# A posteriori optimization of parameters in the SUPG method for higher degree FE spaces

Petr Lukáš

**Abstract** This paper is devoted to the numerical solution of the scalar convection–diffusion–reaction equation. We present new results of the adaptive technique for computing the stabilization parameter $\tau$ in the streamline upwind/Petrov–Galerkin (SUPG) method based on minimizing the value of a functional called error indicator. Particularly, we present results for conforming finite element spaces up to the order 5 with the parameter $\tau$ from the piecewise discontinuous finite element spaces, also up to the order 5.

## 1 Introduction

We are seeking the solution of the scalar convection–diffusion–reaction problem

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

Here $\Omega \subset \mathbb{R}^2$ is a bounded domain with a polygonal Lipschitz–continuous boundary $\partial\Omega$ and $\Gamma^D$, $\Gamma^N$ are disjoint and relatively open subsets of $\partial\Omega$ satisfying $\mathrm{meas}_1(\Gamma^D) > 0$ and $\overline{\Gamma^D \cup \Gamma^N} = \partial\Omega$. Furthermore, $\mathbf{n}$ is the outward unit normal vector to $\partial\Omega$, $\varepsilon > 0$ is a constant diffusivity, $\mathbf{b} \in W^{1,\infty}(\Omega)^2$ 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. We make the usual assumption that

$$c - \tfrac{1}{2}\,\mathrm{div}\,\mathbf{b} \geq 0\,. \quad (2)$$

Petr Lukáš

Charles University in Prague, Faculty of Mathematics and Physics, Department of Numerical Mathematics, Sokolovská 83, 186 75 Praha 8, Czech Republic, e-mail: lukas@karlin.mff.cuni.cz

A very important aspect of the numerical solution of (1) are spurious oscillations which often appear in the discrete solution when convection dominates diffusion and standard discretizations are used. Various stabilized methods have been proposed. These methods often depend on parameters whose optimal choice is usually not known. Only bounds for the values of these parameters are derived for some of these methods. Paper [5] describes how these parameters can be optimized in an adaptive way. In this paper we enrich the space of parameters and also the finite element space to get new results. We also provide experimental convergence rates.

We will use the standard notation for usual function spaces and norms, see, e.g., [2]. The notation $(\cdot,\cdot)_G$ is used for the inner product in the space $L^2(G)$ or $L^2(G)^2$ and we set $(\cdot,\cdot) = (\cdot,\cdot)_\Omega$.

## 2 Weak formulation

Let $\widetilde{u}_b \in H^1(\Omega)$ be an extension of $u_b$ (i.e., the trace of $\widetilde{u}_b$ equals $u_b$ on $\Gamma^D$) and let

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

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

$$a(u,v) = (f,v) + (g,v)_{\Gamma^N} \qquad \forall\, v \in V, \tag{3}$$

where $a(u,v)$ is the usual bilinear form

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

From the assumption (2) it follows that the weak formulation has a unique solution.

## 3 Galerkin finite element discretization

Let $\left\{ \mathscr{T}_h \right\}_h$ be a family of triangulations of $\Omega$ parametrized by positive parameters $h$ whose only accumulation point is zero. The triangulations $\mathscr{T}_h$ are assumed to consist of a finite number of open polygonal subsets $K$ of $\Omega$ such that $\overline{\Omega} = \bigcup_{K \in \mathscr{T}_h} \overline{K}$ and the closures of any two different sets in $\mathscr{T}_h$ are either disjoint or possess either a common vertex or a common edge. Further, we assume that any edge of $\mathscr{T}_h$ which lies on $\partial\Omega$ is contained either in $\overline{\Gamma^D}$ or in $\overline{\Gamma^N}$.

