Comput. Methods Appl. Mech. Engrg. 273 (2014) 219–237

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

An adaptive SUPG method for evolutionary convection–diffusion equations

Javier de Frutos [a,1], Bosco García-Archilla [b,2], Volker John [c,d], Julia Novo [e,*,3]

[a] Instituto de Investigación en Matemáticas (IMUVA) y Departamento de Matemática Aplicada, Facultad de Ciencias, Universidad de Valladolid, Campus Miguel Delibes, Paseo Belén 7, 47011, Valladolid, Spain
[b] Departamento de Matemática Aplicada II, Universidad de Sevilla, Camino de los Descubrimientos s/n, 41092, Sevilla, Spain
[c] Weierstrass Institute for Applied Analysis and Stochastics, Leibniz Institute in Forschungsverbund Berlin e. V. (WIAS), Mohrenstr. 39, 10117 Berlin, Germany
[d] Free University of Berlin, Department of Mathematics and Computer Science, Arnimallee 6, 14195 Berlin, Germany
[e] Departamento de Matemáticas, Facultad de Ciencias, Universidad Autónoma de Madrid, Cantoblanco, 28049, Madrid, Spain

ARTICLE INFO

Article history:
Received 17 October 2013
Received in revised form 21 January 2014
Accepted 25 January 2014
Available online 6 February 2014

Keywords:
Evolutionary convection–diffusion–reaction equations
Streamline upwind Petrov–Galerkin (SUPG) method
Residual-based a posteriori error estimators
Effectivity index
Adaptive grid generation

ABSTRACT

An adaptive algorithm for the numerical simulation of time-dependent convection–diffusion–reaction equations will be proposed and studied. The algorithm allows the use of the natural extension of any error estimator for the steady-state problem for controlling local refinement and coarsening. The main idea consists in considering the SUPG solution of the evolutionary problem as the SUPG solution of a particular steady-state convection–diffusion problem with data depending on the computed solution. The application of the error estimator is based on a heuristic argument by considering a certain term to be of higher order. This argument is supported in the one-dimensional case by numerical analysis. In the numerical studies, particularly the residual-based error estimator from [18] will be applied, which has proved to be robust in the SUPG norm. The effectivity of this error estimator will be studied and the numerical results (accuracy of the solution, fineness of the meshes) will be compared with results obtained by utilizing the adaptive algorithm proposed in [5].

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

In this paper, the numerical approximation of evolutionary convection–reaction–diffusion equations with finite element methods is studied in the convection-dominated regime. It is well known that standard finite element approximations produce spurious oscillations and stabilized methods have to be utilized. One of the most popular stabilized finite element methods is the streamline-upwind Petrov–Galerkin (SUPG) method introduced in [3,14]. As stated in [25] there is a, perhaps general, consensus that adaptive methods will provide the most satisfactory approach for solving convection–diffusion problems. A posteriori error estimators or indicators are a necessary tool to perform adaptive algorithms. In [18], a residual-based a posteriori error estimator for the SUPG approximation to steady-state problems is proposed. The error estimator was

* Corresponding author. Tel.: +34 91 4974074; fax: +34 914974889.
E-mail addresses: frutos@mac.uva.es (J. de Frutos), bosco@esi.us.es (B. García-Archilla), volker.john@wias-berlin.de (V. John), julia.novo@uam.es (J. Novo).

[1] The research was supported by Spanish MICINN under grant MTM2010-14919.
[2] The research was supported by Spanish MICINN under grant MTM2012-31821.
[3] The research was supported by Spanish MICINN under grant MTM2010-14919.

http://dx.doi.org/10.1016/j.cma.2014.01.022
0045-7825/© 2014 Elsevier B.V. All rights reserved.

proved, under some hypotheses, to be robust in the SUPG norm, the natural norm in which the error of the method is usually measured. The aim of the present paper consists in deriving and studying an adaptive algorithm for the evolutionary convection–diffusion equation which uses the natural extension of the a posteriori estimator from [18] as criterion for the generation of adaptive meshes.

Adaptive methods and a posteriori error estimation for non-stationary convection–diffusion equations were already studied to some extent in the literature. In [28], residual-based a posteriori error estimates were proved. This paper considers discretizations with the θ-scheme in time and conforming finite elements in space covering the cases of using the standard Galerkin method and the SUPG method. The derived estimator is an extension of the estimator proposed in [27], where robust estimators were obtained in a norm that adds to the energy norm a dual norm of the convective derivative. Applying the estimator from [28] requires the solution of an auxiliary discrete stationary reaction–diffusion problem at each time step. The estimates are uniformly bounded with respect to the diffusion. An extension of the error estimator from [28] to the case of a stabilization by subgrid viscosity is considered in [1]. In [11], a finite element method in space and time with streamline diffusion is used to develop an adaptive method applied to a non-stationary nonlinear one-dimensional model. In [22], a space and time adaptive algorithm was proposed and an a posteriori upper bound was derived which is not robust with respect to the diffusion parameter. An adapted Lagrange–Galerkin method for singularly perturbed unsteady linear convection–diffusion problems was introduced in [12]. The Lagrange–Galerkin method is based on combining the method of characteristics with the standard Galerkin finite element method. An adaptive approach for a Lagrange–Galerkin method can be found in [2], see also [13]. In [8], a framework for a posteriori error estimates in unsteady, nonlinear, possibly degenerated convection–diffusion problems was introduced. The estimators are based on a space–time equilibrated flux reconstruction and were derived for the error measured in a space–time mesh-dependent dual norm.

Most of the above mentioned methods are based on space–time formulations. In [5], as well as in the present paper, we followed the approach of separating spatial and temporal discretizations using the method of lines. An adaptive algorithm based on the standard Galerkin method was introduced in [5, Section 3] to approximate linear evolutionary convection-dominated problems. The algorithm is based on an a posteriori indicator of the size of the oscillations displayed by the finite element approximation. This algorithm will be used for comparison with the adaptive algorithm introduced in the present paper for the SUPG method. The numerical studies in Section 4 will show that the newly proposed method performs generally better.

In this paper, an algorithm will be proposed and studied which allows to apply the natural extension of any a posteriori error estimator for the steady-state problem in the evolutionary setting. In the numerical studies, particularly the residual-based a posteriori error estimator from [18] will be used. The main idea consists in considering the SUPG approximation to the evolutionary problem also as the SUPG approximation to a special steady-state problem. For the latter problem, any of the available a posteriori error estimators for the steady-state case can be applied. The same idea has been used already for evolutionary nonlinear convection–diffusion problems in the diffusion-dominated regime in [4,6,7]. However, it should be noted that the approach presented here is still heuristic with respect to the error estimation. It relies upon the assumption that the error in the SUPG norm between the solution of the continuous evolutionary problem and the special steady-state problem is of higher order. We can justify this assumption by numerical analysis only in the one-dimensional case. But on the other hand, the numerical studies will show that the effectivity index of the used error estimator is robust with respect to the mesh width and the diffusion, see Example 2. This result supports the point of view that the essential information on the error is provided by the error estimator.

The outline of the paper is as follows. In Section 2, the SUPG discretization of the evolutionary convection–diffusion equation is given. The adaptive algorithm which uses the extension of the a posteriori error estimations from [18] is introduced in Section 3. In addition, the analytical results that justify the assumption made in deriving this algorithm are presented in this section. Numerical tests in Section 4 study the effectivity of the error estimator and compare the adaptive meshes computed with the proposed algorithm and the algorithm from [5]. The paper finishes with a summary in Section 5.

2. The equation and its discretization

Let $\Omega \subset \mathbb{R}^d,\ d \in \{2,3\}$, be a bounded domain with polyhedral Lipschitz boundary and let $[0,T]$ be a finite time interval. The time-dependent convection–diffusion–reaction equation which will be considered in this paper is given as follows

$$\begin{aligned}
&\partial_t u - \varepsilon\Delta u + \mathbf{b}\cdot\nabla u + cu = f \quad \text{in } (0,T]\times\Omega,\\
&u = 0 \quad \text{on } [0,T]\times\Gamma_D,\\
&\varepsilon\frac{\partial u}{\partial \mathbf{n}} = \varepsilon\partial_{\mathbf{n}} u = g \quad \text{on } [0,T]\times\Gamma_N,\\
&u(0,\mathbf{x}) = u_0(\mathbf{x}) \text{ in } \Omega,
\end{aligned} \tag{1}$$

where $\varepsilon > 0$, all given functions are sufficiently smooth, and $\mathbf{n}$ is the outward pointing unit normal at the boundary of Ω. It will be assumed that $\partial^-\Omega \subset \Gamma_D$, $\partial^-\Omega$ being the inflow boundary of Ω, i.e., the set of points $\mathbf{x} \in \partial\Omega$ such that $\mathbf{b}(\mathbf{x})\cdot\mathbf{n}(\mathbf{x}) < 0$. For simplicity, in the present section, we assume $g = 0$. A standard assumption in the analysis for equations of type (1) is that the following condition is satisfied

$$c(t,\mathbf{x}) - \frac{1}{2}\operatorname{div}(\mathbf{b}(t,\mathbf{x})) = \mu(t,\mathbf{x}) \geqslant \mu_0 > 0 \quad \forall\ (t,\mathbf{x}) \in [0,T]\times\Omega. \tag{2}$$

If this assumption is not satisfied, the change of variables $v = ue^{-\alpha t}$ with $\alpha > 0$ transforms problem (1) into one satisfying (2).

Finite element spaces are denoted by $V_{h,r}$, where h indicates the fineness of the underlying triangulation $\mathcal{T}_h$ consisting of mesh cells $\{K\}$, based on continuous local polynomials of degree r. Only conforming finite element spaces will be considered, which implies in particular that $v_h = 0$ on $\partial\Omega_D$ for all $v_h \in V_{h,r}$.

The semidiscrete in space Galerkin approximation based on linear polynomials to problem (1) consists in finding $u_h^G:\ (0,T] \to V_{h,1}$ satisfying $u_h(0,\cdot) = u_h^0 \in V_{h,1}$ and

$$(\partial_t u_h^G, v_h) + \varepsilon(\nabla u_h^G, \nabla v_h) + (\mathbf{b}\cdot\nabla u_h^G, v_h) + (cu_h^G, v_h) = (f, v_h), \tag{3}$$

for all $v_h \in V_{h,1}$. It is well known that the standard Galerkin approximation produces spurious oscillations in the convection-dominated regime, which is given if for the mesh Péclet number holds $Pe_K = \|\mathbf{b}\|_{\infty,K} h_K/(2\varepsilon) > 1$, where h_K is some measure for the size of the mesh cell K, e.g., its length in the direction of the convection [16]. Stabilized discretizations are usually applied to cope with this difficulty.

As already mentioned in the introduction, this paper studies the SUPG method. The bilinear form associated with this method is given by

$$a_{\mathrm{SUPG}}(v_h, w_h) = \varepsilon(\nabla v_h, \nabla w_h) + (\mathbf{b}\cdot\nabla v_h, w_h) + (cv_h, w_h) + (-\varepsilon\Delta v_h + \mathbf{b}\cdot\nabla v_h + cv_h, \mathbf{b}\cdot\nabla w_h)_h \quad \forall\ v_h, w_h \in V_{h,r},$$

where $(\cdot,\cdot)_h$ denotes the broken inner product

$$(f,g)_h = \sum_{K\in\mathcal{T}_h} \delta_K (f,g)_K,$$

$\delta_K > 0$ being the stabilization parameter and $(\cdot,\cdot)_K$ is the standard inner product in $L^2(K)$. In the convection-dominated case, the stabilization parameter δ_K is typically defined as $\delta_K = \delta_0 h_K/\|\mathbf{b}\|_{\infty,K}$ for some positive constant δ_0 [16].

Let $u_h:\ (0,T] \to V_{h,1}$ denote the spatial semidiscrete SUPG approximation to (1) satisfying $u_h(0,\cdot) = u_h^0 \in V_{h,1}$ and

$$(\partial_t u_h, v_h) + a_{\mathrm{SUPG}}(u_h, v_h) = (f, v_h) + (f - \partial_t u_h, \mathbf{b}\cdot\nabla v_h)_h, \quad \forall\ v_h \in V_{h,1}. \tag{4}$$

Standard error analysis for stabilized methods is performed in norms which possess a contribution from the stabilization. For the SUPG approximation to (1) this norm has the form

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

see for example [17].

