} \min \left\{ \frac{(\text{diam}(K))^2}{\varepsilon c_{\text{inv}}^2}, \frac{c_0}{\|c\|_{0,\infty,K}^2} \right\}, \quad \mathbf{x} \in K, \tag{8}$$

see [37], where $\text{diam}(K)$ denotes the diameter of K , c_{inv} is a constant from the inverse inequality

$$\|\Delta v_h\|_{0,K} \leq c_{\text{inv}} [\text{diam}(K)]^{-1} \|v_h\|_{1,K} \quad \forall v_h \in V_h,$$

and $\|\cdot\|_{0,\infty,K}$ denotes the $L^\infty(K)$ norm. The first term in the minimum in (8) does not appear for P_1 finite elements and for Q_1 finite elements on rectangles since in these cases $\Delta_h v_h = 0$ for all $v_h \in V_h$.

An important class of convection–diffusion problems possesses the properties $\text{div} \mathbf{b} = 0$, e.g., if \mathbf{b} is the velocity field of an incompressible fluid, and $c = 0$. Hence, (2) holds only with $c_0 = 0$. For this class of problems, one can prove the ellipticity of the SUPG bilinear form (in a weaker SUPG norm than for $c_0 > 0$) if

$$0 \leq y_h(\mathbf{x}) \leq \frac{(\text{diam}(K))^2}{\varepsilon c_{\text{inv}}^2}, \quad \mathbf{x} \in K. \tag{9}$$

For the same reason as above, the bound on the right-hand side of (9) is not needed if P_1 finite elements or Q_1 finite elements on rectangles are used.

In the special case of a constant convection field and a uniform grid, the stabilization parameter given by (7) is the same in all mesh cells, independently of local features of the solution, like layers. This does not seem to be an optimal choice. This paper will present and study an approach for choosing the values of the stabilization parameter locally, based on the minimization of a functional that measures or estimates the accuracy of the computed solution.

3. Optimization of parameters in numerical methods with respect to the minimization of a functional

Let us assume that a numerical method for the solution of (1) is given and let the method depend on a parameter $y_h \in Y_h$. An example is the SUPG method (6). Let $D_h \subset Y_h$ be an open set such that, for any $y_h \in D_h$, the considered method has a unique solution $u_h \in W_h$. To emphasize that u_h depends on y_h , we shall write $u_h(y_h)$ instead of u_h in the following. Let $I_h : W_h \rightarrow \mathbb{R}$ be a functional such that

$$\Phi_h(y_h) := I_h(u_h(y_h))$$

represents a measure of the error of the discrete solution $u_h(y_h)$ corresponding to a given parameter y_h . The aim is to compute a parameter $y_h \in D_h$ for which Φ_h attains a minimum on D_h or is near to a minimum (or the infimum) of Φ_h on D_h . This nonlinear minimization problem has to be solved iteratively. Reasonable iterative schemes require at least information on how Φ_h changes if the parameter y_h is changed, i.e., on the Fréchet derivative of Φ_h . An efficient way to compute this derivative is needed. Such a way will be explained in this section.

For any $y_h \in D_h$, it holds $u_h(y_h) = \tilde{u}_h(y_h) + \tilde{u}_{bh}$ with $\tilde{u}_h : D_h \rightarrow V_h$. Thus, one does not need to consider the space W_h in the optimization process but can work with the space V_h , which is more convenient.

Denote $\tilde{I}_h(w_h) = I_h(w_h + \tilde{u}_{bh})$ for any $w_h \in V_h$. Then $\tilde{I}_h : V_h \rightarrow \mathbb{R}$ and

$$\Phi_h(y_h) = \tilde{I}_h(\tilde{u}_h(y_h)) \quad \forall y_h \in D_h.$$

Let us assume that the mappings $\tilde{I}_h = \tilde{I}_h(w_h)$ and $\tilde{u}_h = \tilde{u}_h(y_h)$ are Fréchet–differentiable. The Fréchet derivatives are denoted by $D\tilde{I}_h : V_h \rightarrow V'_h$ and $D\tilde{u}_h : D_h \rightarrow \mathcal{L}(Y_h, V_h)$. Then, the Fréchet derivative $D\Phi_h : D_h \rightarrow Y'_h$ of Φ_h exists and it is given by

$$D\Phi_h(y_h) = D\tilde{I}_h(\tilde{u}_h(y_h))D\tilde{u}_h(y_h) \quad \forall y_h \in D_h. \quad (10)$$

The naive way of using this formula for computing $D\Phi_h(y_h)$ is very inefficient as the computation of $D\tilde{u}_h(y_h)$ requires the solution of dim Y_h systems of dim V_h algebraic equations.

The problem of efficiently evaluating a derivative of form (10) is well known, e.g., from optimal control of partial differential equations. There is a way for obtaining this derivative that is based on an appropriate adjoint problem, e.g., see [42]. This way will be applied to the situation considered in this paper. The minimization of Φ_h occurs under the condition that $u_h(y_h)$ should fulfill the discretized partial differential equation (6), i.e., for a residual operator $R_h : V_h \times Y_h \rightarrow V'_h$ holds

$$R_h(\tilde{u}_h(y_h), y_h) = 0 \quad \forall y_h \in D_h. \quad (11)$$

For the SUPG method (6), the operator R_h is given by

$$\begin{aligned} \langle R_h(w_h, y_h), v_h \rangle &= a(w_h + \tilde{u}_{bh}, v_h) + s_h(y_h; w_h + \tilde{u}_{bh}, v_h) \\ &\quad - (f, v_h) - (g, v_h)_{\Gamma^N} - r_h(y_h; v_h) \\ &\quad \forall v_h, w_h \in V_h, y_h \in Y_h. \end{aligned}$$

Differentiating (11) with respect to y_h leads to

$$\partial_w R_h(\tilde{u}_h(y_h), y_h)D\tilde{u}_h(y_h) + \partial_y R_h(\tilde{u}_h(y_h), y_h) = 0 \quad \forall y_h \in D_h, \quad (12)$$

provided that the mapping $R_h = R_h(w_h, y_h)$ is Fréchet-differentiable. Note that $\partial_w R_h : V_h \times Y_h \rightarrow \mathcal{L}(V_h, V'_h)$ and $\partial_y R_h : V_h \times Y_h \rightarrow \mathcal{L}(Y_h, V'_h)$. Assume that there is a mapping $\psi_h : D_h \rightarrow V_h$ such that

$$\langle D\Phi_h(y_h), \tilde{y}_h \rangle = -\langle (\partial_y R_h)(\tilde{u}_h(y_h), y_h)\tilde{y}_h, \psi_h(y_h) \rangle \quad \forall y_h \in D_h, \tilde{y}_h \in Y_h. \quad (13)$$

Then, according to (12), one obtains

$$\begin{aligned} \langle D\Phi_h(y_h), \tilde{y}_h \rangle &= \langle (\partial_w R_h)(\tilde{u}_h(y_h), y_h)D\tilde{u}_h(y_h)\tilde{y}_h, \psi_h(y_h) \rangle \\ &= \langle (\partial_w R_h)'(\tilde{u}_h(y_h), y_h)\psi_h(y_h), D\tilde{u}_h(y_h)\tilde{y}_h \rangle \\ &\quad \forall y_h \in D_h, \tilde{y}_h \in Y_h, \end{aligned}$$

where the adjoint operator is defined by

$$\begin{aligned} \langle (\partial_w R_h)'(w_h, y_h)v_h, \tilde{v}_h \rangle &= \langle (\partial_w R_h)(w_h, y_h)\tilde{v}_h, v_h \rangle \\ &\quad \forall v_h, \tilde{v}_h, w_h \in V_h, y_h \in Y_h. \end{aligned}$$

On the other hand, from (10) follows that

$$\begin{aligned} \langle D\Phi_h(y_h), \tilde{y}_h \rangle &= \langle D\tilde{I}_h(\tilde{u}_h(y_h))D\tilde{u}_h(y_h), \tilde{y}_h \rangle \\ &= \langle D\tilde{I}_h(\tilde{u}_h(y_h)), D\tilde{u}_h(y_h)\tilde{y}_h \rangle \quad \forall y_h \in D_h, \tilde{y}_h \in Y_h. \end{aligned}$$

The two representations of $D\Phi_h(y_h)$ suggest to define $\psi_h(y_h)$ as the solution of the adjoint problem, cf., e.g., [13,38],

$$(\partial_w R_h)'(\tilde{u}_h(y_h), y_h)\psi_h(y_h) = D\tilde{I}_h(\tilde{u}_h(y_h)) \quad \forall y_h \in D_h. \quad (14)$$

Then $\psi_h(y_h)$ satisfies (13) and hence the Fréchet derivative of Φ_h is given by

$$D\Phi_h(y_h) = -(\partial_y R_h)'(\tilde{u}_h(y_h), y_h)\psi_h(y_h) \quad \forall y_h \in D_h. \quad (15)$$

The adjoint operator is defined by

$$\begin{aligned} \langle (\partial_y R_h)'(w_h, y_h)v_h, \tilde{y}_h \rangle &= \langle (\partial_y R_h)(w_h, y_h)\tilde{y}_h, v_h \rangle \\ &\quad \forall v_h, w_h \in V_h, y_h, \tilde{y}_h \in Y_h. \end{aligned}$$

To clarify the approach, we would like to give its algebraic version. All operators and functionals are defined using finite-dimensional spaces, such that their Fréchet derivatives can be represented by matrices and vectors. Let $y_h \in D_h$ be given and denote by $D\Phi_h \in \mathbb{R}^{1 \times \dim Y_h}$ and $D\tilde{I}_h \in \mathbb{R}^{1 \times \dim V_h}$ the vectors representing the derivatives $D\Phi_h(y_h)$ and $D\tilde{I}_h(\tilde{u}_h(y_h))$, respectively. Furthermore, let $D\tilde{u}_h \in \mathbb{R}^{\dim V_h \times \dim Y_h}$, $\partial_w R_h \in \mathbb{R}^{\dim V_h \times \dim V_h}$, and $\partial_y R_h \in \mathbb{R}^{\dim V_h \times \dim Y_h}$ be

the matrices representing the derivatives $D\tilde{u}_h(y_h)$, $\partial_w R_h(\tilde{u}_h(y_h), y_h)$, and $\partial_y R_h(\tilde{u}_h(y_h), y_h)$, respectively. Then, equation (10) holds true if and only if

$$D\Phi_h \mathbf{y} = D\tilde{I}_h D\tilde{u}_h \mathbf{y} \quad \forall \mathbf{y} \in \mathbb{R}^{\dim Y_h}. \quad (16)$$

