International Journal of Applied Mathematics

Volume 28 No. 6 2015, 667-689
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
doi: http://dx.doi.org/10.12732/ijam.v28i6.3

ON A STABILIZED FINITE ELEMENT METHOD WITH MESH ADAPTIVE PROCEDURE FOR CONVECTION–DIFFUSION PROBLEMS

M. Farhloul[1 §], A. Serghini Mounim[2], A. Zine[3]

[1]Département de Mathématiques et de Statistique
Université de Moncton
Moncton, N.B., E1A 3E9, CANADA

[2]Department of Mathematics and Computer Science
Laurentian University
Sudbury, Ontario, P3E 2C6, CANADA

[3]Département de Mathématiques et Informatique
Université de Lyon
Institut Camille Jordan, CNRS–UMR5208
Ecole Centrale de Lyon, 36 av. Guy de Collongue
69134 Ecully Cedex, FRANCE

Abstract: Computing solutions of convection–diffusion equations is an important and challenging problem from the numerical point of view. We present in this work a numerical scheme to study this problem. The scheme combines a stabilized finite element method introduced in [Serghini Mounim, A stabilized finite element method for convection–diffusion problems, Mumer. Methods Partial Differential Eq 28: 1916-1943, 2012], with an adaptive mesh refinement procedure which is based on the residual a posteriori error estimators. It is worthwhile to point out that the numerical results indicate that the stabilization parameter introduced in [Serghini Mounim, A stabilized finite element method for convection–diffusion problems, *Numer. Methods Partial Differential Eq.* **28** (2012), 1916-1943] gives much better results than the standard Streamline upwind/Petrov–Galerkin (SUPG) one.

Received: July 27, 2015

§Correspondence author

AMS Subject Classification: 65N30, 65N15, 65N50
Key Words: convection–diffusion equations, streamline upwind/Petrov-Galerkin method, residual-free bubble method, residual *a posteriori* error estimates, adaptive mesh refinement

1. Introduction

Scalar convection–diffusion problems govern a variety of physical phenomena. The equation governing such natural phenomena is composed of a diffusive part and a convective part which dominates in general. The mathematical analysis and the numerical solution of convection–dominated equations have been of great interest over the last decades, for an overview of new developments see Roos et al. [22]. The weaknesses, particularly in the convection dominated regime, of classical Galerkin finite element methods to treat such problems have focused attention on developing stable approximations. To enhance the stability and accuracy, various stabilization techniques have been developed. Among them, the most popular is the Streamline upwind/Petrov–Galerkin (SUPG) introduced by Brooks and Hughes [9]. This stabilized method consists in adding a local amount $O(\tau_K)$ of diffusion terms to the Galerkin formulation preserving the consistency of the variational formulation (only in the convection direction). The accuracy is strongly connected to the parameters contained in the additional terms. An important drawback of the method is that the amount of added diffusion has to be tuned by a parameter τ. For an overview of these successful approaches, see [11] and references therein.

Baiocchi et al. [2] and Brezzi et al. [3] showed that the SUPG method is linked to the process of adding suitable bubble functions. The optimal value of the streamline diffusion parameter τ is then related to the best choice of the bubble shape. This last method, proposed by Brezzi–Russo [8], is called Residual–Free Bubble (RFB). The RFB method requires the solution of a boundary value problem in each triangle K. However, the exact computation of this function is as difficult as the original convection–diffusion problem itself. Approximations of this function have been considered in some papers.

In a recent work, Serghini Mounim [24] proposed a new procedure for deriving a stabilized method which is "parameter-free" to solve the convection-diffusion equation. This approach leads to a streamline–diffusion finite element method with an explicit formula for the stability parameter that is capable of adapting to all regimes.

Despite the recent progress there is still room for improvement. So we added

to these stabilization techniques an adaptive mesh refinement which could possibly help to prevent the problem of sharp boundary layers. The use of *a posteriori* error estimators for estimating the global error as well as for obtaining information for adaptive mesh–refinement techniques is nowadays a standard component in numerical codes for solving partial differential equations. During the last decades, fundamental and general approaches for analyzing a posteriori error estimators for finite element solutions of many classes of partial differential equations have been developed, see for example [26, 1, 21, 18] and references therein.

The drawback with most of stabilized methods is that the solution layers are not very well resolved, because of the numerical diffusion added to the discretization. On the other hand, most of the standard error estimators involve equivalence constants depending on negative powers of the diffusion parameter, which lead to very poor results in advective dominated case. A way to overcome these difficulties is to combine the efficiency of the stabilization techniques and an adaptive mesh refinement procedure.

In this work, we introduce and analyze from the numerical point of view an adaptive scheme to efficiently solve the convection–diffusion problem. We perform several numerical experiments to show the effectiveness of our approach to capture the inner and boundary layers. Furthermore, we show the superiority of the RFB method compared to the SUPG method.

The paper is organized as follows. In Sections 2 and 3, the scalar convection–diffusion equations and their discretization are presented with a short survey of the stabilization techniques. Section 4 contains a description of the *a posteriori* error estimators. In Section 5, we deal with a general presentation of numerical results. The numerical tests with respect to the estimation of the global error are presented in Section 5.1 and with respect to the adaptive mesh refinement in Section 5.2. Finally, Section 6 presents some conclusions and prospects.

2. Scalar Convection–Diffusion Problem

Let Ω be a polygonal domain of $\mathbb{R}^2$ of boundary $\Gamma = \Gamma_D \cup \Gamma_N$, with $\Gamma_D \cap \Gamma_N = \emptyset$ and $meas(\Gamma_D) > 0$. Let $f \in L^2(\Omega)$, $g_D, g_N \in L^2(\Gamma)$. We consider the following scalar convection-diffusion problem:

$$\begin{cases} -\mathrm{div}\,(\epsilon \nabla u) + \boldsymbol{\beta} \cdot \nabla u = f \quad \text{in} \quad \Omega, \\ u = g_D \quad \text{on} \quad \Gamma_D, \\ \varepsilon \nabla u \cdot \mathbf{n} = g_N \quad \text{on} \quad \Gamma_N, \end{cases} \tag{1}$$

where ∇ and div denote the gradient and divergence operators respectively and $\mathbf{n}$ the outward unit normal to Γ.

We shall assume that $\epsilon > 0$ is a positive constant, the velocity field $\boldsymbol{\beta} \in \left(W^{1,\infty}(\Omega)\right)^2$ is such that $\mathrm{div}(\boldsymbol{\beta}) = 0$ and, for the sake of simplicity, we take $g_D = 0$ and $g_N = 0$.