3. The adaptive algorithm based on the SUPG discretization

This section explains the a posteriori error estimator which will be used for the SUPG discretization (4) of the evolutionary problem (1). The main goal is to show that the a posteriori error estimator for the steady-state case from [18] can be extended to the evolutionary case. This extension is in some way heuristic since we can only justify analytically the one-dimensional case. The underlying idea can also be applied to any other error estimator for the steady-state problem and it has been used before in evolutionary nonlinear convection–diffusion problems in the diffusion-dominated regime in [4,6]. At the end of the section, the adaptive algorithm will be presented.

3.1. Residual-based estimators for an auxiliary steady-state problem

Consider the following steady-state problem

$$\begin{aligned} -\varepsilon\Delta\tilde{u} + \mathbf{b}\cdot\nabla\tilde{u} + c\tilde{u} &= f - \partial_t u_h && \text{in } \Omega,\\ \tilde{u} &= 0 && \text{on } \Gamma_D,\\ \varepsilon\partial_{\mathbf{n}}\tilde{u} &= g && \text{on } \Gamma_N. \end{aligned} \tag{6}$$

The continuous piecewise linear SUPG approximation to problem (6) is: find $u_h \in V_{h,1}$ such that

$$a_{\mathrm{SUPG}}(u_h, v_h) = (f - \partial_t u_h, v_h) + (f - \partial_t u_h, \mathbf{b}\cdot\nabla v_h)_h \quad \forall\ v_h \in V_{h,1}.$$

It coincides with (4) such that u_h is also the SUPG approximation to the evolutionary problem (1). Our goal consists in deriving, under certain assumptions, an a posteriori estimate for the error of u_h, where the error should be bounded in the norm (5), which is used in the a priori error analysis.

The starting point for getting an upper bound for the error is the decomposition

$$\|u - u_h\|_{\mathrm{SUPG}} \leqslant \|u - \tilde{u}\|_{\mathrm{SUPG}} + \|\tilde{u} - u_h\|_{\mathrm{SUPG}}. \tag{7}$$

Now, the main idea consists in neglecting the first term in this decomposition and to consider just the second one. Justifications for this approach will be discussed below.

To bound the second term on the right-hand side of (7), the residual-based a posteriori error estimator will be used which was proposed in [18] for the steady-state problem. Since u_h is the SUPG approximation to problem (6) with right-hand side $g = f - \partial_t u_h$, applying the estimator from [18], the error $\|\tilde{u} - u_h\|_{\mathrm{SUPG},K}$ at every mesh cell can be bounded by the quantity

$$\eta_K = \left(\min\left\{ \frac{C}{\mu_0}, C\frac{h_K^2}{\varepsilon}, 24\delta_K \right\} \|R_K(u_h)\|_{0,K}^2 + 24\delta_K \|R_K(u_h)\|_{0,K}^2 + \sum_{E\in\partial K} \min\left\{ \frac{24}{\|\mathbf{b}\|_{\infty,E}}, C\frac{h_E}{\varepsilon}, \frac{C}{\varepsilon^{1/2}\mu_0^{1/2}} \right\} \|R_E(u_h)\|_{0,E}^2 \right)^{1/2}, \tag{8}$$

where R_K and R_E are the inner and edge residuals, respectively, of problem (6), which are defined as follows

$$R_K(u_h) := f - \partial_t u_h + \varepsilon\Delta u_h - \mathbf{b}\cdot\nabla u_h - cu_h|_K, \tag{9}$$

$$R_E(u_h) := \begin{cases} -\varepsilon[\![\partial_{\mathbf{n}_E} u_h]\!] & \text{if } E \in \mathcal{E}_{h,\Omega}, \\ g - \varepsilon\partial_{\mathbf{n}_E} u_h & \text{if } E \in \mathcal{E}_{h,N}, \\ 0 & \text{if } E \in \mathcal{E}_{h,D}. \end{cases}$$

Here, the set of all faces in the finite element partition is denoted by $\mathcal{E}_h$ and $\mathcal{E}_{h,\Omega}$, $\mathcal{E}_{h,N}$, and $\mathcal{E}_{h,D}$ refer to interior faces, faces on the Neumann boundary, and faces on the Dirichlet boundary, respectively. In addition, the jump of any piecewise continuous function v across E in an arbitrary but fixed direction $\mathbf{n}_E$ orthogonal to E is denoted by $[\![v]\!]$.

As explained in the introduction, residual-based a posteriori error estimators for the steady-state problem, which estimate the error in other norms, see the overview in [18], will be used in the numerical studies as well. With the traditional approach of deriving residual-based estimators, see [26], and taking care only on the dependency of the weights on the local mesh width, one gets

$$\|\nabla(u - u_h)\|_0^2 \leqslant C\left(\sum_{K\in\mathcal{T}_h} h_K^2 \|R_K(u_h)\|_{0,K}^2 + \sum_{E\in\mathcal{E}_h} h_E \|R_E(u_h)\|_{0,E}^2 \right) + h.o.t., \tag{10}$$

$$\|u - u_h\|_0^2 \leqslant C\left(\sum_{K\in\mathcal{T}_h} h_K^4 \|R_K(u_h)\|_{0,K}^2 + \sum_{E\in\mathcal{E}_h} h_E^3 \|R_E(u_h)\|_{0,E}^2 \right) + h.o.t. \tag{11}$$

The estimators (10) and (11) are not robust in the convection-dominated regime, i.e., the constants C depend on the Péclet number, cf. the numerical studies in [15]. They become robust if diffusion dominates. The higher order terms describe data approximation errors.

An estimator that is robust in the $L^2(\Omega)$ norm was proposed in [10]. It has the form

$$\|u - u_h\|_0^2 \leqslant 2\sum_{K\in\mathcal{T}_h} \left(\eta_{L^2,K}^2 \|R_K(u_h)\|_{0,K}^2 + \frac{1}{4}\sum_{E\subset\partial K} \frac{|E|}{|K|} \eta_{L^2,K}^2 \|R_E(u_h)\|_{0,E}^2 \right), \tag{12}$$

where $|E|$ and $|K|$ are the measures of E and K, respectively, and

$$\eta_{L^2,K} = \min\left\{ \frac{\tilde{h}_K}{\sqrt{2}\|\mathbf{b}\|_{L^\infty(K)}}, \frac{h_K^2}{3\sqrt{10}\varepsilon}, \frac{1}{\|c\|_{L^\infty(K)}} \right\}$$

for the Q_1 finite element (bilinear element over quadrilateral grids), see [10, Appendix C]. Here, $\tilde{h}_K$ is the cell diameter in the direction of the convection vector.

The aim of this paper is to use (8) for estimating the error in every mesh cell and to control an adaptive algorithm. This estimator is the only one that has been proved, under some hypotheses, to be robust in the SUPG norm for the steady-state problem, see [18]. In one of the numerical examples, the results will be compared with the results obtained by using the estimators (10)–(12).

3.2. Discussion of neglecting the first term on the right-hand side of (7)

Next, it will be justified in some sense why the term $\|u - \tilde{u}\|_{\mathrm{SUPG}}$ in (7) is negligible when compared with $\|\tilde{u} - u_h\|_{\mathrm{SUPG}}$, and, consequently, why we only consider the second term. Note that the only difference in the equation for $\tilde{u}$ compared with the equation for u is the replacement of $\partial_t u$ by $\partial_t u_h$ on the right-hand side.

In the diffusion-dominated regime, the first term in an analogous decomposition to (7) has been proved to have a higher rate of decay than the error being estimated, see for example [4,6]. More precisely, the first term of the decomposition is proved to be smaller than the second one by a factor $\mathcal{O}(h|\log(h)|)$. The higher rate of decay is obtained in both the $L^2(\Omega)$ and $H^1(\Omega)$ norm, except for the linear finite element case in which the improved convergence is achieved only in the $H^1(\Omega)$ norm. This fact has been used to a posteriori estimate the errors in [4,6].

Let us observe that the function $\tilde{u}$ solving (6) belongs to an infinite dimensional space. In order to get some bounds for the error $u - \tilde{u}$, one needs to approximate $\tilde{u}$ by a function which is contained in a finite dimensional space. Since we are interested in the convection-dominated regime, the first simplification we apply to study the size of $\|u - \tilde{u}\|_{\text{SUPG}}$ consists in replacing $\tilde{u}$ by an approximation to the steady-state problem (6) based on the SUPG method. Since the error to the piecewise linear approximation u_h should be estimated, a straightforward idea is to consider a higher order SUPG approximation $\tilde{u}_h$ to (6). For this purpose, a second order approximation is used.

In the following we will assume that $\tilde{u}_h$ is a sufficiently good approximation to $\tilde{u}$ in the sense that

$$\|u - \tilde{u}\|_{\text{SUPG}} \approx \|u - \tilde{u}_h\|_{\text{SUPG}}.$$

By the triangle inequality it is clear that this assumption is satisfied if $\|\tilde{u} - \tilde{u}_h\|_{\text{SUPG}}$ is small enough. For this reason, we choose $\tilde{u}_h$ as the SUPG approximation based on quadratic polynomials to the steady-state problem (6) with solution $\tilde{u}$. Then $\|\tilde{u} - \tilde{u}_h\|_{\text{SUPG}}$ converges with order 5/2, which is larger than the rate of convergence of the terms being compared with.

The goal of this section is to prove that, in one space dimension and under certain regularity assumptions on the solution u, $\|u - \tilde{u}_h\|_{\text{SUPG}}$ is in the convection-dominated regime of higher order than $\|u - u_h\|_{\text{SUPG}}$. More precisely, while the order of the error $\|u - u_h\|_{\text{SUPG}}$ is 3/2, it will be shown that $\|u - \tilde{u}_h\|_{\text{SUPG}}$ is of order 2.

The model problem, which will be considered, has the form

$$\begin{aligned} &\partial_t u - \varepsilon \partial_{xx} u + \partial_x u + u = f \quad \text{in } (0,T] \times (0,1), \\ &u(0,t) = u(1,t) = 0 \quad \text{on } [0,T], \\ &u(0,x) = u_0(x) \quad \text{in } (0,1). \end{aligned} \tag{13}$$

For simplicity, a uniform partition $\{0 = x_0 < x_1 < \cdots < x_N = 1\}$ of $[0,1]$ of size $h = 1/N$ is used. The same value for the stability parameter at every subinterval K of the partition is applied: $\delta_K = \delta = h/2$.

The SUPG approximation to the corresponding steady-state problem to (6) based on quadratic polynomials consists in finding $\tilde{u}_h \in V_{h,2}$ such that

$$(\varepsilon + \delta)(\partial_x \tilde{u}_h, \partial_x v_h) + (\partial_x \tilde{u}_h, v_h) + (\tilde{u}_h, v_h) + \delta(\tilde{u}_h, \partial_x v_h) = (f - \partial_t u_h, v_h) + \delta(f - \partial_t u_h, \partial_x v_h) + \sum_{K \in \mathcal{T}_h} \delta\varepsilon(\partial_{xx} \tilde{u}_h, \partial_x v_h)_K, \tag{14}$$

for all $v_h \in V_{h,2}$.

For the analysis of this section, in addition the following auxiliary steady-state problem is considered, where the right-hand side depends on $\partial_t u$,

$$-\varepsilon \partial_{xx} w + \partial_x w + w = g, \quad 0 < x < 1, \quad w(0) = w(1) = 0, \tag{15}$$

with $g = f - \partial_t u$. It is clear that at any fixed time t the solution $u(t,\cdot)$ of (13) is also the solution of (15). Denote by w_h the SUPG approximation to (15) based on quadratic polynomials which satisfies

$$(\varepsilon + \delta)(\partial_x w_h, \partial_x \varphi_h) + (\partial_x w_h, \varphi_h) + (w_h, \varphi_h) + \delta(w_h, \partial_x \varphi_h) = (g, \varphi_h) + \delta(g, \partial_x \varphi_h) + \sum_{K \in \mathcal{T}_h} \delta\varepsilon(\partial_{xx} w_h, \partial_x \varphi_h)_K \quad \forall\ \varphi_h \in V_{h,2} \tag{16}$$

In the following analysis, the splitting $u - \tilde{u}_h = (u - w_h) + (w_h - \tilde{u}_h)$ will be applied. The error bounds for w_h are the standard bounds for the SUPG approximation in the steady-state case, e.g., see [25, Theorem 10.5] or [20]: if $u \in H^3(\Omega)$ then

$$\|u - w_h\|_{\text{SUPG}} \leqslant C h^{5/2} \|u\|_3. \tag{17}$$

Two lemmas will be proved to obtain the main result of the section. First, in Lemma 1, the auxiliary steady-state problem

$$-\varepsilon \partial_{xx} w + \partial_x w = g, \quad \text{in } (0,1), \quad w(0) = w(1) = 0, \tag{18}$$

will be studied. In Lemma 2 the evolutionary problem

$$\begin{aligned} &\partial_t u - \varepsilon \partial_{xx} u + \partial_x u = f \quad \text{in } (0,T] \times (0,1), \\ &u(0,t) = u(1,t) = 0 \quad \text{on } [0,T], \\ &u(0,x) = u_0(x) \quad \text{in } (0,1), \end{aligned} \tag{19}$$

will be considered.