For each $h$, we introduce a finite element space $W_h \subset H^1(\Omega)$ defined on $\mathscr{T}_h$ and approximating the space $H^1(\Omega)$ in the usual sense, see [2]. Furthermore, for each $h$, we introduce a function $\widetilde{u}_{bh} \in W_h$ whose trace on $\Gamma^D$ 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 $u_h - \widetilde{u}_{bh} \in V_h$ and

$$a(u_h, v_h) = (f, v_h) + (g, v_h)_{\Gamma^N} \qquad \forall\, v_h \in V_h. \tag{4}$$

## 4 SUPG stabilization

It is well known that the Galerkin discretization (4) is inappropriate if convection dominates diffusion, since then the discrete solution is usually globally polluted by spurious nonphysical 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 streamline upwind/Petrov–Galerkin (SUPG) method [1], also called streamline diffusion finite element method (SDFEM), which is frequently used because of its stability properties and higher-order accuracy.

The SUPG stabilization depends on a stabilization parameter which will be denoted by $\tau_h$ in the following. We assume that all admissible stabilization parameters form a set $Y_h \subset L^\infty(\Omega)$. The SUPG discretization of (1) reads: Find $u_h \in W_h$ such that $u_h - \widetilde{u}_{bh} \in V_h$ and

$$a(u_h, v_h) + s_h(\tau_h; u_h, v_h) = (f, v_h) + (g, v_h)_{\Gamma^N} + r_h(\tau_h; v_h) \qquad \forall\, v_h \in V_h, \tag{5}$$

where

$$s_h(\tau_h; u_h, v_h) = (-\varepsilon \Delta_h u_h + \mathbf{b} \cdot \nabla u_h + c\, u_h, \tau_h\, \mathbf{b} \cdot \nabla v_h),$$
$$r_h(\tau_h; v_h) = (f, \tau_h\, \mathbf{b} \cdot \nabla v_h).$$

The SUPG method requires that the functions from $W_h$ are $H^2$ on each element of $\mathscr{T}_h$, which is satisfied for common finite element spaces. The notation $\Delta_h$ denotes the Laplace operator defined elementwise.

The parameter $\tau_h$ is often defined, on an element $K \in \mathscr{T}_h$, by the formula

$$\tau_h|_K = \frac{h_K}{2\,|\mathbf{b}|}\, \xi(Pe_K) \quad \text{with} \quad \xi(\alpha) = \coth \alpha - \frac{1}{\alpha}, \quad Pe_K = \frac{|\mathbf{b}|\, h_K}{2\,\varepsilon}, \tag{6}$$

where $h_K$ is the element diameter in the direction of the convection vector $\mathbf{b}$, $|\cdot|$ is the Euclidean norm, and $Pe_K$ is the local Péclet number which determines whether the problem is locally (i.e., within a particular element) convection dominated or diffusion dominated.

## 5 Optimization of parameters

Let $D_h \subset Y_h$ be an open set such that, for any $\tau_h \in D_h$, the SUPG method (5) has a unique solution $u_h \in W_h$. To emphasize that $u_h$ depends on $\tau_h$, we shall write $u_h(\tau_h)$ instead of $u_h$ in the following. Let $I_h : W_h \to \mathbb{R}$ be an error indicator, i.e.,

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

represents a measure of the error of the discrete solution $u_h(\tau_h)$ corresponding to a given parameter $\tau_h$.

We use two different indicators in our tests (not all reported here) proposed by [5]. The first indicator is based on the value of the residue. It is defined as

$$I_h(w_h) = \sum_{K \in \mathscr{T}_h, \overline{K} \cap \partial\Omega = \emptyset} h_K^2 \; \| - \varepsilon \Delta w_h + \mathbf{b} \cdot \nabla w_h + cw_h - f \|_{0,K}^2 \qquad \forall w_h \in W_h \qquad (7)$$

and the second indicator which is described as an "indicator with crosswind derivative control term" is given by

$$I_h(w_h) = \sum_{K \in \mathscr{T}_h, \overline{K} \cap \partial\Omega = \emptyset} \left( \| - \varepsilon \Delta w_h + \mathbf{b} \cdot \nabla w_h + cw_h - f \|_{0,K}^2 + \|\phi(|\mathbf{b}^\perp \cdot \nabla w_h|)\|_{0,1,K} \right) \quad (8)$$