The variational formulation corresponding to (1) reads:

$$\begin{cases} u \in \mathcal{V}; \\ \mathcal{A}(u, v) = \mathcal{L}(v), \quad \forall\, v \in \mathcal{V}, \end{cases} \tag{2}$$

where $\mathcal{V} = \left\{ v \in H^1(\Omega);\ v|_{\Gamma_D} = 0 \right\}$ and

$$\mathcal{A}(u, v) = \epsilon \int_\Omega \nabla u \cdot \nabla v + \int_\Omega (\boldsymbol{\beta} \cdot \nabla u) v, \quad \mathcal{L}(v) = \int_\Omega f v.$$

The problem (2) fits into the Lax-Milgram framework. Both continuity on $\mathcal{V} \times \mathcal{V}$ and the $\mathcal{V}$–ellipticity of the bilinear form $\mathcal{A}(.,.)$ hold true (see, e.g., [20]). This problem is then well posed for any $\epsilon > 0$ and $\boldsymbol{\beta}$, a velocity field, such that $\mathrm{div}(\boldsymbol{\beta}) = 0$.

3. Standard Galerkin Finite Element Method and Stabilization Techniques

For a finite element discretization of (2), the domain Ω is partitioned into a finite number of non–overlapping elements K:

$$\overline{\Omega} = \bigcup_{K \in \mathcal{T}_h} \overline{K},$$

where $\mathcal{T}_h$ is a regular family of triangles K. For any triangle K, h_K is the diameter (longest edge) of K, E is an edge of K and h_E, the length of the edge E. Let ∂K_- and ∂K_+ denote, respectively, the inflow and outflow parts of the ∂K:

$$\partial K_- = \{ \boldsymbol{x} \in \partial K;\ \boldsymbol{\beta} \cdot \mathbf{n}(\boldsymbol{x}) < 0 \}, \quad \partial K_+ = \{ \boldsymbol{x} \in \partial K;\ \boldsymbol{\beta} \cdot \mathbf{n}(\boldsymbol{x}) \geq 0 \}.$$

The standard Galerkin approximation to the variational problem reads:

$$\begin{cases} u_L \in \mathcal{V}_L; \\ \mathcal{A}(u_L, v_L) = \mathcal{L}(v_L), \qquad \forall v_L \in \mathcal{V}_L, \end{cases} \tag{3}$$

where the subspace $\mathcal{V}_L$ of $\mathcal{V}$ consisting of continuous functions that are piecewise polynomials on $\mathcal{T}_h$:

$$\mathcal{V}_L = \{ v \in \mathcal{V};\ v_{|K} \in P_1(K), \forall K \in \mathcal{T}_h \},$$

and where $P_1(K)$ denotes the space of linear functions defined on the triangle K.

As mentioned in Introduction, despite the apparent simplicity of the problem (2), its numerical solution is still a challenge when advection terms are strongly dominant. More precisely, the FE solution of (2) is characterized by the local non–dimensional Péclet number:

$$\mathcal{P}_{e_K} = \frac{\| \boldsymbol{\beta} \| h_K}{2\epsilon}, \quad \|\boldsymbol{\beta}\| = \|\boldsymbol{\beta}\|_{\infty,K}\,. \tag{4}$$

In general, oscillatory solutions are observed for high Péclet, i.e., $\mathcal{P}_{e_K} \gg 1$; for example, if there are any boundary layers which are not resolved by the mesh. An oscillation–free solution cannot be achieved with the Galerkin finite element method GFEM. As a remedy, particularly for the high $\mathcal{P}_{e_K}$ regime, stabilized FEMs were proposed.

3.1. Streamline Upwind/Petrov–Galerkin (SUPG)

Following [9, 10], to ensure stability at high Péclet numbers ($\mathcal{P}_{e_K} \gg 1$), stabilization terms should be added to the Galerkin variational formulation. The SUPG method consists then of adding to the original bilinear form $\mathcal{A}(.,.)$ and linear form $\mathcal{L}(\cdot)$ terms which introduce a suitable amount of artificial diffusion in the streamline direction. For the convection diffusion problem (1), the SUPG method reads:

$$\begin{cases} u_L \in \mathcal{V}_L; \\ \mathcal{A}(u_L, v_L) + \displaystyle\sum_{K\in\mathcal{T}_h} \tau_K^{\text{supg}} \int_K (\boldsymbol{\beta}\cdot\nabla u_L - f)\,(\boldsymbol{\beta}\cdot\nabla v_L) = \mathcal{L}(v_L), \quad \forall\, v_L \in \mathcal{V}_L, \end{cases}$$

where $\tau_K^{\text{supg}} \geq 0$ is a stabilization parameter. Several definitions for τ_K^{supg} may be found in the literature, but its value is not determined precisely by the available theory. One popular choice for the stabilizing parameter for continuous piecewise linear finite elements on quasi-uniform mesh and element-wise constant f is (see [10]):

$$\tau_K^{\text{supg}} = \frac{h_K}{2\| \boldsymbol{\beta} \|} \left(\coth(\mathcal{P}_{e_K}) - \frac{1}{\mathcal{P}_{e_K}} \right). \tag{5}$$

Since, $\lim_{x\to+\infty}(\coth(x)-\frac{1}{x})=1$ and $\lim_{x\to 0^+}\dfrac{\coth(x)-\dfrac{1}{x}}{x}=\dfrac{1}{3}$, in practice it is usual to use the asymptotic expressions of (5):

$$\tau_K^{\text{supg}}=\begin{cases}\dfrac{h_K}{2\|\boldsymbol{\beta}\|} & \text{if } \mathcal{P}_{e_K}\geq 1, \text{ (convection–dominated)},\\[2ex] \dfrac{h_K^2}{12\epsilon} & \text{if } \mathcal{P}_{e_K}<1, \text{ (diffusion–dominated)}.\end{cases} \tag{6}$$

The reason for the popularity of the stabilization parameter (5) is that it gives in one dimension a nodally exact solution and monotonicity preserving approximant (see [10, 9]). However, the optimality property of (5) in one dimension does not generalize to the two-dimensional case, even for a constant function f. Indeed, this parameter has been defined in the 2D case by a simple analogy with the one-dimensional formula.

A way to intrinsically recover the value of τ_K^{supg} is to use the residual free bubbles (RFB) methodology [8, 12, 23, 5], or the variational multiscale method [16, 17].