It will be shown in Lemma 1 that, if the solution is sufficiently smooth, the $L^2(\Omega)$ norm of the error of the SUPG approximation to (18), based on linear elements, behaves at least as $\mathcal{O}(h^2)$ instead of $\mathcal{O}(h^{3/2})$, which is the bound for the general case, see [25, Theorem 10.5]. In Lemma 2, these results are extended to the evolutionary problem (19). In Remark 2 it will be discussed that the results of Lemma 2 are also valid for the original model (13). Finally, in Theorem 1 it will be proved that, under certain regularity assumptions for the solution u of (13), the error $\|u - \tilde{u}_h\|_{\text{SUPG}}$ is $\mathcal{O}(h^2)$ in the convection-dominated regime.

Lemma 1. *Let w be the solution of (18) and assume that $w \in C^2([0,1])$. Consider a mesh with $\varepsilon \leqslant h/6$ and denote by w_h the SUPG approximation based on linear finite elements. Then, the following bound holds*

$$\|w - w_h\|_0 \leqslant Ch^2,$$

where the constant C depends on $K_0 = \max_{[0,1]}|\partial_{xx} w|$.

Proof. Denote by $\pi_h w \in V_{h,1}$ the elliptic projection of w into $V_{h,1}$ given by

$$(\partial_x \pi_h w, \partial_x v_h) = (\partial_x w, \partial_x v_h) \quad \forall\ v_h \in V_{h,1}.$$

Starting with the weak form of (18) for test functions from $V_{h,1}$ and using the definition of $\pi_h w$, straightforward calculations show that

$$(\varepsilon + \delta)(\partial_x \pi_h w, \partial_x v_h) + (\partial_x \pi_h w, v_h) = (g, v_h) + \delta(g, \partial_x v_h) + (\partial_x(\pi_h w - w), v_h) + \delta(\varepsilon \partial_{xx} w, \partial_x v_h).$$

Consequently, one obtains for $e_h = \pi_h w - w_h$ the equation

$$(\varepsilon + \delta)(\partial_x e_h, \partial_x v_h) + (\partial_x e_h, v_h) = (\partial_x(\pi_h w - w), v_h) + \delta\varepsilon(\partial_{xx} w, \partial_x v_h).$$

Denoting by e_j the coefficients of e_h in the nodal basis, $e_h = \sum_{j=1}^{N-1} e_j \varphi_j$, taking into account that adding the stabilization term $\delta(\partial_x e_h, \partial_x v_h)$ is the same as doing upwind in the convective term (instead of approximating it by central finite differences) since $\delta = h/2$ (or performing a straightforward calculation), one gets

$$e_j - e_{j-1} = (\partial_x(\pi_h w - w), \varphi_j) + \frac{h\varepsilon}{2}(\partial_{xx} w, \partial_x \varphi_j) - \varepsilon(\partial_x e_h, \partial_x \varphi_j), \quad j = 1, \ldots, N-1.$$

Taking the sum from 1 to j and using $e_0 = 0$, one obtains

$$e_j = F_1 + F_2, \tag{20}$$

where

$$F_1 = (\partial_x(\pi_h w - w), \varphi_1 + \varphi_2 + \cdots + \varphi_j),$$

$$F_2 = \delta\varepsilon(\partial_{xx} w, \partial_x(\varphi_1 + \varphi_2 + \cdots + \varphi_j)) - \varepsilon(\partial_x e_h, \partial_x(\varphi_1 + \varphi_2 + \cdots + \varphi_j)).$$

Integrating by parts and taking into account that $\partial_x(\varphi_1 + \cdots + \varphi_j)$ vanishes in $[x_1, x_j]$ yields

$$|F_1| \leqslant \frac{1}{h}\left| -\int_0^{x_1} (\pi_h w - w)dx + \int_{x_j}^{x_{j+1}} (\pi_h w - w)dx \right| \leqslant \frac{1}{h}\left(h\|\pi_h w - w\|_\infty + h\|\pi_h w - w\|_\infty\right) \leqslant CK_0 h^2. \tag{21}$$

In the last inequality, the bound

$$\|\pi_h w - w\|_\infty \leqslant Ch^2 \max_{[0,1]} |\partial_{xx} w|,$$

was used, see [24]. For the second term, one obtains

$$|F_2| \leqslant \frac{\varepsilon}{2}\left| \int_0^{x_1} \partial_{xx} w dx - \int_{x_j}^{x_{j+1}} \partial_{xx} w dx \right| + \frac{\varepsilon}{h}\left| -\int_0^{x_1} \partial_x e_h dx + \int_{x_j}^{x_{j+1}} \partial_x e_h dx \right| \leqslant \varepsilon h K + \frac{\varepsilon}{h}|-e_1 + e_{j+1} - e_j|. \tag{22}$$

From (20)–(22), it follows that

$$|e_j| \leqslant Ch^2 + 3\frac{\varepsilon}{h}\|e_h\|_\infty, \quad j = 1, \ldots, N-1,$$

and consequently

$$\|e_h\|_\infty \leqslant Ch^2 + 3\frac{\varepsilon}{h}\|e_h\|_\infty,$$

or

$$\left(1 - 3\frac{\varepsilon}{h}\right)\|e_h\|_\infty \leqslant Ch^2.$$

Using the assumption on the mesh width, one gets $\|e_h\|_\infty \leqslant 2Ch^2$, which implies $\|e_h\|_0 \leqslant 2Ch^2$.

Finally, decomposing $(w - w_h) = (w - \pi_h w) + e_h$ and using $\|w - \pi_h w\|_0 = \mathcal{O}(h^2)$ finishes the proof by applying the triangle inequality. □

Remark 1. Note that the order of convergence of the SUPG method in the $L^2(\Omega)$ norm for linear finite elements has been proved to be $\mathcal{O}(h^{3/2})$ in the general multi-dimensional case, see for example [23]. As stated in [25], the apparent loss of half an order in the $L^2(\Omega)$ norm has attracted much attention. In [30], the optimal accuracy of the SUPG method for linear elements was investigated. The use of a very special type of meshes allows to prove that the SUPG method can converge in the $L^2(\Omega)$ norm with any order between 3/2 and 2. In [29], by orienting the mesh in the streamline direction and imposing

a uniformity condition on the mesh, the theoretical order of pointwise convergence is proved to be $\mathcal{O}(h^2|\log(h)|)$. The computational results of [29] show that the SUPG method actually converges in the maximum norm with the optimal order $\mathcal{O}(h^2)$ on many standard quasi-uniform meshes. The question of improved $L^2(\Omega)$ error estimates has also been considered in [21]. Based on these results from the literature, one can conclude that in the particular one-dimensional case the order of convergence obtained in Lemma 1 is the expected one. Also, one can expect that in many numerical simulations, a quadratic order of convergence for the SUPG method in the $L^2(\Omega)$ norm will be observed. In this sense, although the results shown here are only valid in the one-dimensional case, they give some insight in the validity of analogous results in more general situations.

The next lemma shows that the bound of Lemma 1 can be extended to the evolutionary case, improving to $\mathcal{O}(h^2)$ the bound for the $L^2(\Omega)$ norm of the error for the SUPG approximation u_h to (19). Under weaker regularity assumptions for the solution u, the bound for $\|u-u_h\|_0$ obtained in [17] is $\mathcal{O}(h^{3/2})$.

Lemma 2. *Let u be the solution of (19) and assume that $u, \partial_t u \in C^2([0,1])$ for all $t \in [0,T]$. Let u_h be its SUPG approximation based on linear finite elements and denote by $\Pi_h u \in V_{h,1}$ the SUPG approximation to the steady-state problem (18) with right-hand side $g = f - \partial_t u$. Assume that $u_h(0) = \Pi_h u_0$. Then, the following bound holds*

$$\|(u-u_h)(t)\|_0 \leqslant Ch^2, \quad t \in [0,T], \tag{23}$$

where the constant C depends on $\max_{0\leqslant t\leqslant T}(K_1(t), K_2(t))$ with $K_1(t) = \max_{[0,1]}|\partial_{xx}u(t)|$ and $K_2(t) = \max_{[0,1]}|\partial_{txx}u(t)|$.

Proof. Following [17], one compares u_h with $\Pi_h u \in V_{h,1}$, where $\Pi_h u$ is defined by

$$a_{\mathrm{SUPG}}(\Pi_h u, v_h) = a_{\mathrm{SUPG}}(u, v_h) = (f(t) - \partial_t u, v_h) + \delta(f - \partial_t u, \partial_x v_h)$$

for all $v_h \in V_{h,1}$. Applying Lemma 1, one obtains the following bound

$$\|u(t) - \Pi_h u(t)\|_0 \leqslant C(K_1(t))h^2, \quad t \in [0,T]. \tag{24}$$

Moreover, considering (18) with the right-hand side $g = \partial_t f - \partial_{tt} u$ and using $(\Pi_h u(t))_t = \Pi_h(\partial_t u(t))$, one can apply Lemma 1 again and gets

$$\|\partial_t u(t) - \Pi_h \partial_t u(t)\|_0 \leqslant C(K_2(t))h^2, \quad t \in [0,T]. \tag{25}$$

The error is decomposed in the following way

$$(u-u_h)(t) = (u - \Pi_h u)(t) + (\Pi_h u - u_h)(t).$$

To bound the first term on the right-hand side, one can apply (24). For the second term, following the notation in [17], denote $e_h = u_h - \Pi_h u$ and observe that the error satisfies

$$(\partial_t e_h, v_h) + a_{\mathrm{SUPG}}(e_h, v_h) = (T_{\mathrm{tr}}, v_h) + \delta(T_{\mathrm{tr}}, \partial_x v_h) - \delta(\partial_t e_h, \partial_x v_h)$$

for all $v_h \in V_{h,1}$, where $T_{\mathrm{tr}} = \partial_t u - \Pi_h \partial_t u$. Arguing exactly as in the proof of [17, Lemma 5.1] yields

$$\|e_h(t)\|_0^2 \leqslant C\left(\|e_h(0)\|_0^2 + \int_0^t \|T_{\mathrm{tr}}(s)\|_0^2 ds\right).$$

Using the assumption $e_h(0) = 0$ and applying (25) to bound $\|T_{\mathrm{tr}}\|_0$ finishes the proof. □

Remark 2. Observe that the change of variable $v = e^t u$ transforms problem (13) into (19). Hence, the error bounds of Lemma 2 for the SUPG approximation to problem (19) also hold for the SUPG approximation to problem (13). In the next theorem, the original problem (13) is considered for proving the main result of the section.

Theorem 1. *Let $\tilde{u}_h$ be the SUPG approximation defined in (14), let u be the solution of (13) satisfying $u \in H^3(0,1)$ and $u,\ \partial_t u,\ \partial_{tt}^2 u \in C^2([0,1])$ for all $t \in [0,T]$, and assume $u_h(0) = \Pi_h u_0$. Then, in the convection-dominated regime $\varepsilon \ll h$, the following bound holds*

$$\|(u-\tilde{u}_h)(t)\|_{\mathrm{SUPG}} \leqslant Ch^2, \quad t \in [0,T],$$

where the constant C depends on $\|u(t)\|_3$ and on $\max_{0\leqslant t\leqslant T}(K_2(t), K_3(t))$, $K_2(t) = \max_{[0,1]}|(\partial_{txx}u(t))|$, and $K_3(t) = \max_{[0,1]}|(\partial_{ttxx}u(t))|$.

Proof. The proof starts with the decomposition

$$(u-\tilde{u}_h) = (u - w_h) + (w_h - \tilde{u}_h),$$

where w_h is the SUPG approximation based on quadratic polynomials (16) to the steady-state problem (15) with right-hand side $f - \partial_t u$.