for all $w_h \in W_h$, where $\mathbf{b} = (b_1, b_2)$ and

$$\mathbf{b}^\perp(x) = \begin{cases} \frac{(b_2(x), -b_1(x))}{|\mathbf{b}(x)|} & \text{if } \mathbf{b}(x) \neq 0, \\ 0 & \text{if } \mathbf{b}(x) = 0, \end{cases}, \quad \phi(t) = \begin{cases} \sqrt{t} & \text{if } t \geq 1, \\ 0.5(5t^2 - 3t^3) & \text{if } t < 1, \end{cases} \quad (9)$$

and $\| \cdot \|_{0,1,K}$ is the usual $L^1$ norm on an element $K$ in $\mathscr{T}_h$.

For the derivative of Indicator (7) we have

$$\langle D\tilde{I}_h(\tilde{u}_h(\tau_h)), v_h \rangle = \sum_{K \in \mathscr{T}_h, \overline{K} \cap \partial\Omega = \emptyset} h_K^2 \, 2 \, (\mathscr{L}u_h(\tau_h) - f, \mathscr{L}v_h)_K \qquad \forall \, v_h \in V_h, \quad (10)$$

where $\mathscr{L} = -\varepsilon \Delta + \mathbf{b} \cdot \nabla + c$, $\tilde{I}_h(w_h) = I_h(w_h + \tilde{u}_{bh})$ for any $w_h \in V_h$, and $\tilde{u}_h = u_h - \tilde{u}_{bh}$.

For the derivative of Indicator (8) we have

$$\langle D\tilde{I}_h(\tilde{u}_h(\tau_h)), v_h \rangle = \sum_{K \in \mathscr{T}_h, \overline{K} \cap \partial\Omega = \emptyset} \Big( 2(\mathscr{L}u_h(\tau_h) - f, \mathscr{L}v_h)_K$$
$$+ \int_K \mathrm{sgn}(\mathbf{b}^\perp \cdot \nabla u_h(\tau_h))\phi'(|\mathbf{b}^\perp \cdot \nabla u_h(\tau_h)|) \, \mathbf{b}^\perp \cdot \nabla v_h \, dx \Big) \quad (11)$$

for all $v_h \in V_h$. Our aim is to compute a parameter $\tau_h \in D_h$ for which $\Phi_h$ attains its minimum. To this end, it is convenient to compute effectively the Fréchet derivative of $\Phi_h$. This is done by the technique of adjoint approach described in the detail in [5] or [8].

## 6 Numerical results

We consider piecewise polynomial spaces up to degree 5 which are globally continuous (Lagrange finite elements) for the discrete solution. In the following text, we use abbreviations for these spaces derived from the FEniCS [7] software `CG1 − CG5`. This defines the spaces $W_h$ and $V_h = W_h \cap V$.

Based on our former observations in numerical experiments, we have chosen the spaces $Y_h$ for the parameter $\tau_h$ as piecewise discontinuous, to name these spaces we use again the abbreviations from the FEniCS software `DG0 − DG5`. So in addition to the classical choice for the space of parameter $\tau_h$ with $P_0(K)$ for all $K \in \mathcal{T}_h$ (which is the space `DG0` in our notation), we consider spaces of discontinuous piecewise $P_k$ functions, $k = 1, \ldots, 5$.

**Example 1:** As the first example, let us consider Equation (1) in $\Omega = (0,1)^2$ with $\varepsilon = 10^{-8}$, $\mathbf{b} = (-y, x)^T$, $c = 0$, $f = 0$, Neumann condition $\frac{\partial u}{\partial \mathbf{n}} = 0$ on $x = 0$, and

$$u_b = \begin{cases} 1 & \text{if } \frac{1}{3} \leq x \leq \frac{2}{3} \text{ and } y = 0, \\ 0 & \text{otherwise, if } x \neq 0. \end{cases} \tag{12}$$