Relation (12) is equivalent to

$$\mathbf{v}^T \partial_w R_h D\tilde{u}_h \mathbf{y} = -\mathbf{v}^T \partial_y R_h \mathbf{y} \quad \forall \mathbf{v} \in \mathbb{R}^{\dim V_h}. \quad (17)$$

The goal of the adjoint approach consists in reformulating the right-hand side of (16). To this end, choose \mathbf{v} in (17) such that $\mathbf{v}^T \partial_w R_h = D\tilde{I}_h$, i.e.,

$$\boldsymbol{\psi} := \mathbf{v} = (\partial_w R_h)^{-T} D\tilde{I}_h^T,$$

which is the algebraic version of (14). Inserting $\boldsymbol{\psi}$ into (16) and using (17) gives

$$D\Phi_h \mathbf{y} = \boldsymbol{\psi}^T \partial_w R_h D\tilde{u}_h \mathbf{y} = -\boldsymbol{\psi}^T \partial_y R_h \mathbf{y} \quad \forall \mathbf{y} \in \mathbb{R}^{\dim Y_h}.$$

This is equivalent to

$$D\Phi_h = -\boldsymbol{\psi}^T \partial_y R_h,$$

that is the algebraic version of (15).

4. Application to the SUPG method

For the SUPG method (6), there are

$$\begin{aligned} \langle (\partial_w R_h)(w_h, y_h)\tilde{v}_h, v_h \rangle &= a(\tilde{v}_h, v_h) + s_h(y_h; \tilde{v}_h, v_h), \\ \langle (\partial_y R_h)(w_h, y_h)\tilde{y}_h, v_h \rangle &= s_h(\tilde{y}_h; w_h + \tilde{u}_{bh}, v_h) - r_h(\tilde{y}_h; v_h) \end{aligned}$$

for any $y_h, \tilde{y}_h \in Y_h$ and $v_h, \tilde{v}_h, w_h \in V_h$. Thus, for any $y_h \in D_h$, the auxiliary function $\psi_h(y_h) \in V_h$ is the solution of

$$a(v_h, \psi_h(y_h)) + s_h(y_h; v_h, \psi_h(y_h)) = \langle D\tilde{I}_h(\tilde{u}_h(y_h)), v_h \rangle \quad \forall v_h \in V_h \quad (18)$$

and the Fréchet derivative of Φ_h is given by

$$\langle D\Phi_h(y_h), \tilde{y}_h \rangle = -s_h(\tilde{y}_h; u_h(y_h), \psi_h(y_h)) + r_h(\tilde{y}_h; \psi_h(y_h)) \quad \forall \tilde{y}_h \in Y_h.$$

We define Y_h as the space of piecewise constant functions. After having solved (18) for a given stabilization parameter y_h , the Fréchet derivative of Φ_h at y_h with respect to the stabilization parameter is available.

The most popular [34] quasi-Newton method for solving a non-linear minimization problem is the BFGS (Broyden, Fletcher, Goldfarb, Shanno) method [5,8,14,39]. This method requires only the gradient of the functional with respect to the stabilization parameter. By measuring the changes of the gradients, it constructs a model for the functional that delivers information to obtain super-linear convergence. The cost consists in the storage of the gradients, which are piecewise constant finite element functions. For practical reasons, this can be done only for a limited number of gradients. The resulting algorithm is called limited memory BFGS or L-BFGS, see Algorithm 7.5 in [34]. This algorithm is used in the simulations presented below. We could observe a dramatic improvement of efficiency compared with the application of the steepest descent method which was used in preliminary numerical studies.

The L-BFGS method proposes a search direction for updating the stabilization parameter in the k th iteration, $k \geq 0$. In addition, a step length $\alpha^{(k)}$ is needed. In our implementation of the method, the step length is determined such that the decrease of the functional I_h is locally maximized. To this end, the initial guess for each step length $\alpha^{(k)}$ is a value α_{ini} . If the application of α_{ini} leads to a reduction of the target functional, the step length will be doubled. This step is repeated as long as the target functional decreases. If the application of α_{ini} does not lead to a reduction of the value of the target functional, α_{ini} will be divided by 2. The reduction of α_{ini}

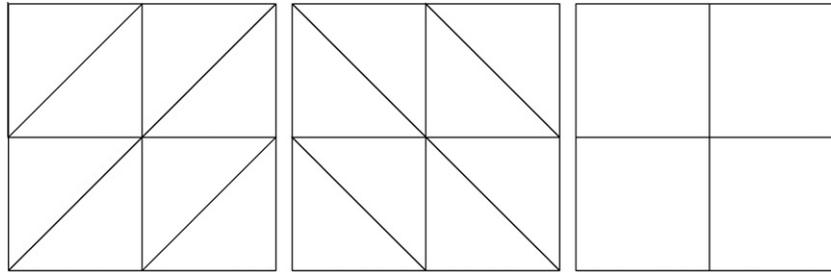


Fig. 1. Types of triangulations used in the computations (left to right): Grid 1, Grid 2, and Grid 3 (level 1).

will be stopped if either a step length is found that leads to a decrease of I_h or if a minimal value α_{\min} for the step length is obtained. The iteration stops either after reaching a prescribed maximal number of iterations k_{\max} or if the decrease of the target functional becomes too slow. The concrete test for the last stopping criterion is

$$\frac{\Phi_h(y_h^{(k-10)}) - \Phi_h(y_h^{(k)})}{\Phi_h(y_h^{(k-10)})} \leq d_{\min}, \quad k \geq 10.$$

If the computation of a search direction with L-BFGS is not successful or if the proposed step length becomes smaller than α_{\min} , L-BFGS is restarted.

Of course, before the solution with a proposed stabilization parameter $y_h^{(k+1)}$ is computed, the stabilization parameter is always restricted to admissible values according to (8) and (9). The values from [17] are used for c_{inv} in (8) and (9).

In our numerical tests, the initial step length parameter was set similarly to a proposal from [34]

$$\alpha_{ini}^{(0)} = 10^{-6},$$

$$\alpha_{ini}^{(k)} = \max \left\{ \min \left\{ 1, \frac{D\Phi_h(y_h^{(k-1)})^T \Delta y_h^{(k-1)}}{D\Phi_h(y_h^{(k)})^T \Delta y_h^{(k)}} \right\}, 10^{-6} \right\}, \quad k \geq 1,$$

where $\Delta y_h^{(k)}$ is the search direction proposed by the L-BFGS method in the k th step. The minimal step length parameter was set to be $\alpha_{\min} = 10^{-12}$, the maximal number of iterations was prescribed with $k_{\max} = 10000$ (which was never reached), at most 100 gradients in the L-BFGS method were stored, and the parameter in the stopping criterion was set to be $d_{\min} = 10^{-4}$. The stabilization parameter was initialized with the standard choice (7). Most of the computations were performed and double-checked with two codes, one of them MoonMMD [29].

5. Proof of concept: parameter optimization with respect to errors

A common approach for supporting error estimates consists in prescribing a solution of (1), that defines also the right-hand side and the boundary conditions of (1), and measuring errors of the numerical solution in certain norms. If errors can be measured, it should be possible with the proposed methodology to compute a SUPG stabilization parameter such that these errors are reduced compared with the standard choice of the SUPG parameter (7). This section studies this topic.

Numerical studies with respect to the error in the $L^2(\Omega)$ norm and the $H^1(\Omega)$ semi norm were performed. For shortness, the detailed presentation will be restricted to the error in the $L^2(\Omega)$ norm

$$I_h(w_h) = \|u - w_h\|_{0,\Omega}^2.$$

Then, the right-hand side of the adjoint problem (18) becomes

$$\langle D\tilde{I}_h(\tilde{u}_h(y_h)), v_h \rangle = -2(u - u_h(y_h), v_h). \tag{19}$$

At the end of this section, some remarks will be given on the error in the $H^1(\Omega)$ semi norm.

A difficulty consists in finding or defining examples that, on the one hand, have a known solution and, on the other hand, possess typical features of solutions of convection-dominated problems, in particular layers. Below, results obtained with two examples defined in [30] will be presented. The solutions of these examples depend on the diffusion coefficient ε , and so the right-hand sides do. As already noticed in [31], high order quadrature rules are necessary to keep the quadrature error for the right-hand side small in the case of small ε . For this reason, the diffusion coefficient was chosen only three or four orders of magnitude smaller than the convection in these examples.

Both examples are defined on the unit square. In the computations, triangular grids (Grid 1 in Fig. 1) with P_1, P_2, P_3 finite elements and square grids (Grid 3) with Q_1, Q_2, Q_3 finite elements were used. Level 0 of Grid 1 consists of two triangles and level 0 of Grid 3 of one square. The grids were regularly refined using so-called red refinement. A quadrature rule that is exact for polynomials of degree 19 was used on triangles and a Gaussian quadrature rule that is exact for polynomials of degree 17 on squares.