The bound (17) can be applied for the first term. To bound the second one, denote $\tilde{e}_h = \tilde{u}_h - w_h$. Subtracting (16) from (14) gives for all $v_h \in V_{h,2}$

$$(\varepsilon+\delta)(\partial_x\tilde{e}_h,\partial_x v_h)+(\partial_x\tilde{e}_h,v_h)+(\tilde{e}_h,v_h)+\delta(\tilde{e}_h,\partial_x v_h)-\sum_{K\in\mathcal{T}_h}\delta\varepsilon(\partial_{xx}\tilde{e}_h,\partial_x v_h)_K=(\partial_t u-\partial_t u_h,v_h)+\delta(\partial_t u-\partial_t u_h,\partial_x v_h).$$

Now, one takes $v_h = \tilde{e}_h$. Then, one gets

$$\sum_{K\in\mathcal{T}_h}\delta\varepsilon(\partial_{xx}\tilde{e}_h,\partial_x e_h)_K\leqslant\frac{h\varepsilon}{2}\sum_{K\in\mathcal{T}_h}\|\partial_{xx}\tilde{e}_h\|_{0,K}\|\partial_x e_h\|_{0,K}\leqslant\frac{C_{\text{inv}}\varepsilon}{2}\|\partial_x e_h\|_0^2,$$

where C_{inv} is the constant in the inverse estimate. A straightforward calculation gives $C_{\text{inv}} = 1/\sqrt{12}$. Applying integration by parts and the Cauchy–Schwarz inequality leads to

$$\left(\frac{\varepsilon}{2}+\delta\right)\|\tilde{e}_h\|_1^2+\|\tilde{e}_h\|_0^2\leqslant\frac{1}{2}\|\partial_t u-\partial_t u_h\|_0^2+\frac{1}{2}\|\tilde{e}_h\|_0^2+\frac{\delta}{2}\|\partial_t u-\partial_t u_h\|_0^2+\frac{\delta}{2}\|\tilde{e}_h\|_1^2,$$

from which it follows that

$$\|\tilde{e}_h\|_{\text{SUPG}}^2=(\varepsilon+\delta)\|\tilde{e}_h\|_1^2+\|\tilde{e}_h\|_0^2\leqslant\|\partial_t u-\partial_t u_h\|_0^2+\delta\|\partial_t u-\partial_t u_h\|_0^2. \tag{26}$$

Now, since $\partial_t u_h$ is the SUPG approximation to the evolutionary problem with solution $\partial_t u$, from Lemma 2, see also Remark 2, arguing as in [17], one gets

$$\|\partial_t u-\partial_t u_h\|_0\leqslant Ch^2, \tag{27}$$

where C depends on $\max_{0\leqslant t\leqslant T}(K_2(t),K_3(t))$, $K_2(t)=\max_{[0,1]}|\partial_{txx}u(t)|$, and $K_3(t)=\max_{[0,1]}|\partial_{ttxx}u(t)|$. Finally, the error at the initial time is bounded by

$$\|\partial_t u(0)-\partial_t u_h(0)\|_0\leqslant\|\partial_t e_h(0)\|_0+\|\Pi_h\partial_t u(0)-\partial_t u(0)\|_0,$$

where, as in Lemma 2, $e_h = u_h - \Pi_h u$. To bound the second term, (25) is applied. For estimating the first term, one can follow [17, Lemma 5.1] and use that $\partial_x e_h(0) = 0$ since $e_h(0) = 0$.

Finally, taking into account $\varepsilon \ll h$, one obtains from (26) and (27) the bound

$$\|\tilde{e}_h\|_{\text{SUPG}}\leqslant Ch^2,$$

which finishes the proof. □

Example 1. Consider problem (1) in $\Omega = (0,1)\times(0,1)$ with homogeneous boundary conditions on $\partial\Omega$, $\mathbf{b} = (-1,1)^T$, $c = 1$, and f and u_0 chosen so that the analytical solution is the same as in [18, Example 6.1]

$$u(x,y,t)=\exp(\sin(2\pi t))\sin(2\pi x)\sin(2\pi y).$$

We consider a uniform partition of size h and compare the errors in the convective derivative of the following approximations:

- u_h, the SUPG approximation based on linear elements to problem (1),
- w_h, the SUPG approximation based on quadratic polynomials to the following steady-state problem

$$-\varepsilon\Delta w+\mathbf{b}\cdot\nabla w+w=f-\partial_t u,\quad\text{in }\Omega,\quad w=0\quad\text{on }\partial\Omega,$$

- $\tilde{u}_h$, the SUPG approximation based on quadratic polynomials to problem (6).

The trapezoidal rule as described in Section 3.3 was used as temporal discretization.

From the general convergence theory for the SUPG method, one expects that $\|\mathbf{b}\cdot\nabla(u-w_h)\|_0=\mathcal{O}(h^2)$ and $\|\mathbf{b}\cdot\nabla(u-u_h)\|_0=\mathcal{O}(h)$. Assuming that the results from Theorem 1 carry over to this two-dimensional example with a smooth solution on a uniform grid, one expects that $\|\mathbf{b}\cdot\nabla(u-\tilde{u}_h)\|_0=\mathcal{O}(h^{3/2})$.

In the left picture of Fig. 1, the three considered errors are depicted for three decreasing values of $h = 1/N$, against t. The errors with respect to w_h and u_h behave as expected. For the error with respect to $\tilde{u}_h$, one can see even second order convergence. In the right picture of Fig. 1, where $\|\mathbf{b}\cdot\nabla(u-u_h)(t)\|_0/\|\mathbf{b}\cdot\nabla(u-\tilde{u}_h)(t)\|_0$ is presented, one can see this ratio increases roughly by a factor of two with one mesh refinement, showing that the order of convergence of the error with respect to $\tilde{u}_h$ is higher by one compared with the order with respect to u_h.

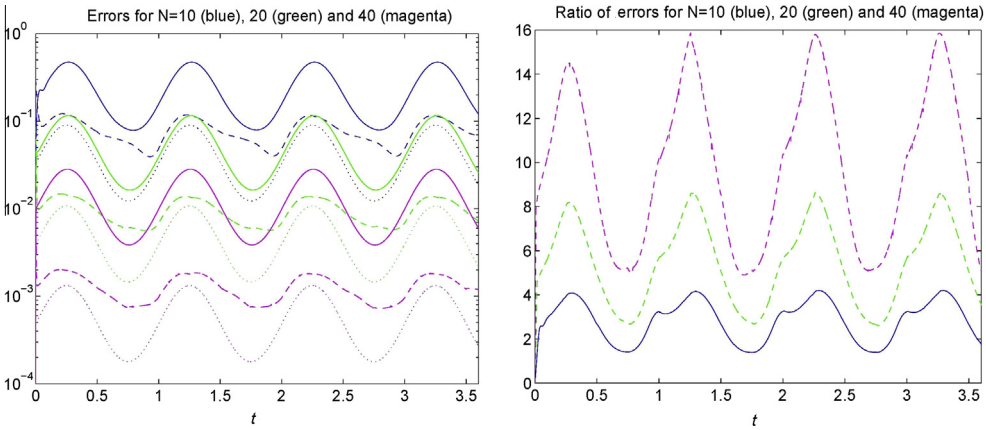

Fig. 1. Left: $L^2(\Omega)$ norm of the convective derivative of the error for u_h (continuous line), $\bar{u}_h$ (discontinuous line), and w_h (dotted line). Right: ratios between the $L^2(\Omega)$ norm of the convective derivative of the errors of u_h and $\bar{u}_h$.

3.3. The adaptive algorithm

As in [5], the trapezoidal rule implemented as described in [9] was used as temporal discretization. This time integrator is a variable step size implementation of the trapezoidal rule that uses the explicit two-step Adams formula for error estimation and employs a stabilization strategy to suppress the $(-1)^n$ mode, see [9] for details. The implementation in [9] is carefully designed to avoid subtractive cancellation. As argued in [9], this method is well suited for time-dependent convection–diffusion problems. The algorithm requires to choose several parameters, whose values were set in our computations exactly the same as used in the numerical simulations in [9], including the initial step size $\Delta t_0 = 10^{-10}$. The only change we made to the algorithm in [9] was that the step size updates suggested in [9, (1.15)] and those after a rejection were further multiplied by 0.85. This provision reduces the number of rejections. We tried different values for the tolerance of the time integrator. As in [5], the value of the tolerance in all the simulations presented in Section 4 was $\mathrm{TOLT} = 10^{-5}$. With this value, the temporal errors arising from the time discretization were smaller than the spatial errors.

Let U_h^n be the fully discrete approximation to Eq. (4) at time t_n. Then, the quantities

$$\eta_K^n = \left(\min\left\{ \frac{1}{\mu_0}, \frac{h_K^2}{\varepsilon}, 24\delta_K \right\} \|R_K(U_h^n)\|_{0,K}^2 + 24\delta_K \|R_K(U_h^n)\|_{0,K}^2 + \sum_{E\in\partial K} \min\left\{ \frac{24}{\|\mathbf{b}\|_{\infty,E}}, \frac{h_E}{\varepsilon}, \frac{1}{\varepsilon^{1/2}\mu_0^{1/2}} \right\} \|R_E(U_h^n)\|_{0,E}^2 \right)^{1/2}, \tag{28}$$

obtained from (8) replacing u_h by U_h^n, and setting $C = 1$ were computed. The residuals are defined as in (9), changing the semidiscrete in space approximations by the fully discrete approximations.

Assume that at a time level t_n an approximation U_h^n, defined in some finite element mesh $\mathcal{T}_h^n$, has been computed. This grid is the last one of a sequence of meshes, $\mathcal{T}_h^0, \mathcal{T}_h^1, \ldots, \mathcal{T}_h^n$, where each one is obtained from the previous one by subdividing or eliminating some mesh cells. Then, the adaptive algorithm reads as follows:

(1) Perform one time step to compute the SUPG approximation U_h^{n+1} at time t_{n+1} on the mesh $\mathcal{T}_h^n$.
(2) Compute the estimator η_K^{n+1} defined in (28).
(3) If $\eta_K^{n+1} > \mathrm{TOL}_1$ the cell K is marked to be refined. If $\eta_K^{n+1} < \mathrm{TOL}_2, K$ is marked as a candidate to be coarsened.
(4) Perform the refining/coarsening procedure to generate the new mesh $\mathcal{T}_h^{n+1}$.
(5) Interpolate U_h^{n+1} to the new mesh $\mathcal{T}_h^{n+1}$ and return to the first step of the algorithm until the final time is reached.

For the fourth point, the same procedure was used as described in detail in [5, Appendix].

4. Numerical studies

In this section, we consider the model problem (1) in the domain $\Omega = (0,1) \times (0,1)$. The temporal discretization is described in Section 3.3. In Example 2, the effectivity index of the global error estimator derived from (28) is studied. Examples 3–5 consider the proposed adaptive algorithm and compare it with the algorithm based on the Galerkin discretization from [5].

Example 2. *The effectivity index of the global error estimator.* This example studies the effectivity index, i.e., the ratio of the estimated error with the global form $\left(\sum_{K\in\mathcal{T}_h}(\eta_K^n)^2\right)^{1/2}$ of the a posteriori error estimator (28), and the actual error in the SUPG norm (5) at time t_n. A study of this kind requires the knowledge of (a good approximation of) the solution of the continuous problem (1). An example with analytically known solution with an interior layer was proposed in [19, Section 7.2], which was used also here. The solution

$$u(t;x,y) = 16\sin(\pi t)x(1-x)y(1-y)\times\left(\frac{1}{2}+\frac{\arctan[2\varepsilon^{-1/2}(0.25^2-(x-0.5)^2-(y-0.5)^2)]}{\pi}\right). \tag{29}$$

describes a hump that changes its height. In this example, the convection–diffusion problem (1) was solved with two values $\varepsilon\in\{10^{-3},10^{-6}\}$, $\mathbf{b}=(2,3)^T$, $c=1$, and the right-hand side and the boundary conditions were chosen such that (29) solves the problem. The trapezoidal rule was used in this example with an equi-distant time step $\Delta t = 10^{-3}$.