**Example 2:** The second example has the setup used by [4]. Equation (1) is considered in $\Omega = (0,1)^2$ with $\varepsilon = 10^{-8}$, $\mathbf{b} = (1,0)^T$,

$$f = \begin{cases} 0 & \text{if } |x - 0.5| \geq 0.25 \text{ or } |y - 0.5| \geq 0.25, \\ -32(x - 0.5) & \text{otherwise,} \end{cases} \tag{13}$$
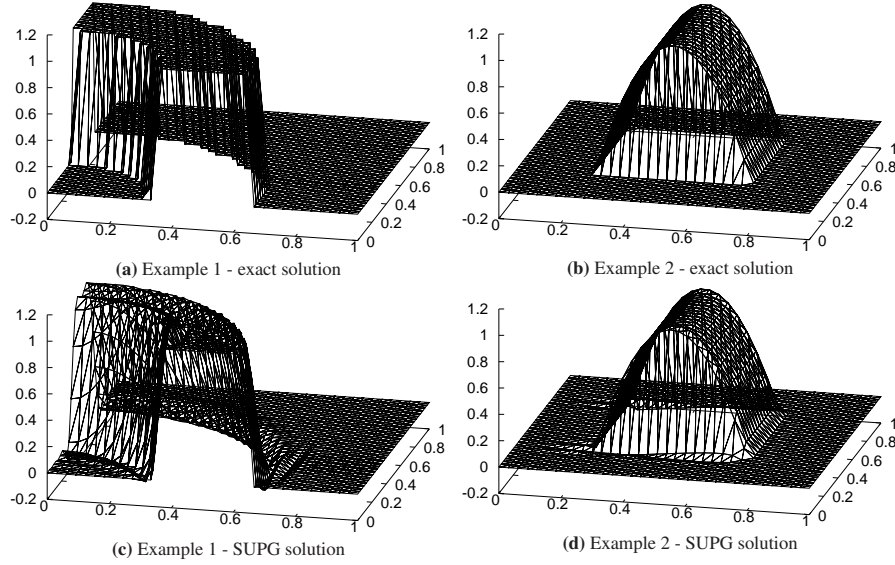


**(a)** Example 1 - exact solution

**(b)** Example 2 - exact solution

**(c)** Example 1 - SUPG solution

**(d)** Example 2 - SUPG solution

**Fig. 1** Interpolations of exact solutions and solutions of the SUPG method with $\tau_h$ defined in (6).

and $u_b = 0$ on $\partial\Omega$. In the following we will refer to the first example as Example 1 and to the second example as Example 2.

The interpolation of the exact solution of Example 1 in the `CG1` space is depicted in Figure 1a on the previous page. The interpolation of Example 2 in the `CG1` space is in Figure 1b. The Lagrange interpolation in $W_h$ of the exact solution $u(x,y)$ of a problem under consideration is denoted as $u_e$ where necessary.

The solution of Example 1 possesses two interior characteristic layers in the direction of the convection starting at $(\frac{1}{3},0)$ and $(\frac{2}{3},0)$. These interior layers are generally not aligned with the direction of elements' sides.

The solution of Example 2 possesses two interior characteristic layers in the direction of the convection starting at $(0.25,0.25)$ and $(0.25,0.75)$. This also means that the resulting discrete solution can be strongly influenced by the choice of the mesh, particularly by alignment of elements' sides.

If not said otherwise we provide results on a structured mesh of Friedrichs–Keller type with 34 nodes in each direction.

In both examples we consider in the paper the Péclet number from (6) is of the order $10^6$. We use Indicator (8) in all of the following numerical tests.

In Figures 1c and 1d on the previous page we show solutions of the SUPG method for the `CG1` FE space where the parameter $\tau_h$ is defined in (6) and in this case we choose $\tau_h$ from the space `DG0`.

The parameter $\tau_h$ is optimized by the L-BFGS-B nonlinear minimization method described in [10] using the default setup from `scipy` library with `gtol: 1e-14` and `ftol: 1e-14`. The method starts with the values given by (6).