Example 5.1 (Example with interior layer). This example is given by $\Omega = (0, 1)^2$, $\Gamma^D = \partial\Omega$, $\varepsilon = 10^{-4}$, $\mathbf{b} = (2, 3)^T$, $c = 2$. The right-hand side f and the Dirichlet boundary condition u_b are prescribed such that

$$u(x, y) = 16x(1 - x)y(1 - y) \times \left(\frac{1}{2} + \frac{\arctan[2\varepsilon^{-1/2}(0.25^2 - (x - 0.5)^2 - (y - 0.5)^2)]}{\pi} \right)$$

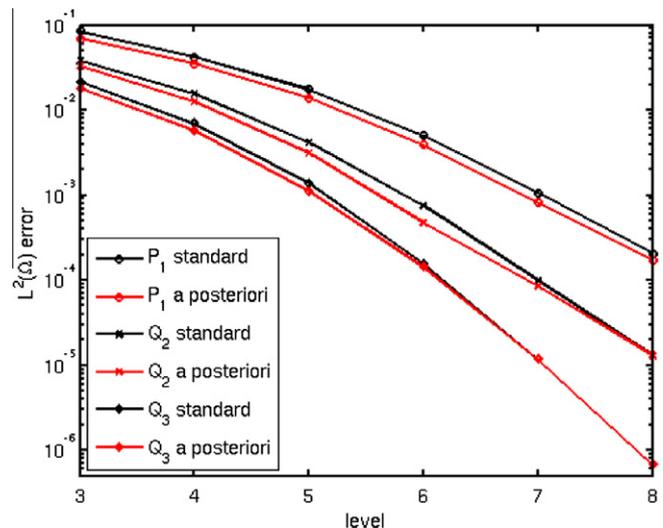


Fig. 2. Example 5.1, $L^2(\Omega)$ errors for different finite elements, comparison of standard parameter choice (7) and the a posteriori choice based on minimizing the $L^2(\Omega)$ error.

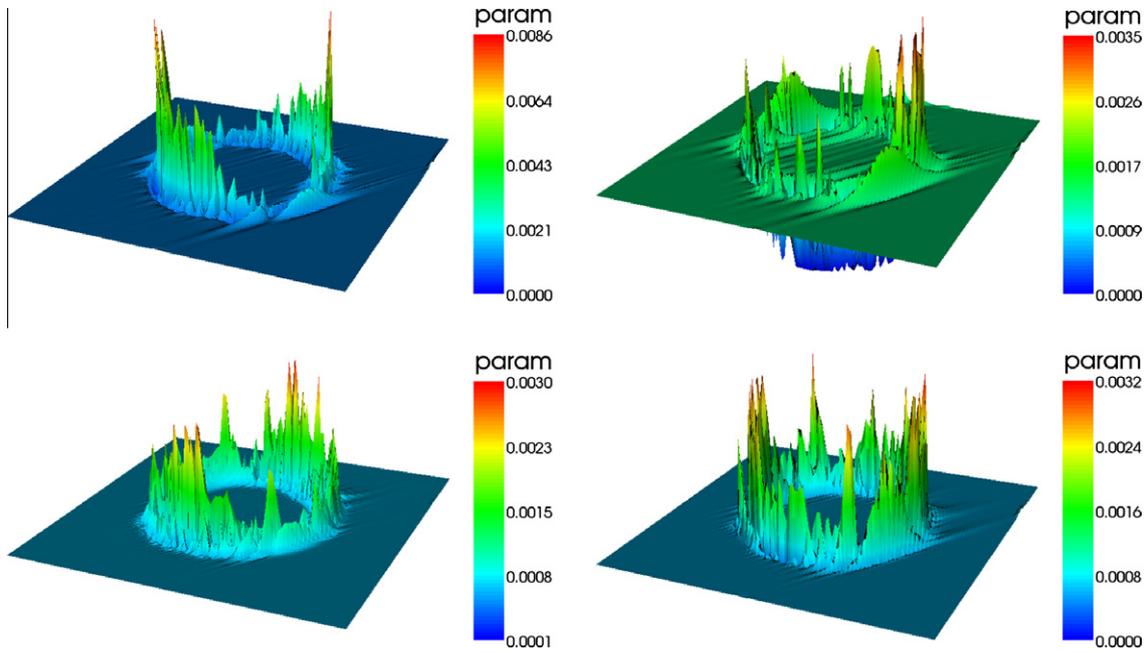


Fig. 3. Example 5.1, a posteriori defined stabilization parameters; top left: P_1 , standard parameter $y_h = 1.294391e - 3$; top right: Q_1 , standard parameter $y_h = 1.294391e - 3$; bottom left: P_2 , standard parameter $y_h = 6.433494e - 4$; bottom right: Q_2 , standard parameter $y_h = 6.433494e - 4$; all level 7 (visualization by projection to P_1 or Q_1 finite element).

is the solution of (1). The solution has the form of a circular hump in the center of the domain.

A comparison of the $L^2(\Omega)$ errors obtained with the standard parameter choice (7) and the a posteriori choice based on the adjoint problem with right-hand side (19) is presented in Fig. 2. It can be observed that the a posteriori parameter choice leads in fact to solutions with smaller $L^2(\Omega)$ error. Naturally, on finer grids, where the stabilization loses importance, the error reduction becomes smaller. Since the convection is constant, the standard parameter (7) is constant on a uniform grid, too. Fig. 3 shows the distribution of the stabilization parameter for different finite elements on certain grid levels. The corresponding standard parameters are given in the caption. It can be seen that the a posteriori methodology changes the parameter in the layer, which is not surprising since the stabilization is needed in the layer. On many mesh cells at the layer, the parameter is increased considerably. A large stabilization parameter can be observed at the front and at the back (with respect to the direction of the convection) of the hump. Note, in few mesh cells at the layer, a reduction of the stabilization parameter is proposed. This reduction is in general much smaller than the increase of the parameter in other mesh cells and therefore it is only visible in the picture for the Q_1 finite element. In summary, the main mechanism to reduce the $L^2(\Omega)$ error was always a significant increase of the stabilization parameter within the layer.

Example 5.2 (Example with boundary layer). This example is defined by $\Omega = (0, 1)^2$, $\Gamma^D = \partial\Omega$, $\varepsilon = 10^{-3}$, $\mathbf{b} = (2, 3)^T$, and $c = 1$. The prescribed solution

$$u(x, y) = xy^2 - y^2 \exp\left(\frac{2(x-1)}{\varepsilon}\right) - x \exp\left(\frac{3(y-1)}{\varepsilon}\right) + \exp\left(\frac{2(x-1) + 3(y-1)}{\varepsilon}\right)$$

defines the right-hand side f and the Dirichlet boundary condition u_b . It possesses boundary layers at $x = 1$ and $y = 1$, see Fig. 7.

Fig. 4 presents comparisons of the $L^2(\Omega)$ errors obtained with the standard and the a posteriori parameter choices. Clearly, the a posteriori parameter choice leads always to a reduction of the $L^2(\Omega)$ errors. However, a higher order of convergence cannot be observed. A posteriori computed parameters are presented in Fig. 5. It can be noticed that the optimization of the $L^2(\Omega)$ error reduces the stabilization parameters in the layers.

Concerning the a posteriori parameter choice based on the error in the $H^1(\Omega)$ semi norm, we could observe essentially the same behavior as for the $L^2(\Omega)$ norm: the $H^1(\Omega)$ semi norm error becomes always smaller than for the solution with the standard parameter (7). However, sometimes the error reduction is very

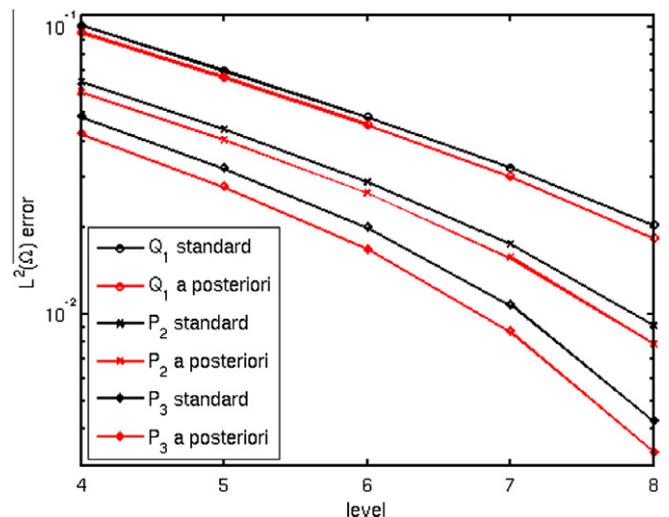


Fig. 4. Example 5.2, $L^2(\Omega)$ errors for different finite elements, comparison of standard parameter choice (7) and the a posteriori choice based on minimizing the $L^2(\Omega)$ error.

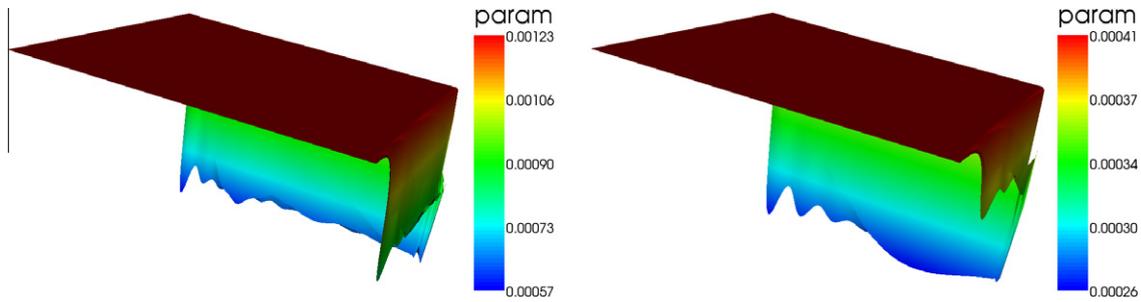


Fig. 5. Example 5.2, a posteriori defined stabilization parameters; left: Q_1 , standard parameter $y_h = 1.225160e - 3$; right: Q_2 , standard parameter $y_h = 5.741186e - 4$; both level 7 (visualization by projection to Q_1 finite element).