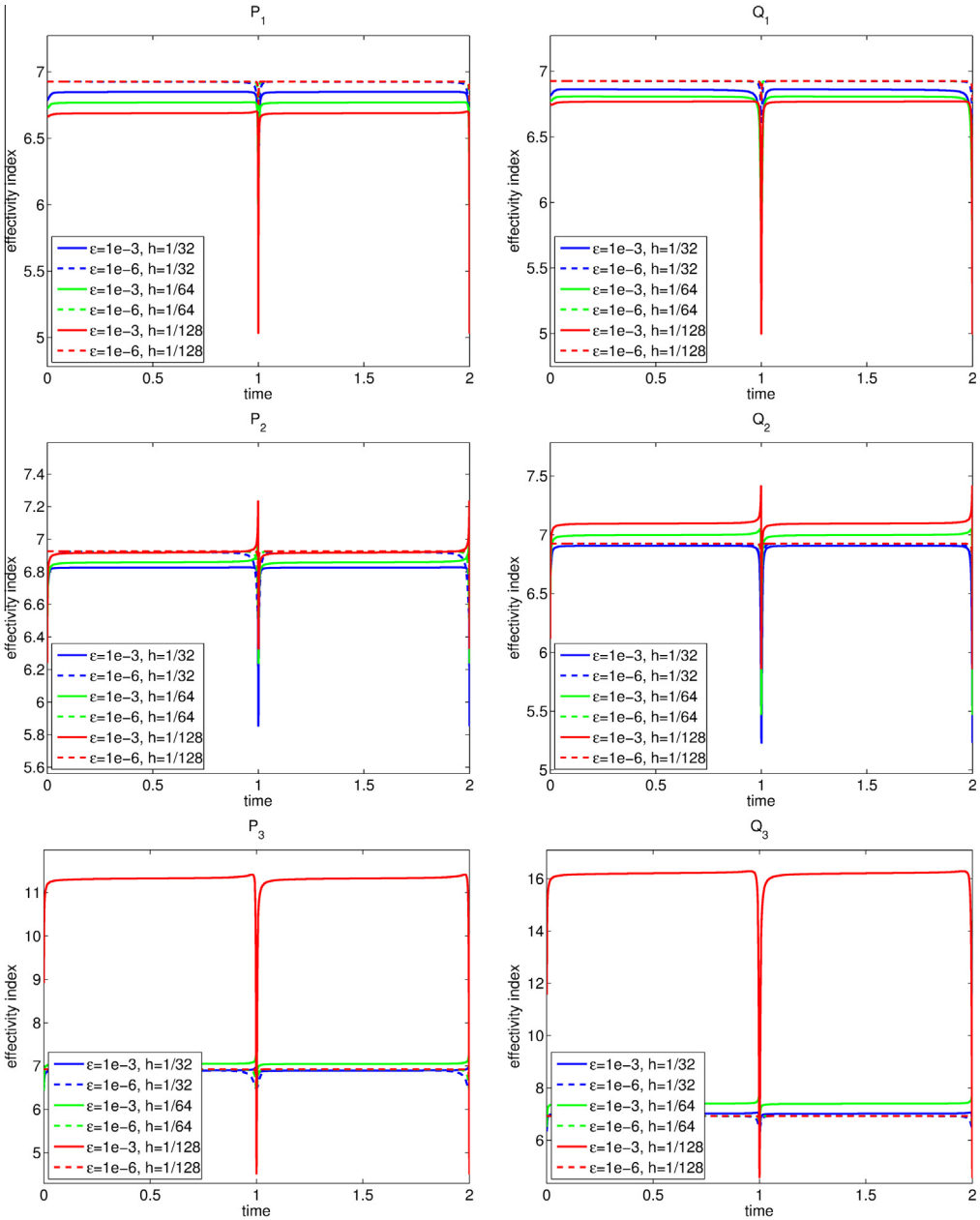

Fig. 2. Example 2. Effectivity indices.

Besides varying the diffusion, simulations were performed on uniform meshes of different mesh width and for triangular and quadrilateral finite elements of different order. The results for the effectivity index, as a function of time, are presented in Fig. 2, where P_i and Q_i $i = 1, 2, 3$, denote continuous finite element spaces of degree i on triangular and quadrilateral meshes, respectively. It can be seen that this index takes generally values of around 7, independent of the mesh size and the value of the diffusion as long as convection dominates. A similar observation was reported for several examples for steady-state problems in [18]. There are some peaks in the curves for the indices at the times $t = 1$ and $t = 2$ where the hump (29) changes its sign and both the continuous solution and its finite element approximation are very small. On the finest grid for the third order finite elements and $\varepsilon = 10^{-3}$, the effectivity index becomes larger. This increase of the index in the diffusion-dominated regime was also observed for steady-state problems in [18].

In summary, for the considered example, whose solution has an interior layer, the a posteriori error estimator (28) turned out to be robust with respect to the mesh width and the value of the diffusion.

Next, the same examples as in [5] will be studied to compare the adaptive algorithm for the SUPG method from Section 3.3 with the adaptive algorithm for the Galerkin method proposed in [5, Section 3]. For simplicity of notation, we will call them SUPG-based algorithm and Galerkin-based algorithm, respectively. Linear finite elements were used, as in [5]. The starting meshes consisted of a uniform 25×25 triangulation. Step (3) of the SUPG-based algorithm appears also in the Galerkin-based algorithm. However, the local error estimator (28) and the local indicator used in the Galerkin-based method are not related such that one cannot expect appropriate values for TOL_1 and TOL_2 to be the same. In [5], different combinations of values for TOL_1 and TOL_2 were investigated. The values $\mathrm{TOL}_1 = 10^{-2}$ and $\mathrm{TOL}_2 = \mathrm{TOL}_1/20$ were found to be best in terms of efficiency (number of degrees of freedom required for a prescribed accuracy) and consequently, results obtained with these values were presented in all pictures shown in [5]. Linear systems of equations were solved with the sparse direct solver UMFPACK (backslash command of MATLAB). We did not find any significant difficulty with this procedure.

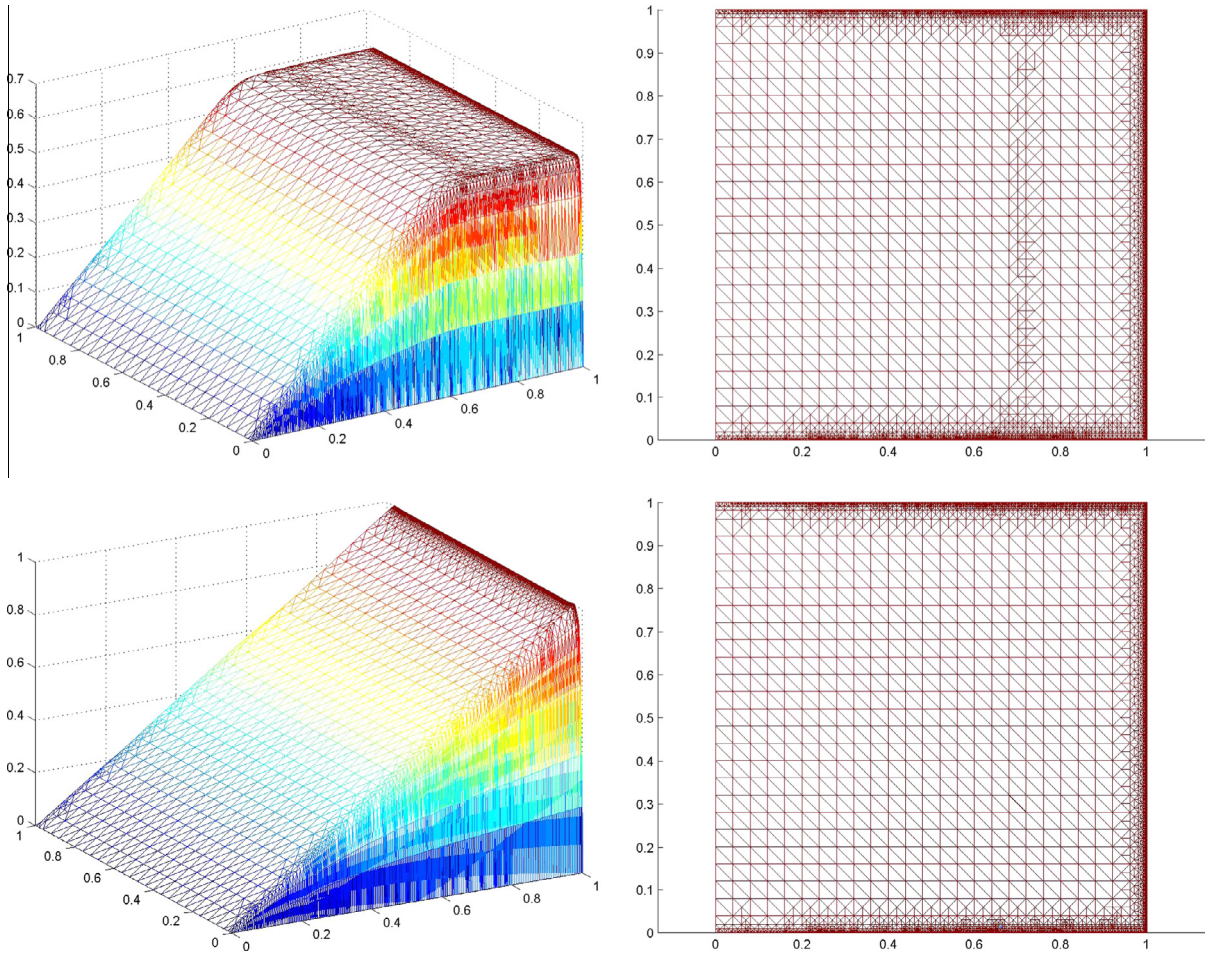

Fig. 3. Example 3. Solution computed with the SUPG-based method (left) and corresponding adaptive mesh (right), at time $t = 0.6$ (top) and $t = 1.2$ (bottom).

Example 3. *Example with an exponential boundary layer and two characteristics boundary layers.* This example is defined by (1) with $\mathbf{b} = (1,0)^T,\ c = 0,\ u_0 = 0,\ f = 1,\ \varepsilon = 10^{-4}$, and homogeneous Dirichlet boundary conditions. The boundary $x = 1$ is the outflow boundary and an exponential boundary layer appears there. At the tangential boundaries $y = 0$ and $y = 1$, characteristic or parabolic boundary layers develop.

The solutions and meshes for the SUPG-based algorithm, obtained with the tolerances $\mathrm{TOL}_1 = 10^{-3}$ and $\mathrm{TOL}_2 = 10^{-3}/20$, are presented in Fig. 3. Comparing them with [5, Fig. 3], the results of both adaptive algorithms are, on the first glance, similar. However, a detailed inspection shows that the SUPG-based algorithm uses considerably less mesh cells and computes a more accurate solution. For instance, at time $t = 0.6$, the mesh produced by the Galerkin-based algorithm has 213704 cells while for the SUPG-based algorithm it has 171931 cells. At the final time, the difference increases and the meshes have 470192 cells for the Galerkin-based algorithm and 222581 cells for the SUPG-based algorithm.

In Fig. 4, a detail of the mesh generated by the SUPG-based algorithm at time $t = 1.2$ is presented. The mesh shows a very different refinement according to the different kinds of layers: the exponential layer at $x = 1$ and the parabolic layer at $y = 0$.

Fig. 5 shows a plot of the solutions computed with both adaptive methods along the line $x = 0.5$. With the SUPG-based algorithm, there are no oscillations at the parabolic layer, which is in contrast to the Galerkin-based algorithm. Hence, the solution computed with the SUPG-based algorithm appears to be more accurate. With both algorithms, the solutions do not exhibit spurious oscillations in the exponential layer, see [5, Fig. 5] for the Galerkin-based algorithm. The picture for the SUPG-base algorithm is nearly identical and its presentation is omitted here.

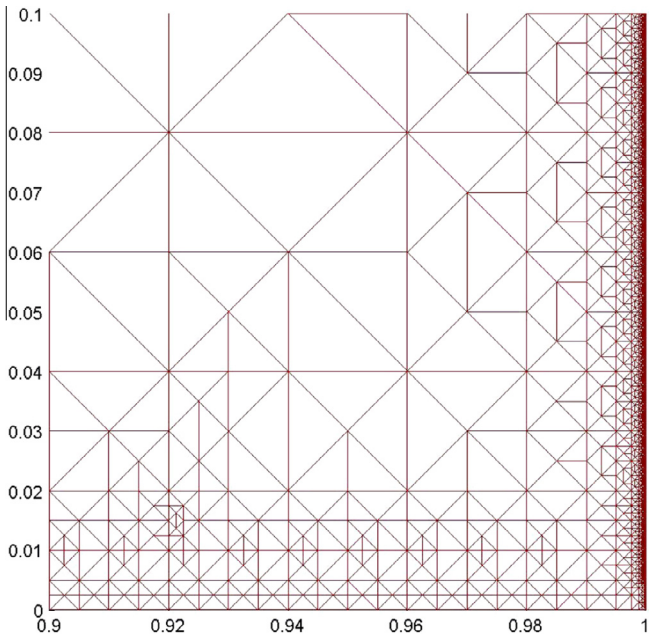

Fig. 4. Example 3. Zoom of bottom right picture of Fig. 3.