We do not provide any figure which would show just the values of the parameter $\tau_h$ from (6), which is the starting point of the minimization procedure. Such an image would be not interesting as the values of $\tau_h$ for both examples



**(a)** FE space `CG1`



**(b)** FE space `CG2`



**(c)** FE space `CG3`



**(d)** FE space `CG4`

**Fig. 2** Optimized parameter $\tau_h$ for different FE spaces of the discrete solution, $\tau_h$ is from the space `DG1`, Ex. 1

we use are (almost) constant on the whole domain. The quality of discrete solutions

**(a)** $\tau_h$ from DG0     **(b)** $\tau_h$ from DG1     **(c)** $\tau_h$ from DG2     **(c)** $\tau_h$ from DG3
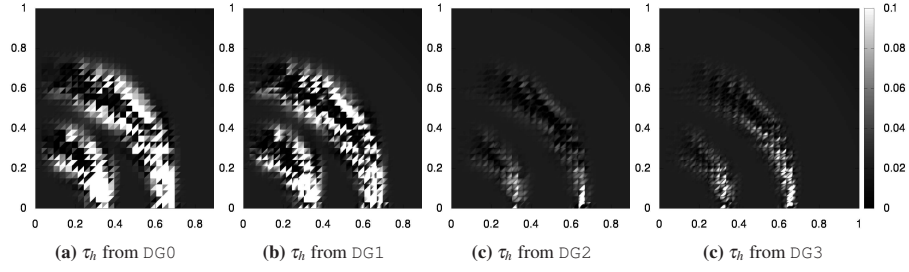
**Fig. 3** Changing the space for the parameter $\tau_h$ and preserving the finite element space CG3 for the discrete solution of Example 1.

after the minimization procedure is so good that images of such a discrete solution would be very similar to images 1a and 1b and this is the reason we do not involve them in the paper. In addition, the quality of discrete solutions after the minimization procedure has been studied in [5] and [8].

The values of the parameter $\tau_h$ after the whole minimization procedure are in Figures 2 and 3. In Figure 2 on the previous page we fix the FE space for the parameter $\tau_h$ and change the FE space of the discrete solution. On the other hand we fix the FE space of the discrete solution in Figure 3 and change the FE space of the parameter $\tau_h$. We can see that higher values of the parameter $\tau_h$ are at places where oscillations in the SUPG method with $\tau_h$ from (6) appear.

We provide in Figure 4a the value of the Indicator (8) after the 30 seconds run of the minimization procedure and in the end of the whole minimization procedure. We can see from the image that using finer FE spaces is not efficient to obtain lower value of an indicator.

We provide numerical convergence results in Figure 4b where we changed the mesh step by step from 8 elements in each direction to 100 elements in each direc-
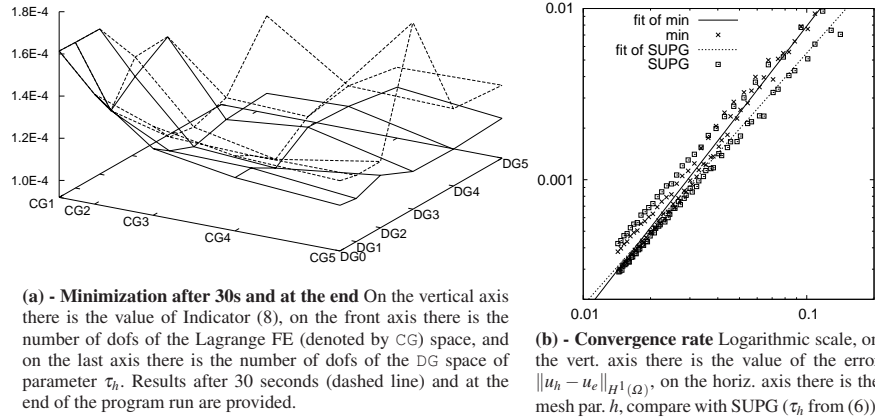


**(a) - Minimization after 30s and at the end** On the vertical axis there is the value of Indicator (8), on the front axis there is the number of dofs of the Lagrange FE (denoted by CG) space, and on the last axis there is the number of dofs of the DG space of parameter $\tau_h$. Results after 30 seconds (dashed line) and at the end of the program run are provided.