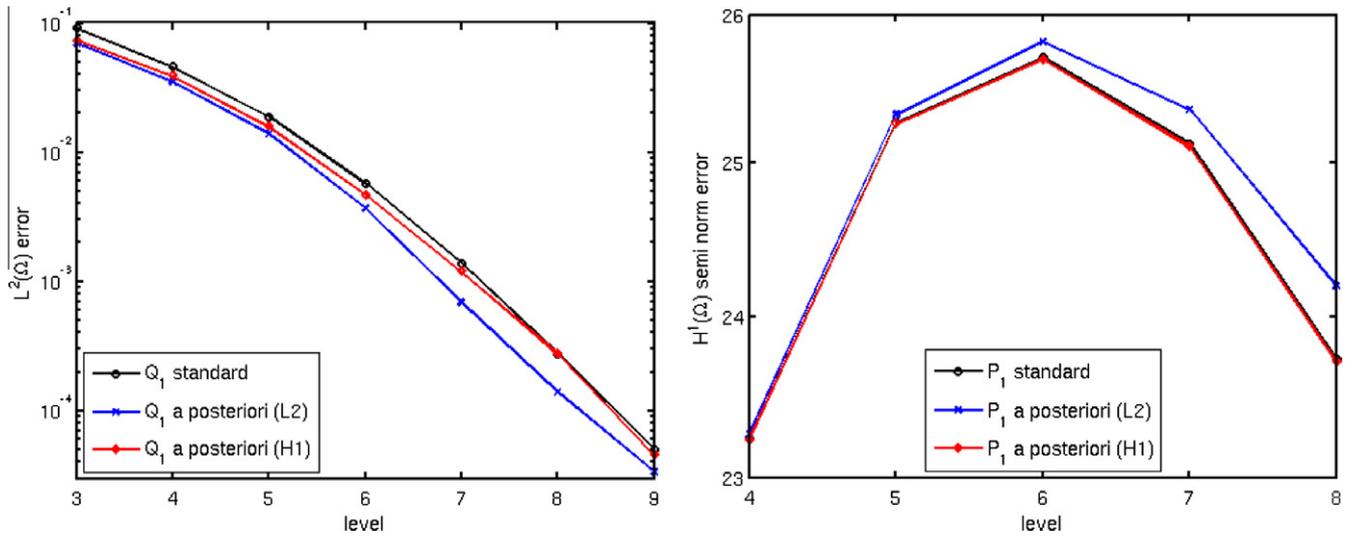


Fig. 6. Left: Example 5.1, Q_1 finite element and $L^2(\Omega)$ error; right: Example 5.2, P_1 finite element and $H^1(\Omega)$ semi norm error; comparison of the different parameter choices.

small. Because of the unresolved layers, in particular in Example 5.2, the error in the $H^1(\Omega)$ semi norm, computed with the above mentioned quadrature rules, even grows on coarse grids, compare Fig. 6.

Considering all three parameter choices (standard, a posteriori based on $L^2(\Omega)$ error, a posteriori based on $H^1(\Omega)$ semi norm error), one can observe that the optimization with respect to the error in one norm might reduce the error in the other norm, too, compared with the standard parameter choice. But the other error might also increase, see Fig. 6. Fig. 7 shows stabilization parameters and corresponding solutions with respect to the optimization of errors in different norms. Whereas the optimization of the $L^2(\Omega)$ error reduces the parameter in the boundary layers, the optimization with respect to the error in the $H^1(\Omega)$ semi norm increases the parameter in the layer at $x = 1$. The different effects on the computed solutions are clearly visible. In the $L^2(\Omega)$ error optimized solution, considerable spurious oscillations can be observed in the layers. They are even larger than in the solution computed with the standard parameter (7). The solution with $H^1(\Omega)$ semi norm error optimization looks much better. This comparison demonstrates already the importance of using an appropriate measure upon which the a posteriori selection of the parameter is based.

Altogether, the results presented in this section demonstrate that the proposed methodology is able to compute a stabilization parameter in the SUPG method in an a posteriori way such that solutions with reduced errors are obtained.

6. Parameter optimization with respect to functionals which are candidates for describing the quality of computed solutions

Generally, the evaluation of errors is not possible as the solution of (1) is not known. In this situation, other functionals are necessary to measure or estimate the quality of computed solutions.

On the first glance, a posteriori error estimators might be an appropriate choice. The construction of reliable error estimators with respect to global norms for convection-dominated problems is difficult. As demonstrated, e.g., in [23], the application of standard estimators for elliptic problems does not lead to reliable error predictions. The numerical studies presented below will consider a residual-based error estimator from [43]

$$I_h(w_h) = \sum_{K \in \mathcal{T}_h} \alpha_K^2 \| -\varepsilon \Delta w_h + \mathbf{b} \cdot \nabla w_h + c w_h - f \|_{0,K}^2 + \sum_{E \in \mathcal{O}K} \varepsilon^{-1/2} \alpha_E \| R_E(w_h) \|_{0,E}^2 \quad \forall w_h \in W_h \tag{20}$$

with

$$R_E(w_h) = \begin{cases} -[\varepsilon \mathbf{n}_E \cdot \nabla w_h]_E, & \text{if } E \notin \partial \Omega, \\ \mathbf{g} - \varepsilon \mathbf{n}_E \cdot \nabla w_h, & \text{if } E \subset \Gamma^N, \\ 0, & \text{if } E \subset \Gamma^D, \end{cases}$$

and

$$\alpha_K = \min \{ \text{diam}(K) \varepsilon^{-1/2}, c_0^{-1/2} \}, \quad \alpha_E = \min \{ \text{diam}(E) \varepsilon^{-1/2}, c_0^{-1/2} \}.$$

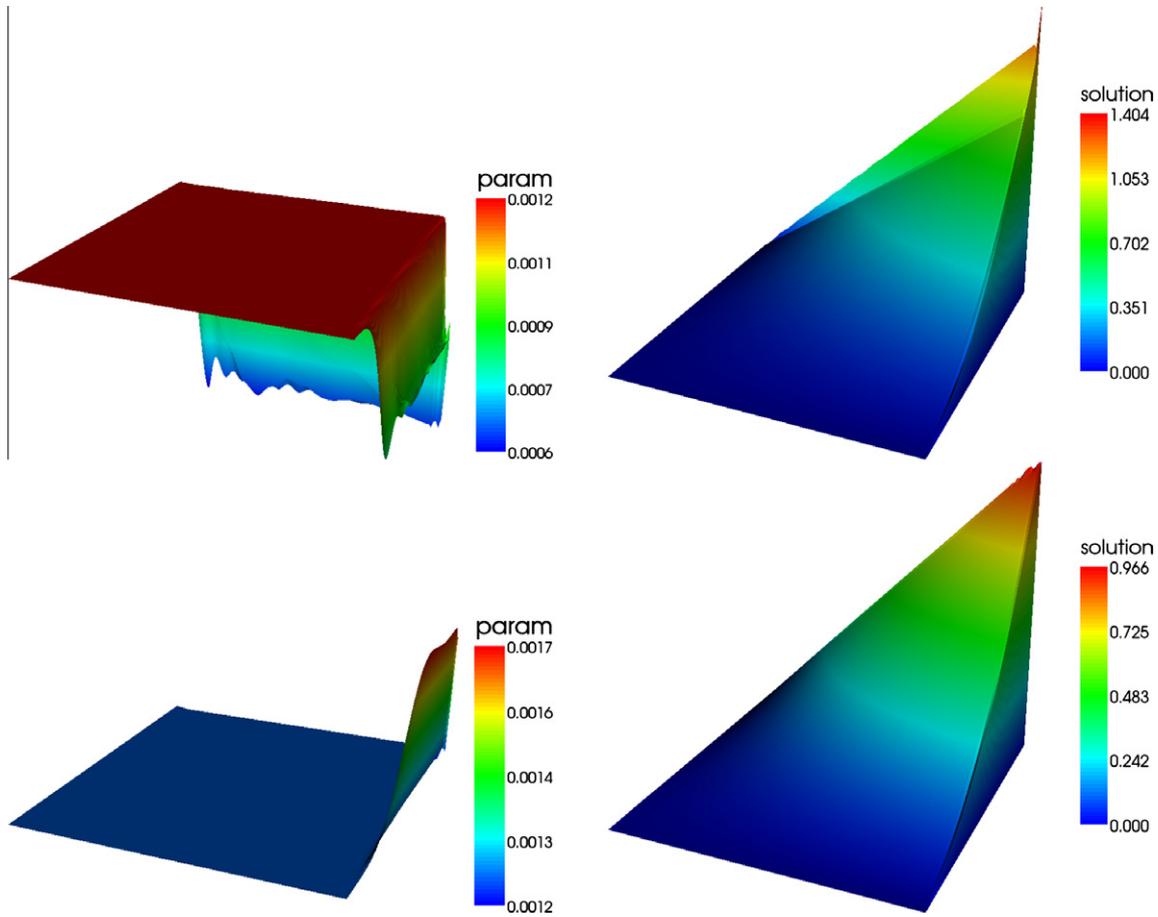


Fig. 7. Example 5.2, a posteriori defined stabilization parameters and computed solutions with the P_1 finite element; top: optimization with respect to the $L^2(\Omega)$ error; bottom: optimization with respect to the $H^1(\Omega)$ semi norm error; both at level 7 (parameter: visualization by projection to P_1 finite element).

Here, $\text{diam}(K)$ and $\text{diam}(E)$ denote the diameters of the mesh cell K and the face E , respectively, \mathbf{n}_E is a unit normal on E , and c_0 is defined in (2). The jump of a function across the face E is denoted by $[\![\cdot]\!]_E$. This error estimator is robust in a norm that is a sum of the standard energy norm and a dual norm of the convective derivative, see [43].

The right-hand side of the adjoint problem for the functional (20) is given by

$$\begin{aligned} \langle \tilde{D}I_h(\tilde{u}_h(y_h)), v_h \rangle &= \sum_{K \in \mathcal{T}_h} 2\alpha_K^2 (-\varepsilon \Delta u_h(y_h) + \mathbf{b} \cdot \nabla u_h(y_h) \\ &\quad + cu_h(y_h) - f, -\varepsilon \Delta v_h + \mathbf{b} \cdot \nabla v_h + cv_h)_K \\ &\quad + \sum_{E \subset \partial K} 2\varepsilon^{-1/2} \alpha_E (R_E(u_h(y_h)), \tilde{R}_E(v_h))_E, \end{aligned}$$

where $\tilde{R}_E(v_h)$ is $R_E(v_h)$ with $g = 0$.