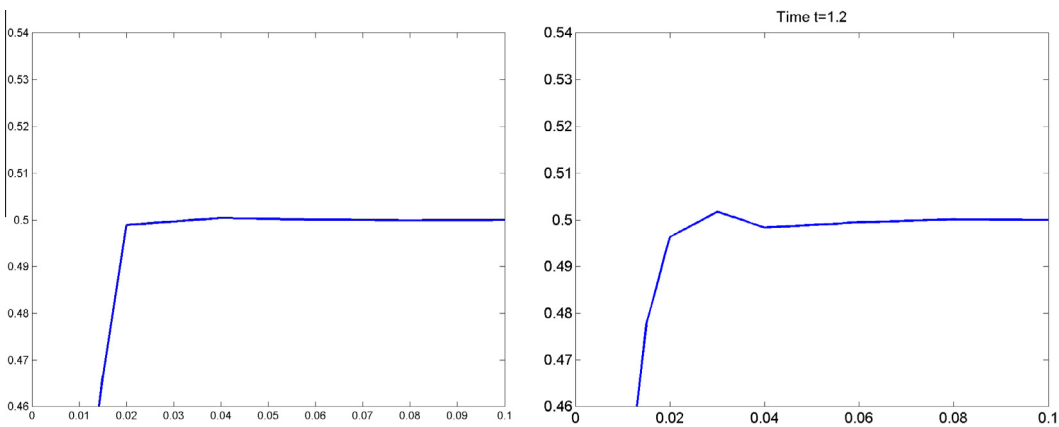

Fig. 5. Example 3. Zoom of the parabolic layer at $x = 0.5$ and $y = 0$ at time $t = 1.2$, SUPG-based algorithm (left), Galerkin-based algorithm (right).

In this example, the solution computed with the SUPG-based algorithm is more accurate than the solution obtained with the Galerkin-based algorithm and the grids generated with the SUPG-based algorithm possess considerably less mesh cells.

Example 4. *Rotating three body problem.* This example considers a configuration of three geometrical bodies that rotate clockwise. It was studied in the hyperbolic limit, $\varepsilon = 10^{-20}$, e.g., in [17,19].

In this example, the rotation is driven by $\mathbf{b} = (y - 0.5, 0.5 - x)^T$. The forcing term is $f = 0$ and it is $c = 0$ as well. Homogeneous Dirichlet boundary conditions were imposed. Results obtained for the value $\varepsilon = 10^{-6}$ will be reported here. The initial condition, consisting of three disjoint bodies, is represented in Fig. 6. It is zero outside the three bodies. More precisely, for a given (x_0, y_0), let $r(x,y) = \sqrt{(x-x_0)^2 + (y-y_0)^2}/r_0$. The center of the slotted cylinder is located at $(x_0, y_0) = (0.5, 0.75)$ and its shape is defined by

$$u(0;x,y) = \begin{cases} 1 & \text{if } r(x,y) \leqslant 1,\ |x - x_0| \geqslant 0.0225 \text{ or } y \geqslant 0.85, \\ 0 & \text{otherwise.} \end{cases}$$

The hump at the left-hand side is defined by $(x_0, y_0) = (0.25, 0.5)$ and

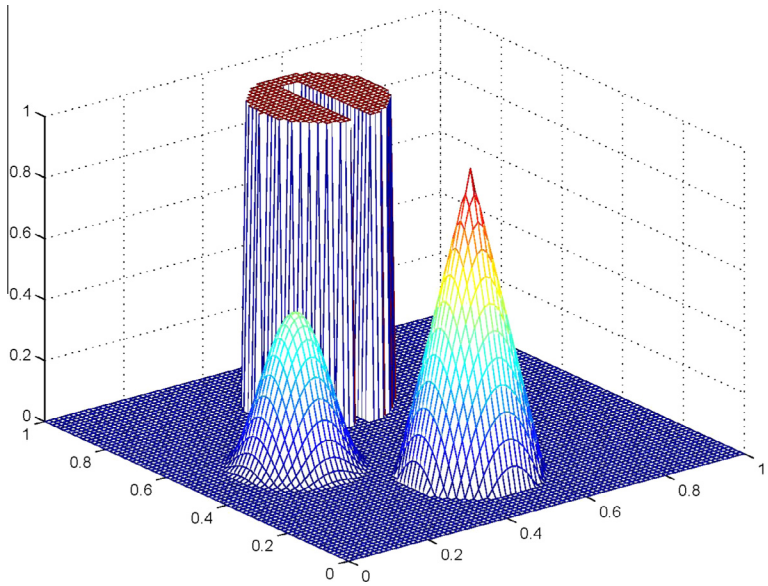

Fig. 6. Example 4. Initial condition.

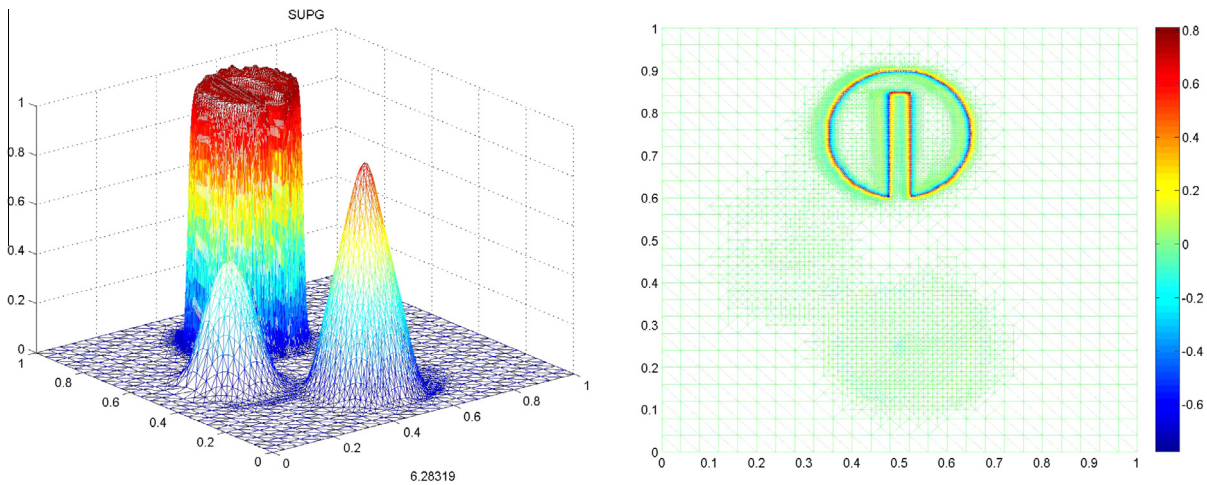

Fig. 7. Example 4. SUPG-based algorithm, approximation after one rotation (left) and difference to the initial condition (right).

$$u(0;x,y) = \frac{1}{4}(1 + \cos(\pi \min\{r(x,y), 1\})).$$

The conical body on the lower part is given by $(x_0, y_0) = (0.5, 0.25)$ and

$$u(0;x,y) = 1 - r(x,y).$$

In the SUPG-based algorithm, the tolerances $\mathrm{TOL}_1 = 10^{-3}$ and $\mathrm{TOL}_2 = 10^{-3}/20$ were used. Again, one can observe that the Galerkin-based approach generates meshes with much more cells: 391314 vs. 22250 at the final time, that is a difference of more than one order of magnitude. Comparing the results in Fig. 7 with [5, Figs. 9 and 10], one can see that both solutions are of similar quality. While the fineness of the mesh leads to a somewhat more accurate approximation of the cylinder with the Galerkin-based algorithm, the SUPG-based method computes a more accurate solution in the rest of the domain. The variation of the solution computed with the SUPG-based algorithm at time 2π is given by $\max(u_h) - \min(u_h) = 1.0509 - (-0.0635) = 1.1144$. This quantity can be taken as a measure of the spurious oscillations still remaining in the numerical approximation, in particular at the cylinder. The corresponding value for the Galerkin-based method 1.0159 from [5] shows the better accuracy obtained with this method at the cylinder (with about 17 times more mesh cells).

Fig. 8 presents the adaptive meshes for the SUPG-based algorithm at times $t \in \{\pi/2, \pi, 3\pi/2, 2\pi\}$. It can be observed that this algorithm is able to follow very well the movement of the three bodies, thereby refining only around them and keeping the mesh essentially coarse in the rest of the domain. This behavior is in strong contrast to the Galerkin-based algorithm [5, Fig. 11], which produces, in the whole time interval, a refinement of a large part of the domain.

In this example, it was possible to compute with the SUPG-based algorithm a qualitatively similar solution as with the Galerkin-based algorithm on a grid with much less cells. The good capturing of the layers and coarsening of the mesh at the smooth parts of the solution by the SUPG-based algorithm could be observed well.

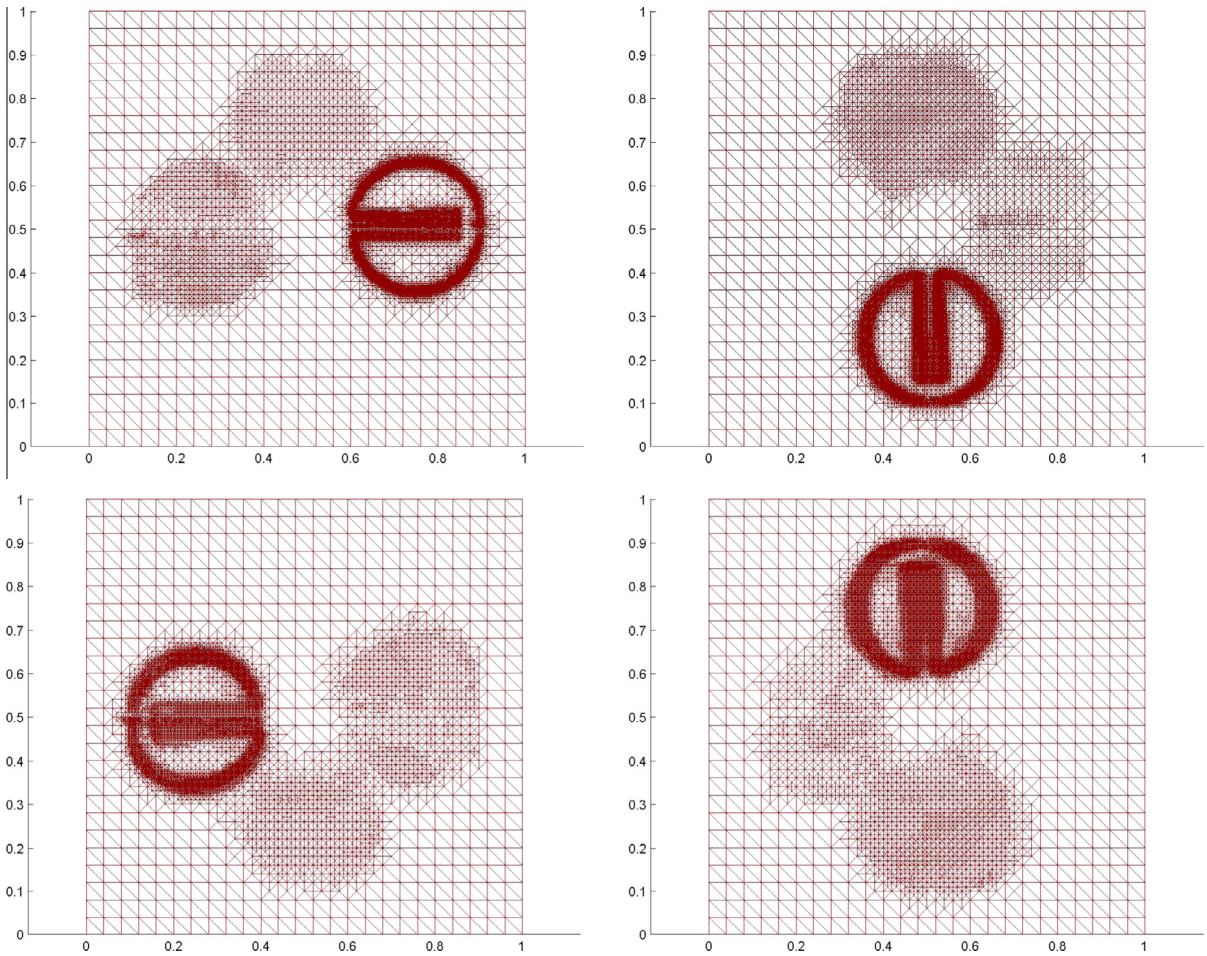

Fig. 8. Example 4. SUPG-based method, computational meshes at times $t \in \{\pi/2, \pi, 3\pi/2, 2\pi\}$.