3.2. Residual Free Bubbles (RFB) approach

3.2.1. General Methodology

Following [4], we write the Galerkin variational formulation of (1) without a stabilizing term:

$$\begin{cases} u_h\in\mathcal{V}_h;\\ \mathcal{A}(u_h,v_h)=\mathcal{L}(v_h), \quad \forall\, v_h\in\mathcal{V}_h. \end{cases} \tag{7}$$

The RFB method consists in an unusual approximating space $\mathcal{V}_h$:

$$\mathcal{V}_h=\mathcal{V}_L\oplus B_h,$$

where B_h contains generic functions confined to the interior of the elements:

$$B_h=\bigoplus_{K\in\mathcal{T}_h} H_0^1(K).$$

Consequently, the solution u_h admits a unique decomposition into the sum of linear piecewise polynomial u_L and the bubble part u_b:

$$u_h=u_L+u_b, \quad u_L\in\mathcal{V}_L \text{ and } u_b=\sum_{K\in\mathcal{T}_h}\alpha_K b_K \ \in B_h, \tag{8}$$

where α_K is a constant and b_K denotes a bubble function. Note that u_L is the usual nodal interpolant of u_h (macro–scale), while u_b reflects the local behaviour (micro–scale), which can be neglected without a significant loss of accuracy. Using the splitting (8), and projecting (7) on the sub–spaces $\mathcal{V}_L$ and $\mathcal{B}_h$, we get:

$$\begin{cases} \mathcal{A}(u_L, v_L) + \mathcal{A}(u_b, v_L) = \mathcal{L}(v_L), \quad \forall\, v_L \in \mathcal{V}_L, \\ \\ \mathcal{A}(u_L, v_b) + \mathcal{A}(u_b, v_b) = \mathcal{L}(v_b), \quad \forall\, v_b \in \mathcal{B}_h. \end{cases} \tag{9}$$

As the divergence of the vector $\boldsymbol{\beta}$ is zero, considering the local character of the bubble, one can easily show that $(\alpha_K)_{K\in\mathcal{T}_h}$, the degrees of freedom associated with bubbles, may be eliminated in the equations of (9) giving rise to the following system containing only the linear part of the solution as unknown:

$$\mathcal{A}(u_L, v_L) + \sum_{K\in\mathcal{T}_h} \tau_K^{\mathrm{rfb}} \int_K (\boldsymbol{\beta}\cdot\nabla u_L - f)\,(\boldsymbol{\beta}\cdot\nabla v_L) = \mathcal{L}(v_L),\ \forall v_L \in \mathcal{V}_L. \tag{10}$$

Following the RFB approach, proposed by Brezzi–Russo in [8], the parameter which can be considered as *quasi-optimal* from the theoretical point of view is given by:

$$\tau_K^{\mathrm{rfb}} = \frac{1}{|K|}\int_K b_K,\ |K| = \mathrm{meas}(K), \tag{11}$$

where the optimal bubble b_K solves the following elliptic boundary-value problem:

$$\begin{cases} -\epsilon\Delta b_K + \boldsymbol{\beta}_K \cdot \nabla b_K = 1 \text{ in } K, \\ b_K = 0 \text{ on } \partial K. \end{cases} \tag{12}$$

Hence the problem of finding the optimum value for τ_K would be solved if we knew explicitly the exact solution of problem (12). However, the exact computation of this function is as difficult as the original problem itself, except in the one-dimension where the solution is known and its average over K is the coth-formula (5). In higher dimensions the computation of the residual–free bubbles becomes a major task. Therefore, the problem requires some suitable approximations, and there are some possibilities known to accomplish this. We refer to [8, 6, 7, 4, 13, 14, 15] for more details.

3.2.2. Novel RFB-Based Scheme

In [24], for the coarse-scale space consisting only of piecewise linear functions, it is shown that residual free bubbles type can be constructed to completely

recover the SUPG with a special choice of the streamline diffusion parameter. Knowing that, to compute the contribution of the RFB functions to the problem, we need to solve problem (12) in each element, separately. An efficient approach is introduced in [24] to compute the approximate solution of the bubble problem. Here we briefly describe this method (for more details, see [24]). Specify a "sub-mesh" $\mathcal{R}_h$ of each K that consists of non–overlapping parallelograms R^{e_i}, such that each element has one corner point $\mathbf{x}_i^-$ where $\mathbf{x}_i^- \in \partial K^-$, and one corresponding adjacent side being oriented in the streamline direction $(e_i) = (\mathbf{x}_i^-, \mathbf{x}_i^+), \mathbf{x}_i^+ \in \partial K^+$. Next, in each e_i of K let us define the solution b_{e_i} of the following one dimensional problem:

$$\begin{cases} \mathcal{L}_{e_i} b_{e_i} &= 1 & \text{in } e_i =]\mathbf{x}_i^-, \mathbf{x}_i^+[, \\ b_{e_i}(\mathbf{x}_i^-) &= b_{e_i}(\mathbf{x}_i^+) = 0, \end{cases} \tag{13}$$

where $\mathcal{L}_{e_i}$ is the convection–diffusion operator:

$$\mathcal{L}_{e_i} v = -\varepsilon (v)^{''} + \boldsymbol{\beta}_{e_i} (v)^{'},$$

$\boldsymbol{\beta}_{e_i}$ being the projection of $\boldsymbol{\beta}$ along e_i and $(\cdot)^{'}$ is the derivative along the edge e_i. The solution b_{e_i} is given in each e_i of K by:

$$b_i(\mathbf{x}) := \frac{|e_i|}{\| \boldsymbol{\beta} \|} \left(\varphi_i(\mathbf{x}) - \frac{\exp(p_{e_i} \varphi_i(\mathbf{x})) - 1}{\exp(p_{e_i}) - 1} \right), \tag{14}$$

for all $\mathbf{x} \in (\mathbf{x}_i^-, \mathbf{x}_i^+)$ where $\mathbf{x}_i^- \in \partial K^-$ and $\mathbf{x}_i^+ \in \partial K^+$ denote the inflow point and the outflow point respectively corresponding to $\mathbf{x}$ and where

$$\varphi_i(\mathbf{x}) := \frac{\|\mathbf{x} - \mathbf{x}_i^-\|}{|e_i|} \text{ and } p_{e_i} = \frac{|e_i| \| \boldsymbol{\beta} \|}{\epsilon}. \tag{15}$$

Using the relation between the local $\xi = (s,t)$ and the global coordinates system:

$$\begin{cases} s &= & (x - x_i^-) e_{(i,1)} + (y - y_i^-) e_{(i,2)}, \\ t &= - & (x - x_i^-) e_{(i,2)} + (y - y_i^-) e_{(i,1)}, \end{cases} \tag{16}$$

where $\mathbf{e}_i := (e_{(i,1)}, e_{(i,2)})^T$ ($\| \boldsymbol{\beta} \| = \boldsymbol{\beta} \cdot \mathbf{e}_i$) denotes the unit vector from $\mathbf{x}_i^- = (x_i^-, y_i^-)$ to $\mathbf{x}_i^+$, the equation (15) can be expressed in each R^{e_i} as $\varphi_i(s) = \dfrac{s}{|e_i|}$. Therefore, in each R^{e_i} the following definition of the bubble was suggested:

$$b_{R^{e_i}}(\xi) = \begin{cases} b_i(s - t/m_i) & \text{if } \xi \in R^{e_i}, \\ 0 & \text{elsewhere in } K. \end{cases} \tag{17}$$