Applying the estimator (20) as functional for the parameter optimization, it turns out that the global errors are dominated by the local contributions from the mesh cells in layers at the Dirichlet boundary. This effect comes from the nature of the underlying problem. For a local error estimate to be small, in particular the strong residual on a mesh cell (first term in (20)) has to be small. This cannot be achieved in mesh cells with boundary layers since the layers are not resolved. Even a nodally exact numerical solution leads to a large residual in those mesh cells. Thus, a significant reduction of the residual in such mesh cells is not possible. As the optimization algorithm concentrates on the reduction of the dominating errors, consequently, the errors in mesh cells away from the Dirichlet boundary are also not reduced notably. For this reason, an error indicator that excludes the mesh cells at the

Dirichlet boundary will be considered, too. Furthermore, we could observe that the influence of the residuals on the edges in (20) is negligible. One obtains practically the same results with and without using these terms. Thus, besides (20), the error indicator

$$I_h(w_h) = \sum_{K \in \mathcal{T}_h, \bar{K} \cap \Gamma^D = \emptyset} \alpha_K^2 \| -\varepsilon \Delta w_h + \mathbf{b} \cdot \nabla w_h + cw_h - f \|_{0,K}^2 \quad \forall w_h \in W_h \tag{21}$$

will be considered. Note, the mesh cells at the Dirichlet boundary do not contribute to the error indicator, but the stabilization parameter in these cells is still included into the optimization process.

The most serious drawback of using the SUPG method are the spurious oscillations that might appear in a vicinity of the layers. An optimization of the stabilization parameter should try above all to reduce them. These oscillations are connected to large derivatives of the computed solutions in crosswind direction. For this reason, a third functional that contains, besides the residual, also a control of the crosswind derivative will be included into the studies

$$\begin{aligned} I_h(w_h) &= \sum_{K \in \mathcal{T}_h, \bar{K} \cap \Gamma^D = \emptyset} \left(\| -\varepsilon \Delta w_h + \mathbf{b} \cdot \nabla w_h + cw_h - f \|_{0,K}^2 \right. \\ &\quad \left. + \|\phi(|\mathbf{b}^\perp \cdot \nabla w_h|)\|_{0,1,K} \right) \quad \forall w_h \in W_h, \end{aligned} \tag{22}$$

where

$$\mathbf{b}^\perp(\mathbf{x}) = \begin{cases} \frac{(b_2(\mathbf{x}), -b_1(\mathbf{x}))}{|\mathbf{b}(\mathbf{x})|}, & \text{if } \mathbf{b}(\mathbf{x}) \neq \mathbf{0}, \\ \mathbf{0}, & \text{if } \mathbf{b}(\mathbf{x}) = \mathbf{0}, \end{cases}$$

and

$$\phi(x) = \begin{cases} \sqrt{x}, & \text{if } x \geq 1, \\ 0.5(5x^2 - 3x^3), & \text{if } x < 1. \end{cases}$$

The special choice of $\phi(x)$ ensures that this functional is Fréchet differentiable. Its derivative can be computed in the usual way.

The numerical studies will consider a standard example, defined on the unit square, that is often used for the evaluation

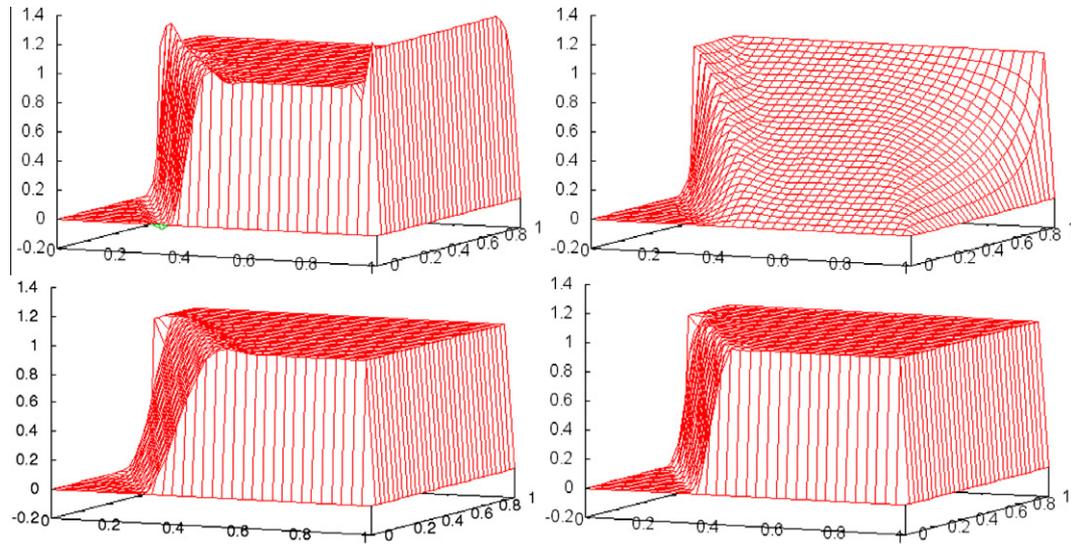


Fig. 8. Example 6.1: P_1 , level 5 (1089 d.o.f.), solution with standard parameter (7), minimization of (20), minimization of (21), and minimization of (22), left to right, top to bottom.

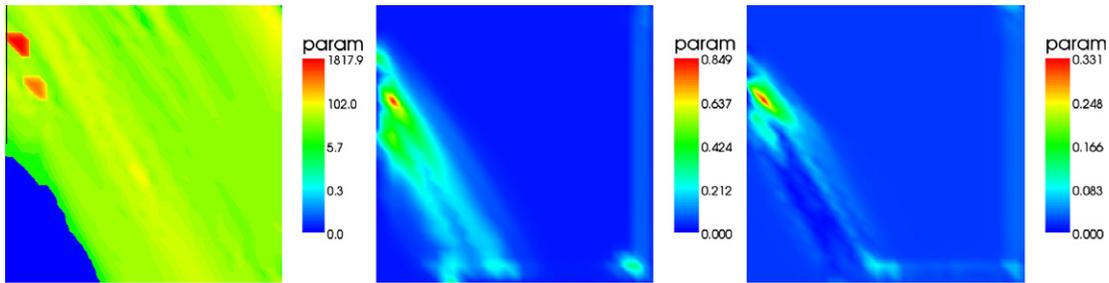


Fig. 9. Example 6.1: P_1 , level 5 (1089 d.o.f.), stabilization parameter (standard parameter (7) $y_n = 0.018042$), minimization of (20) (logarithmic scale), minimization of (21), and minimization of (22), left to right (visualization by projection to P_1 finite element).

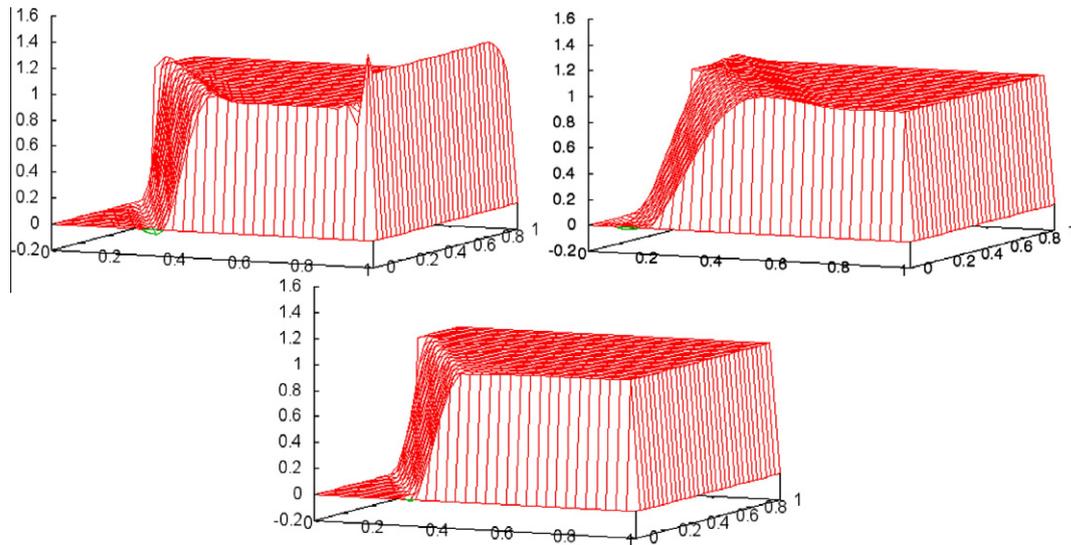


Fig. 10. Example 6.1: Q_1 , level 5 (1089 d.o.f.), solution with standard parameter (7), minimization of (21), and minimization of (22), left to right, top to bottom.

of stabilized methods, and an example in a more complicated domain that attracted some attention in the past years. Both examples have the properties $\text{div} \mathbf{b} = 0$, $c = 0$, such that the upper bound (9) for the stabilization parameter applies.

Example 6.1 (Example with interior and exponential boundary layers). This example was proposed in [22]. It is given by $\Omega = (0, 1)^2$, $\Gamma^D = \partial\Omega$, with the data $\varepsilon = 10^{-8}$, $\mathbf{b} = (\cos(-\pi/3), \sin(-\pi/3))^T$, $c = 0$, $f = 0$, and

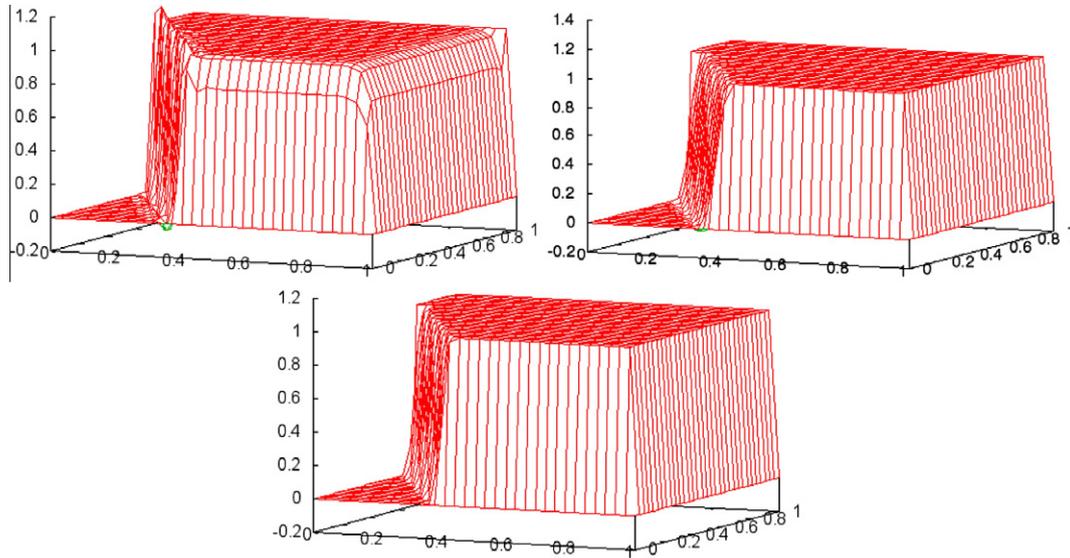


Fig. 11. Example 6.1: P_2 , level 5 (4225 d.o.f.), solution with standard parameter (7), minimization of (21), and minimization of (22), left to right, top to bottom (visualization by projection to P_1 finite element).