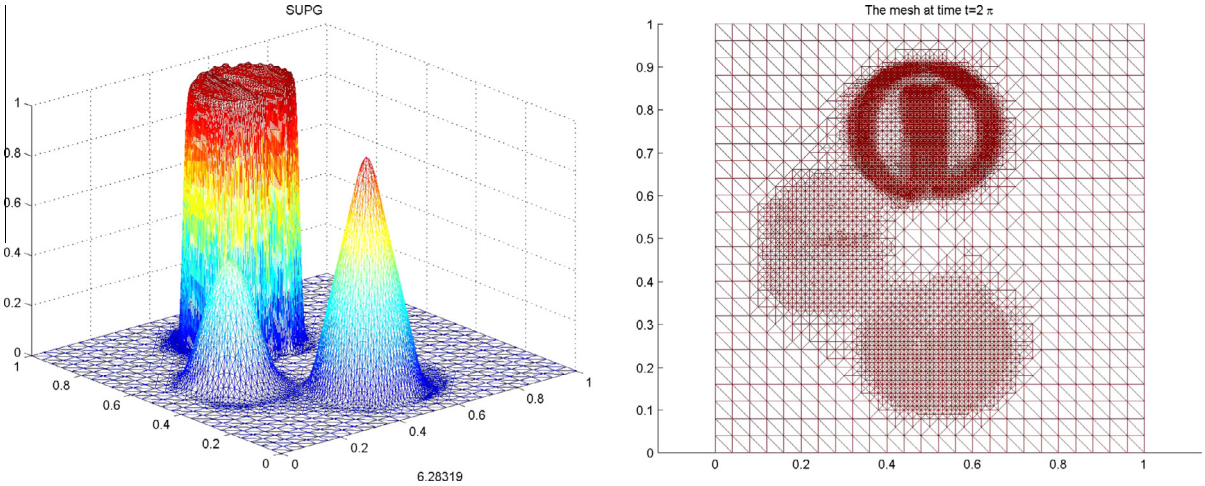

Fig. 9. Example 4. Approximation at time 2π obtained with the error estimator (10) and the corresponding mesh.

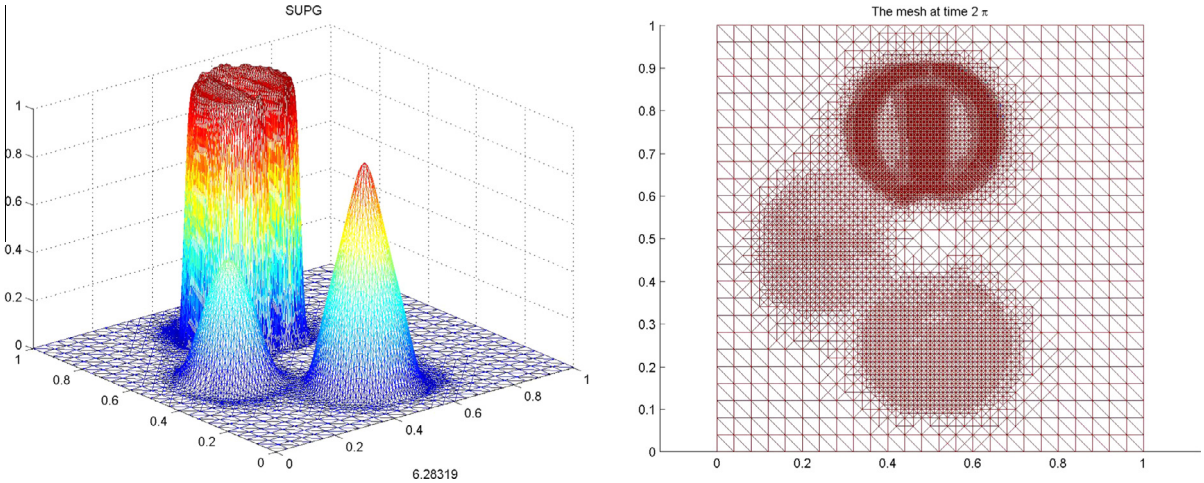

Fig. 10. Example 4. Approximation at time 2π obtained with the error estimator (11) and the corresponding mesh.

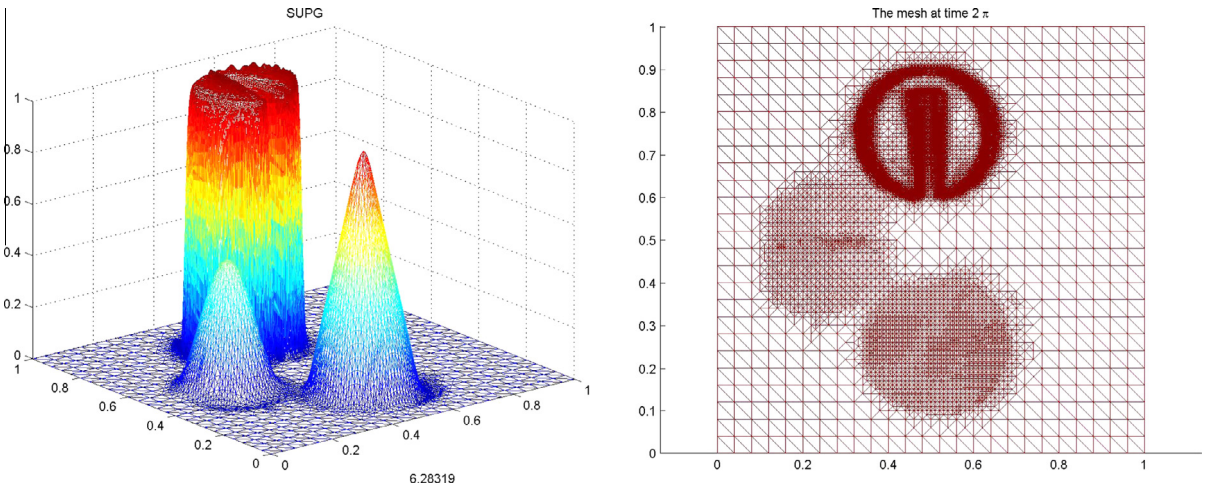

Fig. 11. Example 4. Approximation at time 2π obtained with the error estimator (12) and the corresponding mesh.

Figs. 9–11 present the solutions which were computed using instead of (28) the local error estimators (10)–(12), respectively. The size of the variations in the final solutions are $\max(u_h) - \min(u_h) = 1.0619 - (-0.0597) = 1.1215$, $\max(u_h) - \min(u_h) = 1.0688 - (-0.00613) = 1.1301$, and $\max(u_h) - \min(u_h) = 1.0491 - (-0.0519) = 1.1010$, respectively, and the number of cells at the final meshes are 23 106, 20 600, and 42 684. The only solution giving a slightly smaller variation than the solution computed with the estimator (28) is the last one, associated to the error estimator (12). However, the number of cells needed to compute this approximation was almost twice as large as the one with (28). Altogether, one can conclude that, in this example, the SUPG-based algorithm with all estimators (10)–(12), and (28) shows a similar behavior.

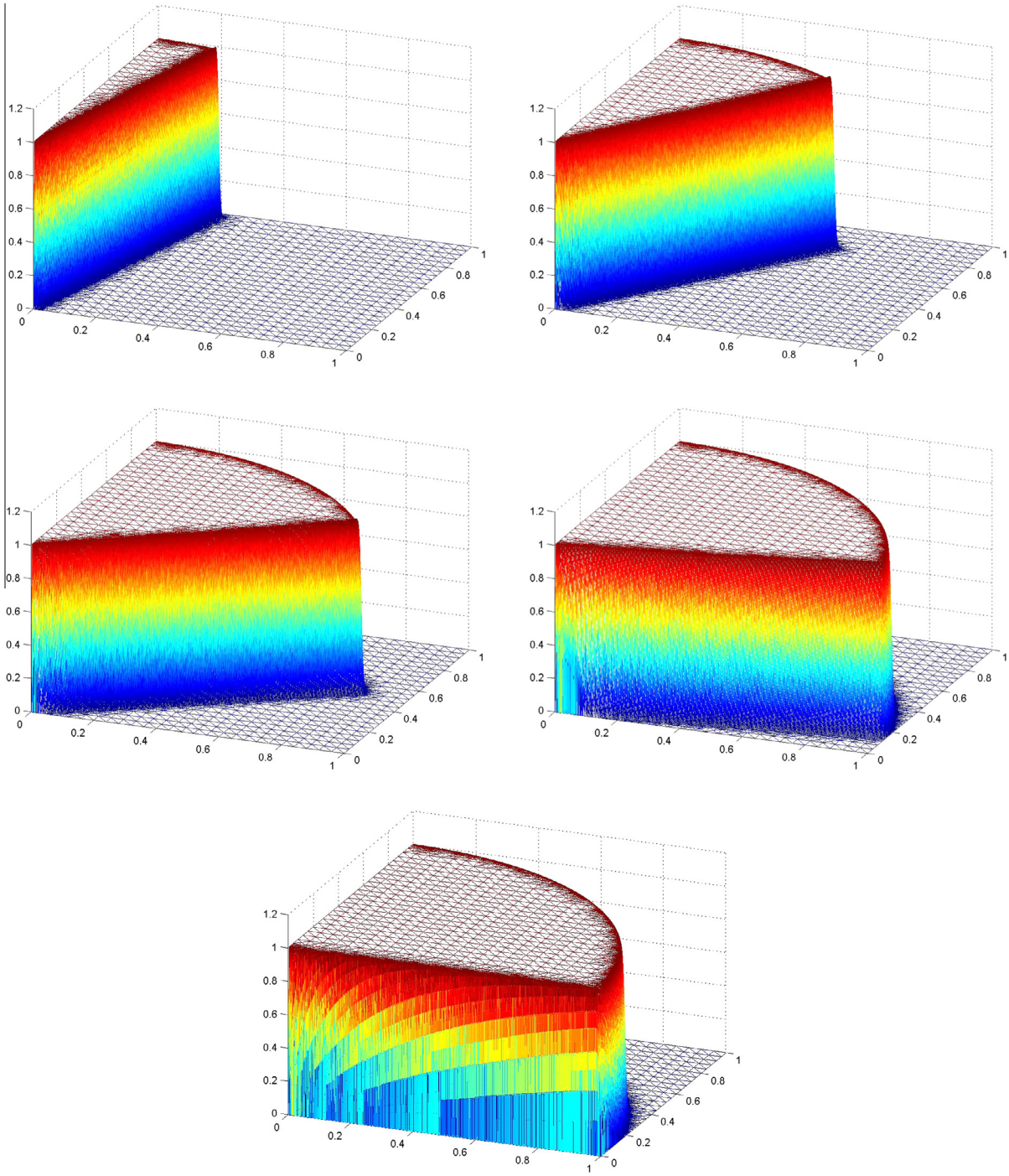

Fig. 12. Example 5. SUPG-based algorithm, solutions at times $t \in \{0.2, 0.6, 1.0, 1.4, 1.8\}$.