Here $m_i = \tan(\alpha_i)$ is the slope of the straight line segments $(\partial R^{e_i} - (\partial R_1^{e_i\|} \cup \partial R_2^{e_i\|}))$, where $\partial R_1^{e_i\|}$ and $\partial R_2^{e_i\|}$ denote the edges of the mesh parallelogram R^{e_i} that are parallel to $\boldsymbol{\beta}$. Finally, in each K the bubble $b_{\mathcal{R}_h} = \sum_{R^{e_i} \in \mathcal{R}_h} b_{R^{e_i}}$ is introduced. It is worth pointing out that by construction, the proposed bubbles allow static condensation. After the bubbles have been eliminated, the Stabilized Method approximation gives the SUPG method for the convection-diffusion problem.

Let $h_{\beta,K}$ denote the longest line segment parallel to the convection vector $\boldsymbol{\beta}$ and contained in the element K. The expression of the parameter provided by the Stabilized Method approach is given by (see [24]):

$$\begin{aligned} \tau_K &= \frac{h_{\beta,K}}{3\|\boldsymbol{\beta}\|} + \frac{\epsilon}{\|\boldsymbol{\beta}\|^2}\left(\frac{2}{\mathcal{P}_{e_K}^2}\int_0^{\mathcal{P}_{e_K}} \frac{x^2}{\exp(x)-1}\,\mathrm{d}x - 1\right) \\ &= \frac{h_{\beta,K}}{3\|\boldsymbol{\beta}\|} + \frac{h_{\beta,K}}{\|\boldsymbol{\beta}\|}\left(2\mathcal{F}(\mathcal{P}_{e_K}) - \frac{1}{\mathcal{P}_{e_K}}\right). \end{aligned} \tag{18}$$

Here the Péclet number is defined as $\mathcal{P}_{e_K} = \frac{h_{\beta,K}\|\boldsymbol{\beta}\|}{\epsilon}$ and

$$\mathcal{F}(\mathcal{P}_{e_K}) = \int_0^1 \mathcal{G}(x, \mathcal{P}_{e_K})\,\mathrm{d}x \text{ with } \mathcal{G}(x, \mathcal{P}_{e_K}) = \begin{cases} \frac{x^2}{\exp(\mathcal{P}_{e_K} x)-1} & \text{if} \quad x \neq 0, \\ 0 & \text{else.} \end{cases} \tag{19}$$

As it is proved in [24], the streamline diffusion parameter satisfies:
- $\tau_K = O(h_K)$ for the convection-dominated case.
- $\tau_K = O(h_K^2)$ for the diffusion-dominated case.

This allows to define a doubly asymptotic approximation of τ_K by

$$\tau_K \sim \frac{h_{\beta,K}}{3\,\|\boldsymbol{\beta}\|} \min\left\{\frac{\mathcal{P}_{e_K}}{8}, 1\right\}. \tag{20}$$

The comparison of this value with the one used in the SUPG method is discussed for instance in [24].

4. A Posteriori Error Estimators and Indicators

The *a posteriori* estimators provide an estimate of the error between the computed discrete solution u_h of (7) and the unknown solution u of the continuous

problem (1) in a prescribed norm $\|\cdot\|$ using only informations which are available during the solution process, mainly the discrete solution itself and the data of the problem.

Several types of *a posteriori* error estimators are available in the literature. They may be classified into three families: the family of gradient indicator and gradient recovery based on the Zienkiewicz–Zhu work [28], the family of residual–based, and local–Neumann family. See John [18] for a numerical comprehensive discussion of these estimators. According to the numerical conclusions of [18], all these estimators have strengths and weaknesses.

A first step in the *a posteriori* estimation is to choose a suitable norm $\|\cdot\|$ in which the error should be estimated. The question of what are appropriate norms for error estimation for convection–diffusion equations is still under discussion within the scientific community. We chose to use in this work residual–based estimators with both H^1–semi norm and the L^2–norm, see remark 4.1. In fact, the implementation of such estimators is quite easy, and following [18], the residual-based estimator have a general good behavior.

We will first begin with general considerations on *a posteriori* estimates. In particular, we give some definitions of the estimators, efficiency index and the local indicators that are directly involved in the numerical studies and in adapting meshes.

Let η_K be the local error indicator on the mesh element K, see (24), and

$$\eta = \sqrt{\sum_{K\in\mathcal{T}_h} \eta_K^2}, \tag{21}$$

the global error indicator. The error estimation should satisfy the following global upper and a local lower estimates:

$$\begin{cases} \|u-u_h\| \leq C\eta,\ C>0, \\ \\ \eta_K \leq C_K \|u-u_h\|_{\mathcal{U}(K)},\ \ \mathcal{U}(K) \text{ is a neighbourhood of the element } K. \end{cases} \tag{22}$$

Summation of the second inequality of (22) over all mesh elements gives a lower estimate of the global error. Note that the first inequality of (22), gives information on the global error, and the discrete solution u_h is sufficiently accurate if $C\eta \leq \delta$, for a given tolerance δ.

The efficiency index I_{eff} is defined by:

$$I_{eff} = \frac{\eta}{\|u-u_h\|}. \tag{23}$$

Then an *a posteriori* error estimator is called efficient, if I_{eff} and I_{eff}^{-1} are bounded for all triangulations and an efficient *a posteriori* error estimator is called robust if I_{eff} and I_{eff}^{-1} are bounded independently of the coefficients of the problem. We refer to [18], for details on computations of I_{eff} for various *a posteriori* error estimators.