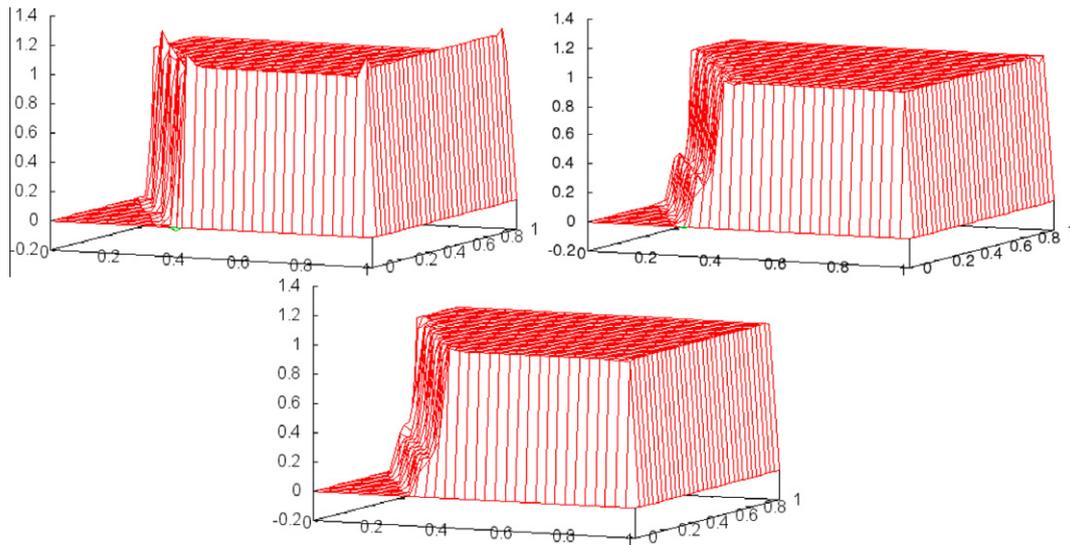


Fig. 12. Example 6.1: P_3 , level 5 (9409 d.o.f.), solution with standard parameter (7), minimization of (21), and minimization of (22), left to right, top to bottom (visualization by projection to P_1 finite element).

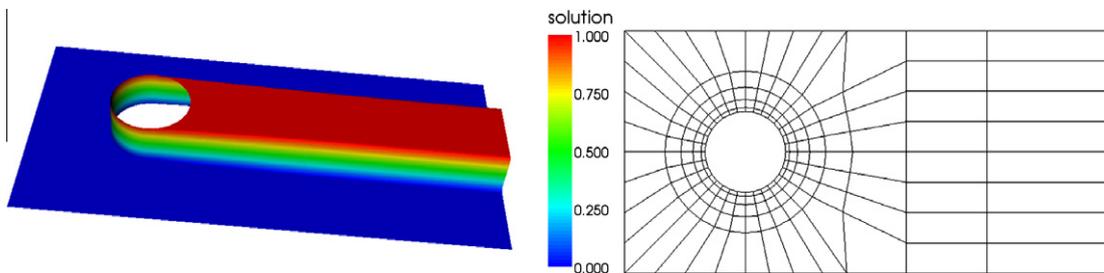


Fig. 13. Solution and initial grid for Example 6.2.

$$u_b(x,y) = \begin{cases} 0, & \text{for } x = 1 \text{ or } y \leq 0.7, \\ 1, & \text{else.} \end{cases}$$

The simulations were performed on Grid 2 and Grid 3 from Fig. 1. For shortness of presentation, only results on a rather coarse mesh are shown in Figs. 8–12. We could observe that the principal behavior for P_k and Q_k finite elements, with the same k , was always similar.

As already mentioned above, the minimization of (20) does not lead to useful results. This is demonstrated exemplarily for the P_1 finite element in Fig. 8. It can be seen that the spurious oscillations are removed but the layers are extremely smeared. The plot of the stabilization parameter in Fig. 9 shows that this is caused by very large values of this parameter (note the logarithmic scale in this picture). Minimizing (21) instead of (20) leads to a considerable improvement with respect to the extreme smearing. However, a notable smearing of the interior layer can still be observed. The

reason is the prediction of a rather large stabilization parameter in this layer, see Fig. 9. Nearly perfect solutions are obtained with the parameter choice based on minimizing (22). The spurious oscillations are almost removed, only around 2% are left. A large stabilization parameter is proposed in all layers, but its maximal value is smaller than the maximal value of the parameter computed with minimizing (21). Only the interior layer in the solution with the P_3 finite element is somewhat smeared. We think, the reason is the use of a piecewise constant stabilization parameter in this case. This polynomial degree of the parameter might not be sufficiently flexible for the changes of the finite element solution within a mesh cell that occur for higher order finite elements.

Example 6.2 (The Hemker example). This example was defined in [19]. The simulations were performed with $\Omega = \{(-3,8) \times (-3,3)\} \setminus \{(x,y); x^2 + y^2 \leq 1\}$, $\varepsilon = 10^{-6}$, $\mathbf{b} = (1,0)^T$, and $c = f = 0$. At

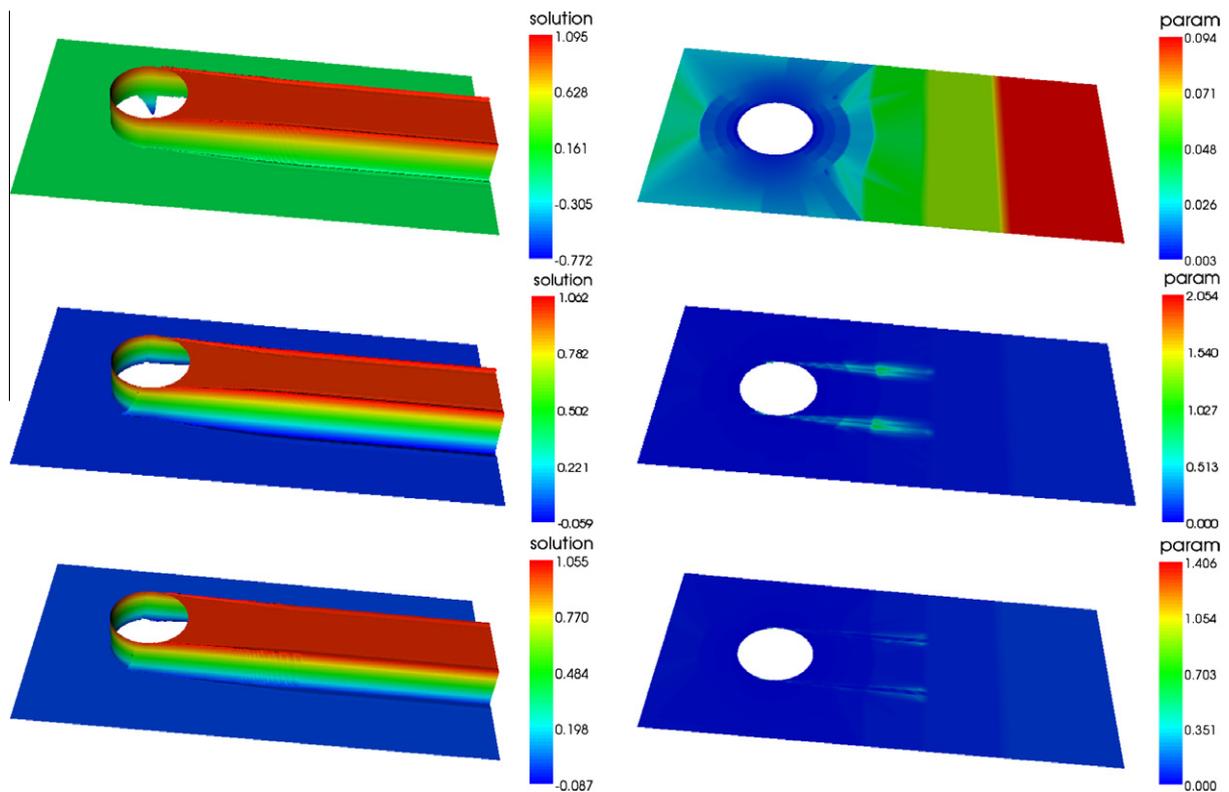


Fig. 14. Example 6.2: Q_1 , level 4 (47 664 d.o.f.), standard parameter (7), minimization of (21), and minimization of (22), top to bottom (stabilization parameter visualization by projection to Q_1 finite element).

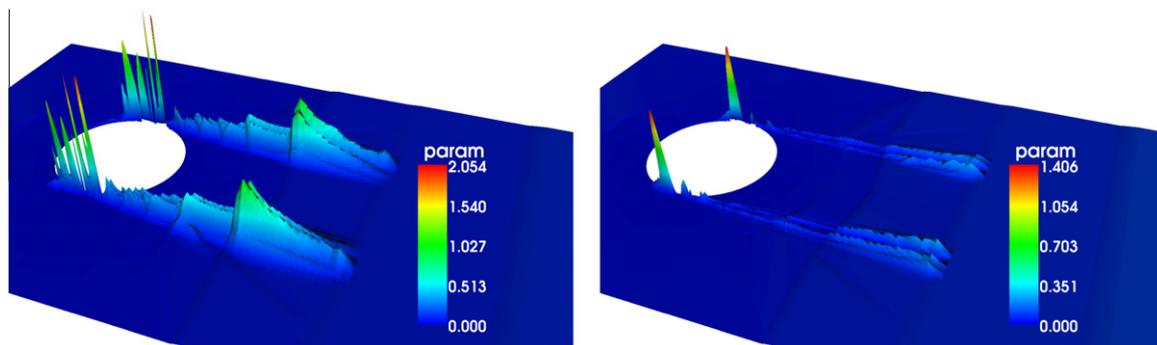


Fig. 15. Example 6.2: Q_1 , level 4 (47 664 d.o.f.), details of the stabilization parameter, minimization of (21), and minimization of (22), left to right (visualization by projection to Q_1 finite element).