**(b) - Convergence rate** Logarithmic scale, on the vert. axis there is the value of the error $\|u_h - u_e\|_{H^1(\Omega)}$, on the horiz. axis there is the mesh par. $h$, compare with SUPG ($\tau_h$ from (6)).

**Fig. 4** Minimization after 30s and at the end **(a)** and convergence results **(b)** for Example 1

tion and run everytime the whole optimization process. We see that in this case the convergence rate of our method (labeled `min`) is approximately the same as of the SUPG method with parameter from (6) (labeled `SUPG`).

To obtain precisely the convergence rates we use a different technique than it is usual in other papers from the numerical mathematics branch. We use a technique of curve fitting which is rather known to physicists. This approach is justified by having a lot of data for such a "physical" fitting. The objective function $f(a,b)$, whose parameters $a$ and $b$ are fitted, has the form

$$f(a,b) = a \cdot h^b. \tag{14}$$

Python `scipy` library uses the Levenberg–Marquardt algorithm in `curve_fit` function of `scipy` module. Paper [9] describes how this algorithm is implemented in the python library. An important fact about this approach is that we obtain also a rigorous uncertainty or standard deviation of fitted coefficients.

In Figure 5a we can see the convergence results of SUPG method for Example 2 and in Figure 5b we can see the convergence results obtained by minimizing according to the Indicator (8). As we see, if the meshes are chosen properly we are able to obtain a higher order convergence using our technique. A properly chosen mesh in Example 2 is apparently, regarding the exact solution or regarding the convergence graphs, a structured mesh which has $2 + 4k$ sides (or equivalently $3 + 4k$ nodes), $k \in \mathbb{N}$ at $x = 0$. We will refer to such a mesh as `4k` mesh. The `4k` mesh has the inner layers well resolved by an element's side.

The resulting coefficients of fitted function for the SUPG method with $\tau_h$ defined in (6) are, together with their standard deviation: $f(h) = (0.5 \pm 0.3) \cdot h^{(2.5 \pm 0.1)}$. The values of the coefficient $a$ naturally differ slightly among the types of meshes with
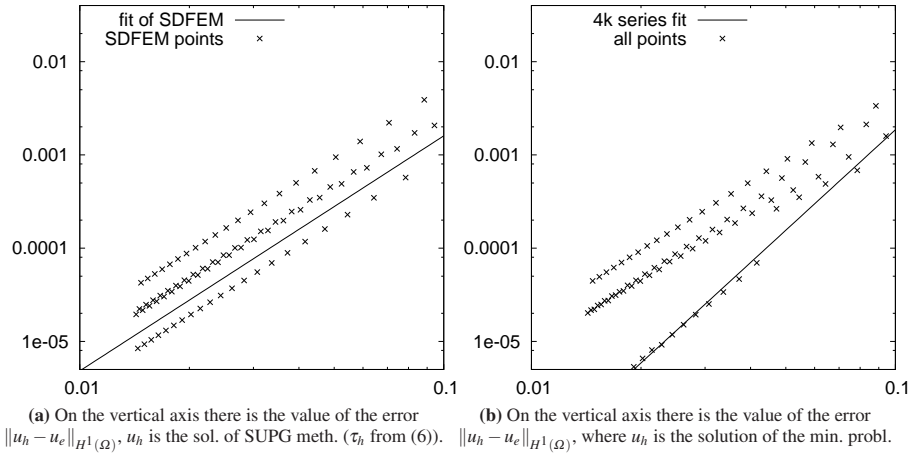


**(a)** On the vertical axis there is the value of the error $\|u_h - u_e\|_{H^1(\Omega)}$, $u_h$ is the sol. of SUPG meth. ($\tau_h$ from (6)).

**(b)** On the vertical axis there is the value of the error $\|u_h - u_e\|_{H^1(\Omega)}$, where $u_h$ is the solution of the min. probl.