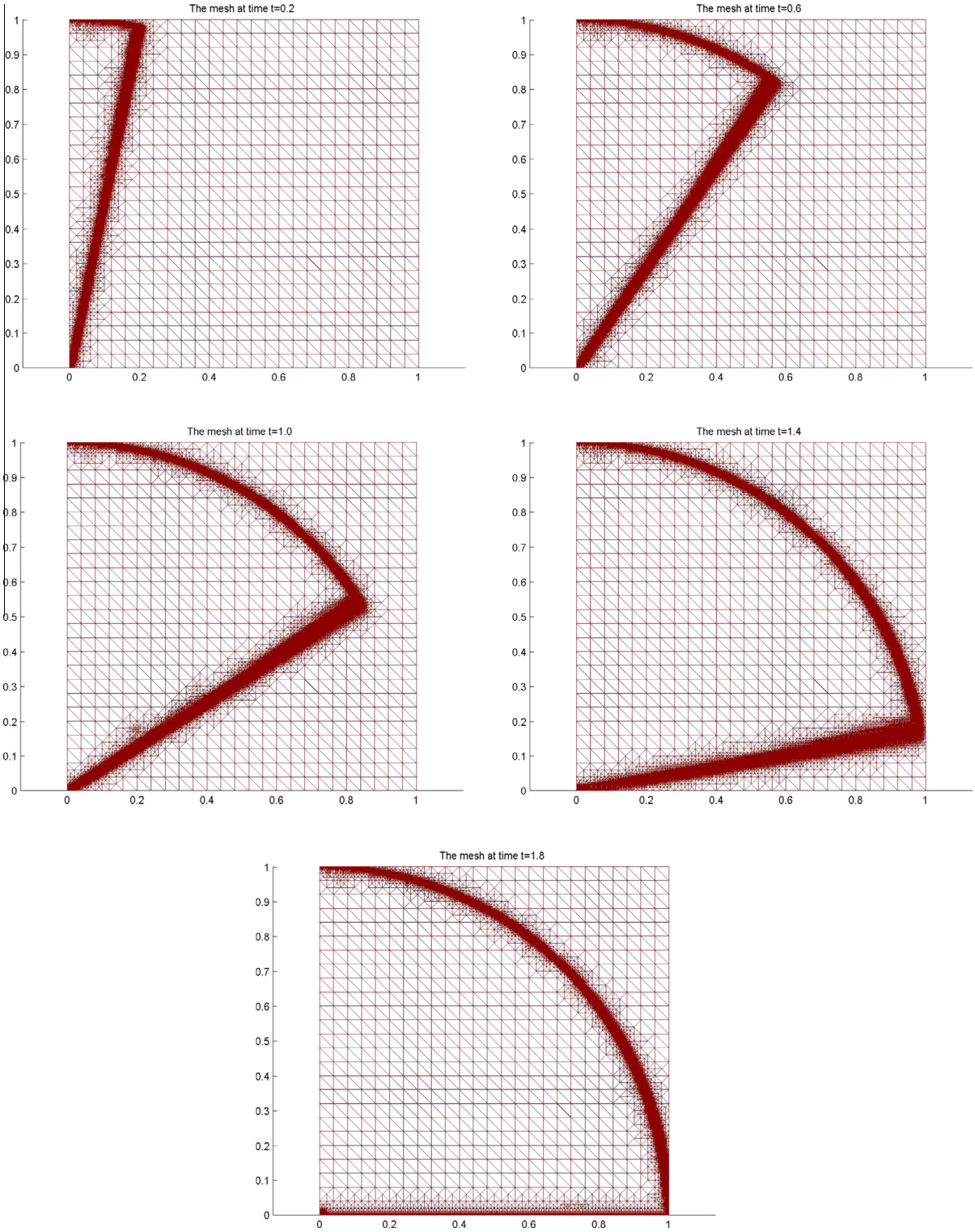

Fig. 13. Example 5. SUPG-based algorithm, meshes at times $t \in \{0.2, 0.6, 1.0, 1.4, 1.8\}$.

Example 5. *Example with moving internal layers and a developing exponential layer.* Problem (1) is considered with $\mathbf{b} = (y, -x)^T$, $c = 0$, $u_0 = 0$, $f = 0$, and $\varepsilon = 10^{-5}$. Dirichlet boundary conditions are given by $u = 0$ at the boundaries $x = 1$, $y = 0$, and $y = 1$ and by $u = 1$ at $x = 0$. In this example, a front is generated at the side $x = 0$ and, as time evolves, it moves through the domain, thereby rotating around the origin. A curved interior layer is created. When the front arrives at the boundary $y = 0$, an exponential boundary layer is generated there. In the steady state, the curved internal layer coexists with the exponential boundary layer at $y = 0$.

In the simulations, the tolerances $\mathrm{TOL}_1 = 2 \cdot 10^{-4}$ and $\mathrm{TOL}_2 = \mathrm{TOL}_1/20$ were used. Fig. 12 represents the numerical solutions computed with the SUPG-based algorithm at times $t \in \{0.2, 0.6, 1.0, 1.4, 1.8\}$. One can observe that the solutions are essentially non-oscillating and that the meshes are adapted to the characteristics of the solution.

The meshes corresponding to these numerical solutions are plotted in Fig. 13. Comparing with [5, Fig. 13], where the meshes generated with the Galerkin-based approach are shown, one gets the impression that the behavior of the two algorithms is not very different. However, a look at the actual number of mesh cells gives for the Galerkin-based algorithm 92017, 119138, 144336, 180375, and 2210665 at the five different times, respectively, while the numbers of cells for the SUPG-based algorithm are 99240, 115948, 122877, 116690, and 1313316. Except for the first mesh at $t = 0.2$, where both algorithms produce almost the same number of cells, the SUPG-based algorithm generates meshes with a smaller number of cells. In addition, the ratio of the number of cells generated with the Galerkin-based and the SUPG-based algorithm increases as time evolves, being for the final time equal to 1.68.

In this example, the solutions of both algorithms are similar but the SUPG-based algorithm obtained its solution on a mesh with considerably less cells.

Remark 3. From the numerical experiments, we infer that in spite of using adaptive mesh refinement, the number of mesh cells may still be too high in some cases. The use of anisotropic mesh refinement could possibly help to alleviate this problem.

5. Summary

In this paper, an adaptive algorithm for evolutionary convection–diffusion–reaction equations was proposed which allows the application of the natural extension of any error estimator which is known for the steady-state problem. This approach is based on the observation that the SUPG solution of the evolutionary problem is also the solution of an appropriate steady-state problem. In the derivation of the algorithm, the heuristic argument was used that the first term on the right-hand side of (7) is of higher order. This argument was supported in one dimension with an error analysis. In the numerical studies, in particular the residual-based estimator from [18] was considered. It was shown for an example with interior layer that also for the time-dependent problem the error estimates in the SUPG norm are robust with respect to the polynomial degree of the finite element space, the mesh width, and the value of the diffusion. Comprehensive comparisons with the adaptive method proposed in [5] showed that the new method obtains in general more accurate solutions on grids with considerably less mesh cells.

References

[1] B. Achchab, A. Benjouad, M. El Fatini, A. Souissi, G. Warnecke, Robust a posteriori error estimates for subgrid stabilization of non-stationary convection dominated diffusive transport, Appl. Math. Comput. 218 (9) (2012) 5276–5291.
[2] R. Bermejo, J. Carpio, An adaptive finite element semi-Lagrangian implicit-explicit Runge–Kutta–Chebyshev method for convection dominated reaction–diffusion problems, Appl. Numer. Math. 58 (1) (2008) 16–39.
[3] Alexander N. Brooks, Thomas 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. Eng. 32 (1–3) (1982) 199–259. FENOMECH '81, Part I (Stuttgart, 1981).
[4] Javier de Frutos, Bosco García-Archilla, Julia Novo, A posteriori error estimates for fully discrete nonlinear parabolic problems, Comput. Methods Appl. Mech. Eng. 196 (35–36) (2007) 3462–3474.
[5] Javier de Frutos, Bosco García-Archilla, Julia Novo, An adaptive finite element method for evolutionary convection dominated problems, Comput. Methods Appl. Mech. Eng. 200 (49–52) (2011) 3601–3612.
[6] Javier de Frutos, Bosco García-Archilla, Julia Novo, Nonlinear convection–diffusion problems: fully discrete approximations and a posteriori error estimates, IMA J. Numer. Anal. 31 (4) (2011) 1402–1430.
[7] Javier de Frutos, Julia Novo, A posteriori error estimation with the *p*-version of the finite element method for nonlinear parabolic differential equations, Comput. Methods Appl. Mech. Eng. 191 (43) (2002) 4893–4904.
[8] Vít Dolejší, Alexandre Ern, Martin Vohralík, A framework for robust a posteriori error control in unsteady nonlinear advection-diffusion problems, SIAM J. Numer. Anal. 51 (2) (2013) 773–793.
[9] Philip M. Gresho, David F. Griffiths, David J. Silvester, Adaptive time-stepping for incompressible flow. I. Scalar advection-diffusion, SIAM J. Sci. Comput. 30 (4) (2008) 2018–2054.
[10] Guillermo Hauke, Daniel Fuster, Mohamed H. Doweidar, Variational multiscale a-posteriori error estimation for multi-dimensional transport problems, Comput. Methods Appl. Mech. Eng. 197 (33–40) (2008) 2701–2718.
[11] Friedrich K. Hebeker, Rolf Rannacher, An adaptive finite element method for unsteady convection-dominated flows with stiff source terms, SIAM J. Sci. Comput. 21 (3) (1999) 799–818 (electronic).
[12] Paul Houston, Endre Süli, Adaptive Lagrange–Galerkin methods for unsteady convection–diffusion problems, Math. Comput. 70 (233) (2001) 77–106.
[13] Xiaozhe Hu, Young-Ju Lee, Jinchao Xu, Chen-Song Zhang, On adaptive Eulerian–Lagrangian method for linear convection–diffusion problems, J. Sci. Comput. 58 (1) (2014) 90–114.
[14] T.J.R. Hughes, A. Brooks, A multidimensional upwind scheme with no crosswind diffusion, in: Finite Element Methods for Convection Dominated Flows (Papers, Winter Ann. Meeting Am. Soc. Mech. Eng. New York, 1979), volume 34 of AMD, pages 19–35. Am. Soc. Mech. Eng. (ASME), New York, 1979.

[15] Volker John, A numerical study of a posteriori error estimators for convection–diffusion equations, Comput. Methods Appl. Mech. Eng. 190 (5–7) (2000) 757–781.
[16] Volker John, Petr Knobloch, On spurious oscillations at layers diminishing (SOLD) methods for convection–diffusion equations. I. A review, Comput. Methods Appl. Mech. Eng. 196 (17–20) (2007) 2197–2215.
[17] Volker John, Julia Novo, Error analysis of the SUPG finite element discretization of evolutionary convection–diffusion–reaction equations, SIAM J. Numer. Anal. 49 (3) (2011) 1149–1176.
[18] Volker John, Julia Novo, A robust SUPG norm a posteriori error estimator for stationary convection–diffusion equations, Comput. Methods Appl. Mech. Eng. 255 (2013) 289–305.
[19] Volker John, Ellen Schmeyer, Finite element methods for time-dependent convection–diffusion–reaction equations with small diffusion, Comput. Methods Appl. Mech. Eng. 198 (3-4) (2008) 475–494.
[20] Claes Johnson, Uno Nävert, Juhani Pitkäranta, Finite element methods for linear hyperbolic problems, Comput. Methods Appl. Mech. Eng. 45 (1-3) (1984) 285–312.
[21] Uno Nävert, A finite element method for convection–diffusion problems (Ph.D. thesis), Chalmers University of Technology, Göterborg, 1982.
[22] Marco Picasso, Virabouth Prachittham, An adaptive algorithm for the Crank–Nicolson scheme applied to a time-dependent convection–diffusion problem, J. Comput. Appl. Math. 233 (4) (2009) 1139–1154.
[23] Hans-Görg Roos, Martin Stynes, Lutz Tobiska, Robust numerical methods for singularly perturbed differential equations, Springer Series in Computational Mathematics, second ed., vol. 24, Springer-Verlag, Berlin, 2008. Convection–diffusion–reaction and flow problems.
[24] A.H. Schatz, L.B. Wahlbin, On the quasi-optimality in L_∞ of the $\dot{H}^1$-projection into finite element spaces, Math. Comput. 38 (157) (1982) 1–22.
[25] Martin Stynes, Steady-state convection–diffusion problems, Acta Numer. 14 (2005) 445–508.
[26] R. Verfürth, A posteriori error estimation and adaptive mesh-refinement techniques, J. Comput. Appl. Math. 50 (1-3) (1994) 67–83.
[27] R. Verfürth, A posteriori error estimators for convection–diffusion equations, Numer. Math. 80 (4) (1998) 641–663.
[28] R. Verfürth, Robust a posteriori error estimates for nonstationary convection–diffusion equations, SIAM J. Numer. Anal. 43 (4) (2005) 1783–1802 (electronic).
[29] Guo Hui Zhou, Rolf Rannacher, Pointwise superconvergence of the streamline diffusion finite-element method, Numer. Methods Partial Differ. Equ. 12 (1) (1996) 123–145.
[30] Guohui Zhou, How accurate is the streamline diffusion finite element method?, Math Comput. 66 (217) (1997) 31–44.