4.1. A Residual–Based Error Estimator in the H^1–Semi Norm and L^2–Norm

Let $\|\cdot\|_{0,\omega}$ be the $L^2(\omega)$–norm. The general form of residual–based *a posteriori* error estimators for convection–diffusion problems is

$$\eta_K^2 = \delta_K \|\mathcal{R}_h\|_{0,K}^2 + \sum_{E\in\partial K, E\notin\partial\Omega_N} \frac{\delta_E}{2} \|\sigma_h\|_{0,E}^2 + \sum_{E\in\partial K, E\in\partial\Omega_N} \delta_E \|\varepsilon\nabla u_h\cdot\mathbf{n}_E - g_{Nh}\|_{0,E}^2, \quad (24)$$

where $\mathcal{R}_h$ and σ_h denote, respectively, the residual and the jump of ∇u_h across the internal edges of the mesh:

$$\begin{cases} \mathcal{R}_h = f_h + \varepsilon\Delta u_h - \boldsymbol{\beta}_h\cdot\nabla u_h, \\ \sigma_h = [|\varepsilon\nabla u_h\cdot\mathbf{n}_E|]_E, \end{cases} \quad (25)$$

f_h, g_{Nh} and $\boldsymbol{\beta}_h$ are approximations of f, g and $\boldsymbol{\beta}$ respectively. $[|\cdot|]_E$ denotes the jump across an edge E. For a function $v_h\in\mathcal{V}_h$, this jump is defined for $\boldsymbol{x}\in E$ by:

$$[|v_h(\boldsymbol{x})|]_E = \begin{cases} \lim\limits_{t\longrightarrow 0^+} [v_h(\boldsymbol{x}+t\mathbf{n}_E) - v_h(\boldsymbol{x}-t\mathbf{n}_E)] & E\notin\partial\Omega, \\ \lim\limits_{t\longrightarrow 0^+} [-v_h(\boldsymbol{x}-t\mathbf{n}_E)] & E\in\partial\Omega, \end{cases}$$

where $\mathbf{n}_E$ is the outward unit normal on E. If E is the common edge to the elements K_1 and K_2 then, for appropriate choice of oriented normal $\mathbf{n}_E$, we have:

$$[|v_h|]_E = v_h|_{K_2} - v_h|_{K_1}.$$

Remark 4.1. The *a posteriori* error estimators for the H^1–semi norm and L^2–norm are, respectively, obtained by setting in (24):

$$\begin{cases} \delta_K = h_K^2 \text{ and } \delta_E = h_E \ (H^1\text{–semi norm}), \\ \\ \delta_K = h_K^4 \text{ and } \delta_E = h_E^3 \ (L^2\text{–norm}). \end{cases} \tag{26}$$

Remark 4.2. An advantage of the residual-based estimator is that it can be applied to more general spaces like L^p or $W^{1,p}$, $1 < p < \infty$. This is done, simply by applying the appropriate norms of these spaces in the expression of local indicator (24), see for example Verfürth [25, 26].

Let us mention that the estimator (24) is not robust in the convection-dominated regime. For robust estimators we refer to Verfürth [27], however, it is used for comparison purpose with different estimators for convection-diffusion problems in John [18]. Recently, John and Novo [19] have developed a robust SUPG a posteriori error estimator for convection-diffusion equations. In both [18] and [19], the stabilisation method used is SUPG. Our aim in this work is to show the superiority of the RFB method compared to the SUPG method. On other hand, we show that, at least with the estimator (24), the result can be improved (see the section below).

5. Numerical Results

We first choose a type of an estimator as well as a desired norm in which the error will be evaluated. In our studies, we applied the residual–based estimator with H^1–semi norm.

At a given level of refinement k, and after the computation of the error indicators, every mesh element K possesses an indicator $\eta_{K,k}$ computed from (24). With the help of these indicators, it must be decided which mesh elements should be refined or coarsened. In our study, refinement is applied to the elements such that:

$$\eta_{K,k} > 0.5 \times \max_{K \in \mathcal{T}_{h,k}} (\eta_{K,k}), \quad \mathcal{T}_{h,k} \text{ is the triangulation at the refinement level } k.$$

If the solution is known a priori, the adaptive procedure allows to assess the quality of the estimator, which in principle should only occur in areas with large gradients. Moreover, one can calculate the error at each level of refinement and compare it to the global indicator η given by (21). If not, the resulting solution

is satisfactory if it contains the desired qualities and particularly, the position of the boundary layers which are the main difficulties in numerical simulation of convection–diffusion problems.

Two types of numerical tests will be performed. The first one concerns the evaluation of the efficiency of the estimator. This is done by assuming that the exact solution u of the problem (1) is known and considering a sequence of meshes obtained by successive refinement.

The second concerns the automatic refinement of meshes. We always start with a coarse mesh, see Fig. 1, and we launch the automatic refinement and coarsening. The final mesh is obtained by deciding a stopping criterion. This usually covers the maximum number of items. In this case, local indicators are important.

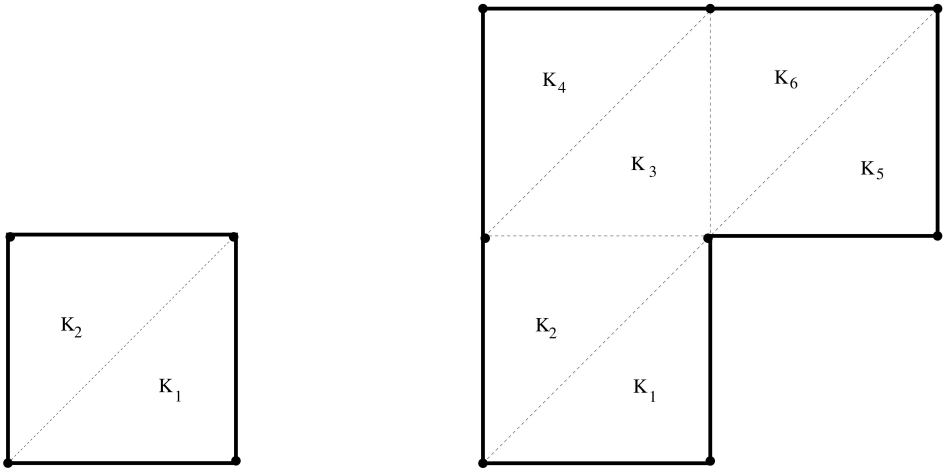

Figure 1: Initial coarse mesh: Grid1 and Grid2

As it was mentioned above, one has to precise some stopping criterion. In all examples discussed below, the computations were stopped after the first mesh on which the sum of degrees of freedom (d.o.f.) and Dirichlet nodes exceeded 10^5.

To address the many situations as possible, we considered the case of diffusion alone and the case of convection–diffusion. The first case allows, firstly to validate the calculations and error estimates. For the second case, we examined several types of inner and boundary layers as well as several kind of singularities.

5.1. Global Error Estimates

For both examples considered in this section, the solution is assumed to be known. Numerical tests will therefore allow to evaluate the numerical error and compare it to the theoretical error $\|u - u_h\|$ given by the a priori estimate. The solution to the problem of diffusion alone, with locally high gradients, allows to verify that the mesh adaptation procedure is working properly and it acts in the vicinity of areas of strong gradients. The solution of convection–diffusion problem depends on the diffusion parameter ε. This allows to see the behaviour of the numerical solution based on this parameter. Moreover, as it has boundary layers, we will then see that these layers are captured by the mesh that adapts well to these situations.