**Fig. 5** Comparison of the results from the SUPG method with parameter $\tau_h$ from (6) and the adaptive method. On the horizontal axis there is the mesh parameter $h$. On the right hand side the fit is only for the points corresponding to the `4k` mesh. Results are for Example 2.

different numbers of elements, the alignment of elements' sides with the direction of the convection is better than in Example 1. The values of parameter $b$ were $(2.5 \pm 0.1)$. So we have $h^{2.5}$ convergence rate for the SUPG method for Example 2 on our structured grid.

But how behaves the solutions of our adaptive method? The Figure 5 suggests a higher rate of convergence than the SUPG method has for a special mesh (every fourth point in the graph has apparently a different order of convergence). After all, for the `4k` mesh defined earlier the fitted function is $f(h) = (7.0 \pm 0.3) \cdot h^{(3.5 \pm 0.1)}$. This means that the convergence rate is $h^{3.5}$. The gain in convergence rate in this setting is 1 in comparison with SUPG method. It is a substantial improvement in this setup and shows us the potential of our adaptive techniques.

## 7 Conclusions

From the numerical tests we have done so far it comes out that using higher order finite elements or higher order discontinuous finite element spaces for the parameter $\tau_h$ has almost no positive effect on the resulting discrete solution of our adaptive technique. By a higher order finite element is meant an element with the polynomial degree higher than 3. This holds also for other indicators we have tested in our adaptive framework.

Much more useful seems to be to use the adaptive method together with a **carefully chosen mesh**. In such a setup the adaptive method can give us satisfactory results since the rate of convergence of the SUPG method can be improved by the order of 1.

Although the nodally exact solution of the initial equation is piecewise flat and the interior layer follows a smooth curve, the oscillations in the SUPG solution still appear along the sharp layers of the solution of (1). The optimization method then chooses the parameter $\tau_h$ so that it minimizes the error indicator. It is natural that the parameter itself is not smooth anymore as it can change even inside one element quite rapidly. To get an insight into this behaviour is not easy as the value of the parameter $\tau_h$ is a product of the process of a nonlinear optimization. On the other hand we see in Figures 2 and 3 that higher values of optimized parameter $\tau_h$ appear where necessary which means in the vicinity of spurious oscillations of the solution of SUPG method.

Our future interest is to implement and test other error indicators which could be more suitable to our adaptive method. We will also apply our adaptive method to other stabilized methods with free parameters.

## References

1. Brooks, A.N., Hughes T.J.R.: Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations. Comput. Methods Appl. Mech. Engrg. **32**, 199–259 (1982).
2. Ciarlet, P.G.: The Finite Element Method for Elliptic Problems. North-Holland, NY (1978).
3. Giles, M.B., Pierce, N.A.: An introduction to the adjoint approach to design. Flow, Turbulence and Combustion **65**, 393–415 (2000).
4. John, V., Knobloch, P.: On 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**, 1997–2014 (2008).
5. John, V., Knobloch, P., Savescu, S.B.: A posteriori optimization of parameters in stabilized methods for convection–diffusion problems - Part I. Comput. Methods Appl. Mech. Engrg. **200**, 2916–2929 (2011).
6. Knopp, T., Lube, G., Rapin, G.: Stabilized finite element methods with shock capturing for advection-diffusion problems. Comput. Methods Appl. Mech. Engrg. **191**, 2997–3013 (2002).
7. Logg, A., Mardal, K.-A., Wells, G.N., et al.: Automated Solution of Differential Equations by the Finite Element Method. Springer (2012) doi:10.1007/978-3-642-23099-8, http://fenicsproject.org.
8. Lukáš, P.: Adaptive choice of parameters in stabilization methods for convection–diffusion equations. Master Thesis, Charles University, Prague, 2011.
9. Moré, J.J.: The Levenberg–Marquardt algorithm: implementation and theory. Numerical analysis **630**, 105–116 (1978).
10. Nocedal, J., Wright, S.J.: Numerical Optimization, Springer, NY (2006).