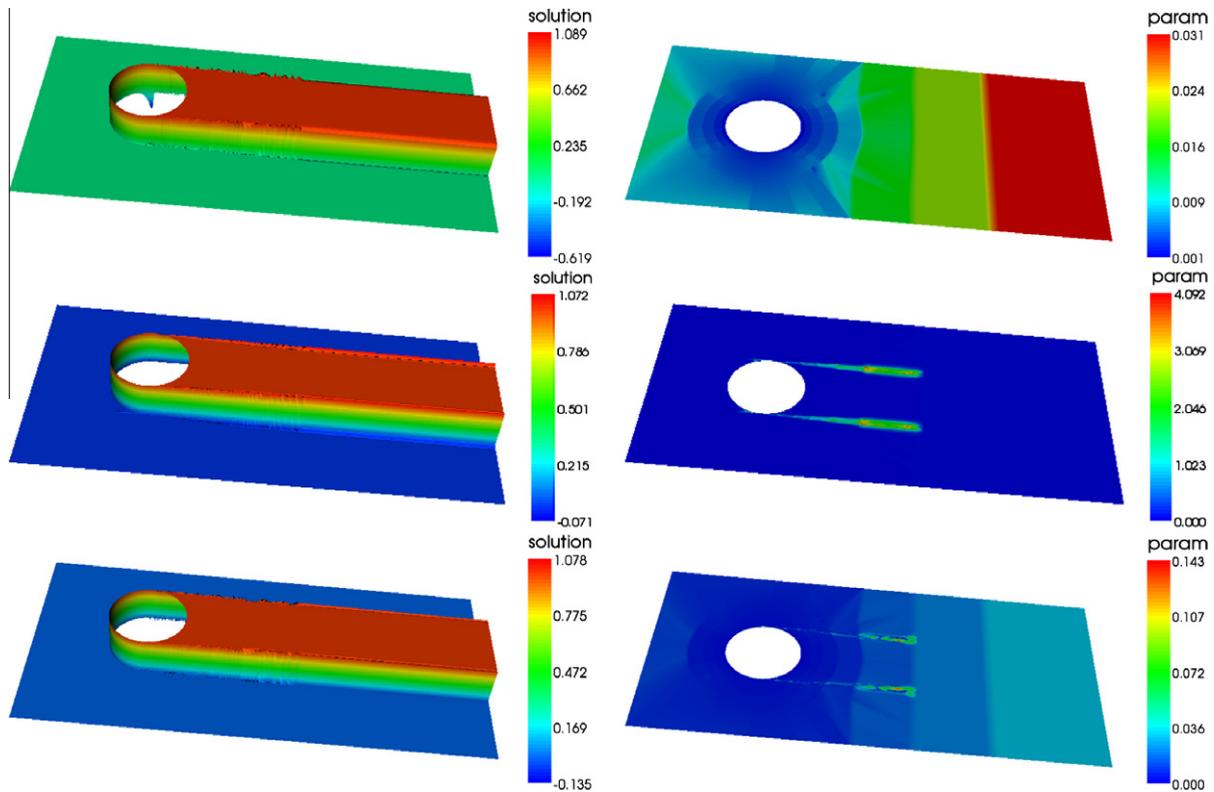


Fig. 16. Example 6.2: Q_3 , level 4 (425 616 d.o.f.), standard parameter (7), minimization of (21), and minimization of (22), top to bottom (visualization by projection to Q_1 finite element).

the inlet $x = -3$, a homogeneous Dirichlet boundary condition is prescribed, at the circle there is $u = 1$, and on all other parts of $\partial\Omega$, homogeneous Neumann boundary conditions are given.

This example attracted recently some interest [16,41] since it is considered to be closer to situations arising in applications than many usual test examples. It can be interpreted as a model of heat transfer from a hot column in the direction of the convection.

Results of numerical simulations are presented for the Q_1 , Q_2 , and Q_3 finite elements in Figs. 14–17. The initial grid (level 0) is shown in Fig. 13. Isoparametric finite elements were used to approximate the curved boundary. It can be seen that the most difficult regions for computing a correct solution are the starting points of the interior layers on top and on bottom of the circle. The SUPG method was applied with the standard parameter choice (7). Considerable negative spurious oscillations at the starting points of the interior layers can be observed for the solutions computed with this choice. Note that the values of the standard parameter are rather small in the vicinity of the circle due to small diameters of mesh cells in this region. Solutions obtained with the minimization of the error estimator (20) are not shown. Similarly as in the previous example, the layers are smeared very much, in particular the layer in front of the circle. For the Q_1 finite element, the minimization of the functional (21) reduces the negative spurious oscillations considerably, compared with the solution obtained with the standard parameter choice, see also Fig. 17. However, the solutions which are based on the minimization of this functional possess the wrong feature that the interior layers start somewhat before the top and bottom of the circle. This feature was reduced or even removed by minimizing (22) for the determination of the stabilization parameter. It can be observed that both, the minimization of (21) and the minimization of (22), lead to an increase of the parameter in the region of the interior layers, in particular at the starting points of the interior layers, cf. Fig. 15.

Since the large undershoots are a distinguished bad feature of the standard SUPG approach, Fig. 17 shows the size of the undershoots obtained in the simulations. For the Q_1 finite element, the parameter choices based on the minimization of (21) and (22) reduce these undershoots on all levels considerably. The situation is different for the Q_2 and Q_3 finite element, where only the minimization of (22) leads to smaller undershoots on most levels. A reason for not observing this on all levels might be the insufficient flexibility of using a piecewise constant stabilization parameter for a higher order finite element, see the discussion at the end of Example 6.1. The overshoots are much less pronounced than the undershoots. They are similar for all simulations, between 0.05 and 0.15.

Altogether, the parameter choice based on the minimization of (22) gave the best results among the considered approaches. However, these results are not yet optimal.

In the optimization process, always a fast decrease of the functionals within the first steps could be observed. To fulfill the stopping criterion formulated in Section 4, in general some dozens to a few hundred L-BFGS steps were necessary. As could be seen in the presented examples, the values of the stabilization parameter have very little effect on the solution in smooth regions and hence varying them has also little influence on the target functional. This observation offers a way for a possible improvement of the efficiency of the algorithm by identifying in the first few steps the values of the stabilization parameter which are important for the decrease of the functional and then restricting the optimization process to those values.

7. Summary and outlook

This paper presented a general framework for optimizing parameters in stabilized finite element methods for convection–diffusion

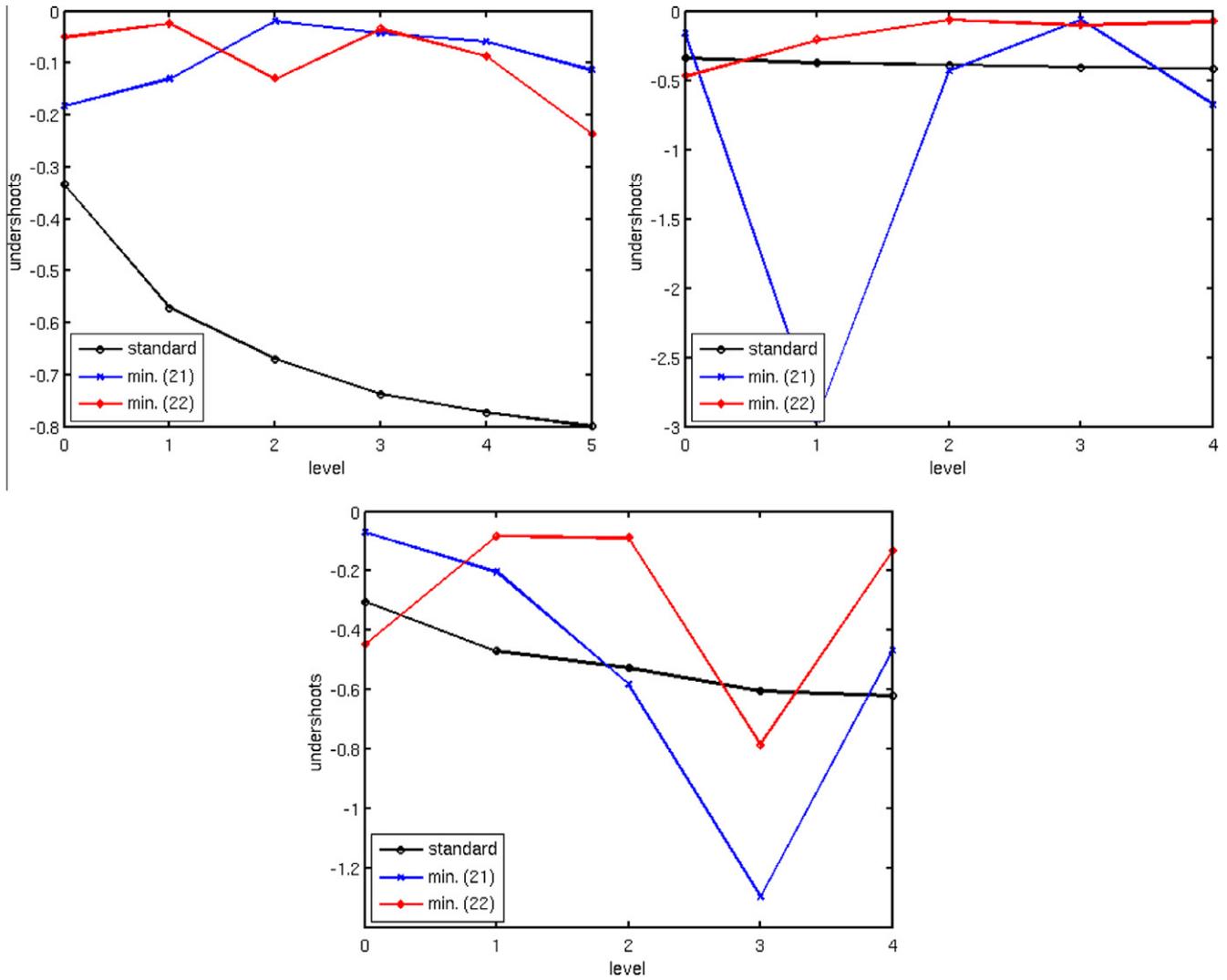


Fig. 17. Example 6.2: Undershoots of the computed solutions, Q_1 finite element, Q_2 finite element, and Q_3 finite element, left to right, top to bottom.

problems. The optimization is based on minimizing a target functional that indicates the quality of the computed solution. The L-BFGS method is used to solve the arising constrained optimization problem. Key of the algorithm is the efficient evaluation of the derivative of the target functional with respect to the stabilization parameter that utilizes the solution of an appropriate adjoint problem. Benefits and difficulties of this basic approach were studied exemplarily at the SUPG finite element method and three different functionals. A main observation is that a straightforward choice, a residual-based a posteriori error estimator, is not appropriate for measuring the quality of computed solutions. A better functional could be found, (22), but the results obtained with this functional are not yet optimal.