5.1.1. Diffusion Problem

We first consider the convection–diffusion problem (1) in a domain $\Omega =]0,1[^2$. The coefficients are

$$\varepsilon = 1, \quad \boldsymbol{\beta} = (0,0)^t,$$

and f is chosen such that the analytical solution is given by:

$$u(x,y) = e^{-100\left((x-0.5)^2+(y-0.5)^2\right)}. \tag{27}$$

The boundary data are:

$$\Gamma_D = \Gamma \quad \text{and} \quad g_D = u|_{\Gamma_D}.$$

This is obviously a diffusion problem whose solution u admits large gradients near the point $(1/2, 1/2)$. Indeed, the derivatives of the solution do not depend on the diffusion parameter ε which is not typical for solutions of convection–diffusion problems. Nevertheless, we can estimate the error in both L^2 and H^1 norms, and the behaviour of the adaptive mesh refinement, see next figures.

Starting from the initial coarse mesh Grid1, the computations were carried out on adaptive meshes which were generated using the indicators η_K given by (24).

As we can see on Fig. 2, the mesh refinement is done primarily in the vicinity of point where the gradients are large. Let N be the number of degrees of freedom, then obviously, N^{-1} asymptotically behaves as h^2. The errors correspond to the theoretical estimates which are of the order of $O(h^2)$ for the L^2–norm, and $O(h)$ for the H^1–semi norm.

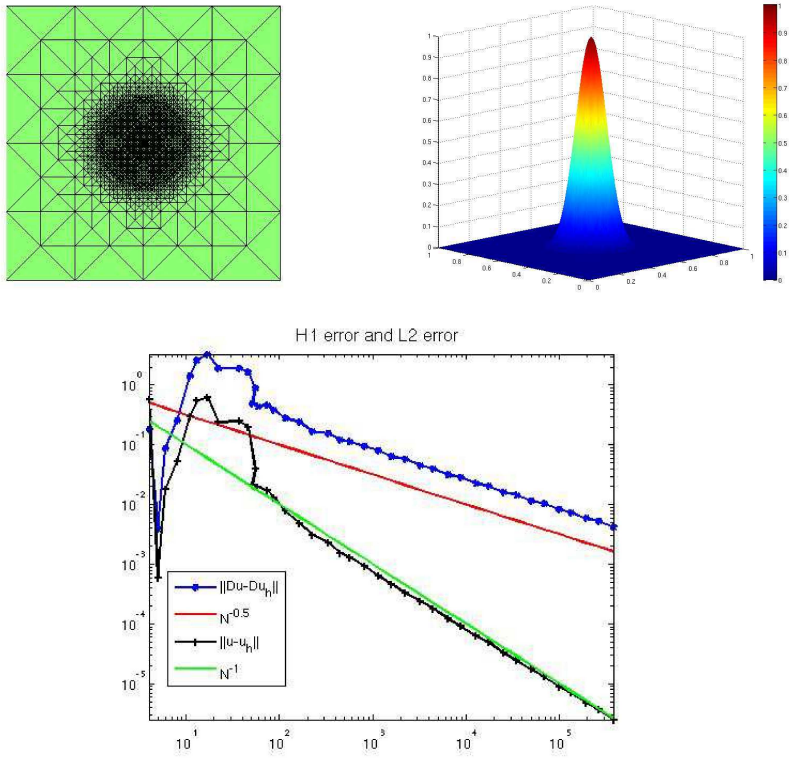

Figure 2: Diffusion problem: adapted mesh, numerical solution and convergence history

5.1.2. Convection–Diffusion Problem: Regular Boundary Layers

In the previous example, the derivatives of the solution do not depend on the diffusion parameter ε which is not typical for solutions of convection–diffusion problems. We consider now a convection–diffusion problem with regular boundary conditions, different values of the diffusion parameter ε and $\boldsymbol{\beta} = (2,3)^t$. The function f is chosen such that the analytical solution is given by:

$$u(x,y) = xy^2 - y^2 exp(\frac{2(x-1)}{\varepsilon}) - x exp(\frac{3(y-1)}{\varepsilon}) + exp(\frac{2(x-1)+3(y-1)}{\varepsilon}), \quad (28)$$

$\Omega =]0,1[^2$, the boundary data are $\Gamma_D = \Gamma$ and $g_D = u|_{\Gamma_D}$. As we can see on Fig. 3, the solution possesses typical regular layers at $x = 1$ and $y = 1$.

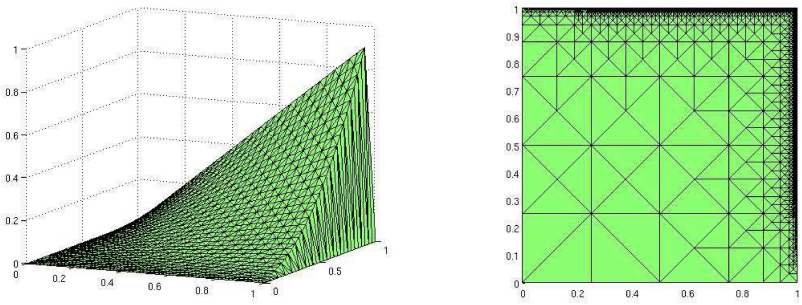

Figure 3: Exact solution and an adapted mesh for $\varepsilon = 10^{-4}$

The solution being dependent on the diffusion parameter, the numerical studies have been performed for three different values of this parameter:

$$\varepsilon = 10^{-2}, \quad \varepsilon = 10^{-4} \quad \text{and} \quad \varepsilon = 10^{-6}.$$

Starting from the initial coarse mesh Grid1, the computations were carried out on adaptive meshes which were generated using the indicators η_K given by (24). In Table 1, we present the obtained errors for various values of the parameter ε. As we can see in Table 1, to achieve the same error in L^2–norm, using the RFB method with the asymptotic parameter τ_{rfb}, we need only 65% of degrees of freedom required by the SUPG method with the asymptotic parameter τ_{supg} (for eps $= 0.01$), 43% for $\varepsilon = 10^{-4}$ and 83% for $\varepsilon = 10^{-6}$. Which clearly shows the superiority of the RFB method in these three situations. In dominant convection case, $\varepsilon = 10^{-6}$, it still requires less degrees of freedom. Furthermore, note that for about the same error, we obtained the same number of degrees of freedom with two parameters τ_{rfb} and $\tau_h = \dfrac{h_{\beta,K}}{3\,\|\beta\|}$. This is explained by the fact that τ_{rfb} goes to $\tau_{\text{asy}} = \dfrac{h_{\beta,K}}{3\,\|\beta\|}$ when ε goes to zero. Similarly, in the case of SUPG method, we obtain the same results for $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-6}$, with τ_{supg} and $\tau_h = \dfrac{h_K}{2\,\|\beta\|}$. Indeed, we know that τ_{supg} goes to $\tau_{\text{asy}} = \dfrac{h_K}{2\,\|\beta\|}$.

	$\varepsilon = 10^{-2}$			$\varepsilon = 10^{-4}$			$\varepsilon = 10^{-6}$		
τ_h	d.o.f.	$\|\cdot\|_{0,\Omega}$	$\|\cdot\|_{1,\Omega}$	d.o.f.	$\|\cdot\|_{0,\Omega}$	$\|\cdot\|_{1,\Omega}$	d.o.f.	$\|\cdot\|_{0,\Omega}$	$\|\cdot\|_{1,\Omega}$
$\frac{h_K}{2\|\beta\|}$	224234	0.0009	0.239	115186	0.002	26.43	225654	0.0017	142.1
$\frac{h_{\beta,K}}{3\|\beta\|}$	57339	0.0009	0.290	56011	0.0019	29.8	187051	0.0017	171.2
τ_{supg}	59694	0.0001	0.21	115186	0.002	26.43	225654	0.0017	142.1
τ_{rfb}	39021	0.0001	0.29	49167	0.002	32.54	187051	0.0017	171.2

Table 1: Comparison of the SUPG and RFB methods and their asymptotic behaviour

5.2. Adaptive Mesh Refinement

This section presents numerical experiments to study the behaviour of the a posteriori estimate of the error with respect to the mesh adaptation procedure. The examples studied here cover almost all types of singularities and layers (boundary or inner). In some examples, the solution does not have the required regularity in the a priori error analysis. However, this kind of test is interesting because generally we do not know the solution, even less its regularity.

Example 5.1 (Solution with parabolic and exponential boundary layers)**.** This example corresponds to the following situation:

$$\Omega =]0,1[^2,\ \varepsilon = 10^{-6}, \boldsymbol{\beta} = (1,0)^t,$$

$$f = 1 \ \text{ in } \ \Omega \ \text{ and } \ g_D = 0, \ \text{ on } \ \Gamma_D = \Gamma.$$

The computations were carried out on the initial mesh Grid1. The analytical solution is not known *a priori*, we compare the numerical results to the solution obtained on a very fine uniform mesh consisting of 263169 degrees of freedom.

As shown in Fig. 4, refinement is achieved in the areas of boundary layers.

Example 5.2 (Corner singularity, regular and inner boundary layers)**.** In this example, we consider the L-shaped geometry $\Omega =]0,1[^2 \backslash [0.5,1[\times]0,0.5]$.

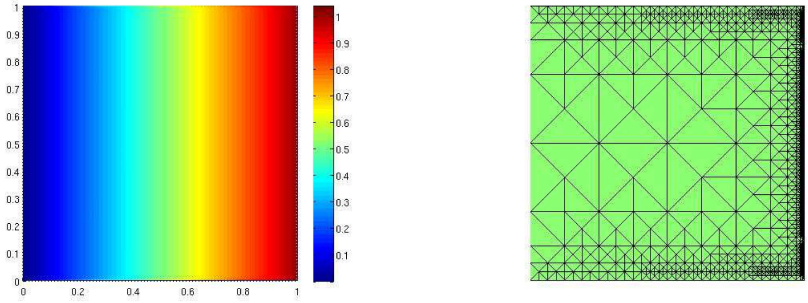

Figure 4: Example 5.1: the computed solution on an uniform mesh and an adapted mesh

The coefficients of the convection–diffusion problem (1) are:

$$\varepsilon = 10^{-6}, \quad \boldsymbol{\beta} = (1,3)^t,$$

$$f(x,y) = 100r\left(r - \tfrac{1}{2}\right)\left(r - \tfrac{\sqrt{2}}{2}\right), \quad r = \sqrt{(x-\tfrac{1}{2})^2 + (y-\tfrac{1}{2})^2}.$$

With such data, the exact solution of problem (1) is unknown but it presents regular boundary layers at:

$$\begin{cases} y = 1, \\ \{(0.5, y); \quad 0 < y \leq 0.5\}, \quad \text{and at} \\ \{(1, y); \quad 0.5 < y \leq 1\}. \end{cases}$$

It presents also an inner layer in the direction of the convection vector $\boldsymbol{\beta}$ starting from the singular point $(0.5; 0.5)$, see Fig. 5. Moreover, because of the re–entrant corner, the domain is not convex, then this solution possesses a singularity corner at $(0.5, 0.5)$, $u \notin H^2(\Omega)$. We are thus confronted with a singular case where we can not apply the theoretical results giving the *a priori* estimates of the error.

Grid2 was used as the initial mesh. As for the previous example, since we do not know the analytical solution of this problem, we compare the behaviour of the error estimators with graphical representation of the computed solution on a very fine uniform mesh. As we can see in Fig. 5, the error estimator produces meshes which are well refined within all layers.

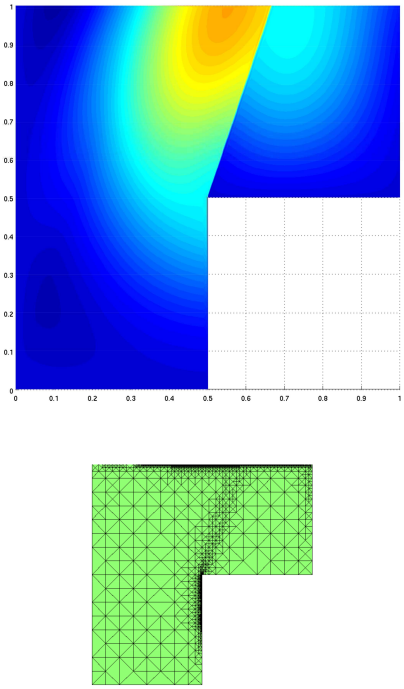

Figure 5: Example 5.2: Computed solution on a very fine uniform mesh and an adapted mesh

Example 5.3 (Discontinuous boundary conditions and inner layers)**.** This example corresponds to the following situation:

$$\Omega = \left]0,1\right[^2,\ \varepsilon = 10^{-6},\ \boldsymbol{\beta} = (1,2)^t,\ \text{and}\ f = 0 \text{ in } \Omega.$$

The boundary conditions are given by:

$$g_D = \begin{cases} 0, & \text{on } \Gamma_1 = \{(x,0);\ 0.25 \le x \le 1\} \cup \{(0,y);\ 0.25 \le y \le 1\}, \\ 1, & \text{on } \Gamma_2 = \{(x,0);\ 0 \le x < 0.25\} \cup \{(0,y);\ 0 \le y < 0.25\}, \end{cases}$$

and $\varepsilon \nabla u \cdot \mathbf{n} = 0$ on $\Gamma - (\Gamma_1 \cup \Gamma_2)$.