Important next steps in the exploration and improvement of the parameter optimization are as follows:

- A very important goal consists in identifying better functionals than used in this manuscript. The chosen functional is the main component of the algorithm that determines the quality of the computed solutions.
- It is known that the introduction of diffusion in streamline direction only, as in the SUPG method, is often not sufficient to obtain satisfactory numerical solutions. Some diffusion orthogonal to the streamlines (in crosswind direction) might

be necessary, as it is done by SOLD methods [27]. A new aspect in the application of the general framework to SOLD methods consists in the optimization of two stabilization parameters.

- Algorithmic improvements are possible. These include, e.g., the restriction of the optimization to important values of the stabilization parameter as discussed at the end of Section 6.
- The considerable decrease of the functionals within the first few optimization steps suggest that the improvement of the solutions occurs mainly also within these steps. This effect will be studied in detail, leading hopefully to an efficient method for just improving (but not optimizing) standard SUPG solutions.

References

- [1] I. Akkerman, K.G. van der Zee, S.J. Hulshoff, A variational Germano approach for stabilized finite element methods, *Comput. Methods Appl. Mech. Engrg.* 199 (2010) 502–513.
- [2] W. Bangerth, R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Lectures in Mathematics, ETH Zürich. Birkhäuser, Basel, 2003.
- [3] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, in: A. Iserles (Ed.), *Acta Numerica*, Cambridge University Press, 2004, pp. 1–102.
- [4] A.N. Brooks, 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 (1982) 199–259.

- [5] C.G. Broyden, The convergence of a class of double-rank minimization algorithms 1. General considerations, *J. Inst. Math. Appl.* 6 (1970) 76–90.
- [6] P.G. Ciarlet, Basic error estimates for elliptic problems, in: P.G. Ciarlet, J.L. Lions (Eds.), *Handbook of Numer. Anal.*, v. 2 – Finite Elem. Methods (pt. 1), North-Holland, Amsterdam, 1991, pp. 17–351.
- [7] R. Codina, E. Oñate, M. Cervera, The intrinsic time for the streamline upwind/Petrov-Galerkin formulation using quadratic elements, *Comput. Methods Appl. Mech. Engrg.* 94 (1992) 239–262.
- [8] R. Fletcher, A new approach to variable metric algorithms, *Comput. J.* 13 (1970) 317–322.
- [9] L.P. Franca, S.L. Frey, T.J.R. Hughes, Stabilized finite element methods: I. Application to the advective-diffusive model, *Comput. Methods Appl. Mech. Engrg.* 95 (1992) 253–276.
- [10] C. Führer, R. Rannacher, An adaptive streamline-diffusion finite element method for hyperbolic conservation laws, *East-West J. Numer. Math.* 5 (1997) 145–162.
- [11] A.C. Galeão, R.C. Almeida, S.M.C. Malta, A.F.D. Loula, Finite element analysis of convection dominated reaction-diffusion problems, *Appl. Numer. Math.* 48 (2004) 205–222.
- [12] M. Germano, U. Piomelli, P. Moin, W. Cabot, A dynamic subgrid-scale eddy viscosity model, *Phys. Fluids A* 3 (1991) 1760–1765.
- [13] M.B. Giles, N.A. Pierce, An introduction to the adjoint approach to design, *Flow, Turbulence Combust.* 65 (2000) 393–415.
- [14] D. Goldfarb, A family of variable metric updates derived by variational means, *Math. Comput.* 24 (1970) 23–26.
- [15] J.-L. Guermond, Stabilization of Galerkin approximations of transport equations by subgrid modeling, *M2AN* 33 (1999) 1293–1316.
- [16] H. Han, Z. Huang, R.B. Kellogg, A tailored finite point method for a singular perturbation problem on an unbounded domain, *J. Sci. Comput.* 36 (2008) 243–261.
- [17] I. Harari, T.J.R. Hughes, What are c and h ? : inequalities for the analysis and design of finite element methods, *Comput. Methods Appl. Mech. Engrg.* 97 (1992) 157–192.
- [18] F.K. Hebeker, R. Rannacher, An adaptive finite element method for unsteady convection-dominated flows with stiff source terms, *SIAM J. Sci. Comput.* 21 (1999) 799–818.
- [19] W.P. Hemker, A singularly perturbed model problem for numerical computation, *J. Comput. Appl. Math.* 76 (1996) 277–285.
- [20] T.J.R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid-scale models, bubbles and the origin of stabilized methods, *Comput. Methods Appl. Mech. Engrg.* 127 (1995) 387–401.
- [21] T.J.R. Hughes, A.N. Brooks, A multidimensional upwind scheme with no crosswind diffusion, in: T.J.R. Hughes (Ed.), *Finite Element Methods for Convection Dominated Flows*, AMD, vol. 34, ASME, New York, 1979, pp. 19–35.
- [22] T.J.R. Hughes, M. Mallet, A. Mizukami, A new finite element formulation for computational fluid dynamics: II. Beyond SUPG, *Comput. Methods Appl. Mech. Engrg.* 54 (1986) 341–355.
- [23] V. John, A numerical study of a posteriori error estimators for convection-diffusion equations, *Comput. Methods Appl. Mech. Engrg.* 190 (2000) 757–781.
- [24] V. John, Large eddy simulation of turbulent incompressible flows, analytical and numerical results for a class of LES models, *Lecture Notes in Computational Science and Engineering*, vol. 34, Springer-Verlag, Berlin, Heidelberg, New York, 2004.
- [25] V. John, S. Kaya, A finite element variational multiscale method for the Navier-Stokes equations, *SIAM J. Sci. Comput.* 26 (2005) 1485–1503.
- [26] V. John, A. Kindl, A variational multiscale method for turbulent flow simulation with adaptive large scale space, *J. Comput. Phys.* 229 (2010) 301–312.
- [27] V. John, P. Knobloch, A comparison of spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations: part I – a review, *Comput. Methods Appl. Mech. Engrg.* 196 (2007) 2197–2215.
- [28] V. John, P. Knobloch, A comparison of spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations: part II – analysis for P_1 and Q_1 finite elements, *Comput. Methods Appl. Mech. Engrg.* 197 (2008) 1997–2014.
- [29] V. John, G. Matthies, MoonMMD – a program package based on mapped finite element methods, *Comput. Visual. Sci.* 6 (2004) 163–170.
- [30] V. John, J.M. Maubach, L. Tobiska, Nonconforming streamline-diffusion-finite-element-methods for convection-diffusion problems, *Numer. Math.* 78 (1997) 165–188.
- [31] V. John, E. Schmeyer, Stabilized finite element methods for time-dependent convection-diffusion-reaction equations, *Comput. Methods Appl. Mech. Engrg.* 198 (2008) 475–494.
- [32] O.A. Ladyzhenskaya, New equations for the description of motion of viscous incompressible fluids and solvability in the large of boundary value problems for them, *Proc. Steklov Inst. Math.* 102 (1967) 95–118.
- [33] D.K. Lilly, A proposed modification of the Germano subgrid-scale closure method, *Phys. Fluids A* 4 (1992) 633–635.
- [34] J. Nocedal, S.J. Wright, *Numerical optimization*, in: Springer Series in Operations Research and Financial Engineering, second ed., Springer, 2006.
- [35] A.A. Oberai, J. Wanderer, A dynamic approach for evaluating parameters in a numerical method, *Int. J. Numer. Methods Engrg.* 62 (2005) 50–71.
- [36] E. Oñate, Derivation of stabilized equations for numerical solutions of advective-diffusive transport and fluid flow problems, *Comput. Methods Appl. Mech. Engrg.* 151 (1998) 233–265.
- [37] H.-G. Roos, M. Stynes, L. Tobiska, *Robust Numerical Methods for Singularly Perturbed Differential Equations, Convection-Diffusion-Reaction and Flow Problems*, second ed., Springer-Verlag, Berlin, 2008.
- [38] R. Schneider, *Applications of the Discrete Adjoint Method in Computational Fluid Dynamics*, PhD thesis, University of Leeds, School of Computing, 2006.
- [39] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, *Math. Comput.* 24 (1970) 647–655.
- [40] J.S. Smagorinsky, General circulation experiments with the primitive equations, *Mon. Weather Rev.* 91 (1963) 99–164.
- [41] P. Sun, L. Chen, J. Xu, Numerical studies of adaptive finite element methods for two dimensional convection-dominated problems, *J. Sci. Comput.* 43 (2010) 24–43.
- [42] F. Tröltzsch, *Optimale Steuerung partieller Differentialgleichungen - Theorie, Verfahren und Anwendungen*, second ed., Vieweg+Teubner Verlag, 2009.
- [43] R. Verfürth, Robust a posteriori error estimates for stationary convection-diffusion equations, *SIAM J. Numer. Anal.* 43 (2005) 1766–1782.