For a graphical representation of the solution see Fig. 6. Due to the jump in the boundary conditions, the solution does not have the required regularity.

Once again, we find ourselves in a situation where we can not guarantee sufficient regularity to apply the theoretical results. However, this test is of great interest. The solution is obviously not known but we know that it represents the transport, in the whole domain, of the boundary condition g_D on the inlet part of the boundary. Of course, this solution allows internal boundary layers along the vector of convection. This aspect of the solution is well produced by the mesh adaptation procedure. We can see from Fig. 6, that the mesh refinement is essentially performed along the lines:

$$y = 2x + 0.25 \quad \text{and} \quad y = 2x - 0.5.$$

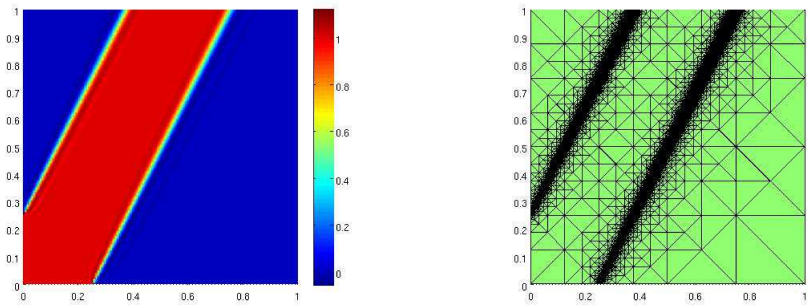

Figure 6: Example 5.3: Numerical solution and an adapted mesh

6. Conclusion

In this work, we showed that the combined use of stabilization techniques and adaptive mesh procedures can efficiently solve convection–diffusion problems and allows to obtain accurate solutions. Moreover, the use of mesh adaptation may significantly reduce the number of degrees of freedom and thus enable a fast and efficient computing.

Several numerical experiments are reported. The examples considered in this work cover almost all types of singularities and layers (boundary or inner). Finally, this approach could be applied to all kind of equations containing transport terms. Certainly, the numerical results can be improved using a robust a posteriori estimator. This work is a first step towards treating more complex problems such as linearized Navier–Stokes equations (Oseen problem), Navier–Stokes equations or the problem of viscoelastic fluid flows, etc.

References

[1] M. Ainsworth, J.T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, John Wiley, 2000.

[2] C. Baiocchi, F. Brezzi, L.P. Franca, Virtual bubbles and Galerkin-least-squares type methods (Ga.L.S.), *Comput. Methods Appl. Mech. Engrg.*, **105** (1993), 125–141.

[3] F. Brezzi, M. O. Bristeau, L.P. Franca, M. Mallet, G. Rogé, A relationship between stabilized finite element methods and the Galerkin method with bubble functions, *Comput. Methods Appl. Mech. Engrg.*, **96** (1992), 117–129.

[4] F. Brezzi, P. Houston, L.D. Marini, E. Süli, Modeling subgrid viscosity for advection-diffusion problems, *Comput. Methods Appl. Mech. Engrg.*, **166** (2000), 1601–1610.

[5] F. Brezzi, T.J.R. Hughes, L.D. Marini, A. Russo, E. Süli, A priori error analysis of residual-free bubbles for advection-diffusion problems, *SIAM J. Numer. Anal.*, **36** (1999), 1933–1948.

[6] F. Brezzi, L.D. Marini, A. Russo, Applications of the pseudo residual-free bubbles to the stabilization of convection-diffusion problems, *Comput. Methods Appl. Mech. Engrg.*, **166** (1998), 51–63.

[7] F. Brezzi, L.D. Marini, A. Russo, On the choice of a stabilizing subgrid for convection-diffusion problems, *Comput. Methods Appl. Mech. Engrg.*, **194** (2005), 127–148.

[8] F. Brezzi, A. Russo, Choosing bubbles for advection-diffusion problems, *Math. Models Methods. Appl. Sci.*, **4** (1994), 571–587.

[9] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equation, *Comput. Methods Appl. Mech. Engrg.*, **32** (1982), 199-259.

[10] I. Christie, D.F. Griffiths, A.R. Mitchell, O.C. Zienkiewicz, Finite element methods for second order differential equations with significant first derivatives, *Int. J. Numer. Methods Engrg.*, **10** (1976), 1389–1396.

[11] L.P. Franca, G. Hauke, A. Masud, Revisiting stabilized finite element methods for the advective–diffusive equation, *Comput. Methods Appl. Mech. Engrg.*, **195** (2006), 1560–1572.

[12] L.P. Franca, A. Russo, Deriving upwinding, mass lumping and selective reduced integration by residual-free bubbles, *Appl. Math. Lett.*, **9** (1996), 83–88.

[13] J.-L. Guermond, Stabilisation par viscosité de sous-maille pour l'approximation de Galerkin des opérateurs linéaires monotones, (In French) [Subgrid viscosity stabilization of Galerkin approximations of monotone linear operators], *C. R. Acad. Sci. Paris, Sér. I Math.*, **328** (1999), 617–622.

[14] J.-L. Guermond, Subgrid stabilization of Galerkin approximations of linear monotone operators, *IMA J. Numer. Anal.*, **21** (2001), 165–197.

[15] J.-L.Guermond, Subgrid stabilization of Galerkin approximations of linear contraction semi-groups of class C^0 in Hilbert spaces, *Numer. Methods Partial Differential Equations*, **17** (2001), 1–25.

[16] T.J.R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, *Comput. Methods Appl. Mech. Engrg.*, **127** (1995), 387–401.

[17] T.J.R. Hughes, G.R. Feijoo, L. Mazzei, J.B. Quincy, The variational multiscale method: a paradigm for computational mechanics, *Comput. Methods Appl. Mech. Engrg.*, **166** (1998), 3–24.

[18] V. John, A numerical study of *a posteriori* error estimators for convection–diffusion equations, *Comput. Methods Appl. Mech. Engrg.*, **190** (2000), 757–781.

[19] V. John, J. Novo, A robust SUPG norm *a posteriori* error estimator for stationary convection–diffusion equations, *Comput. Methods Appl. Mech. Engrg.*, **225** (2013), 289–305.

[20] A. Quarteroni, A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer, Berlin - Heidelberg - New York, 1994.

[21] S.I. Repin, *A Posteriori Estimates for Partial Differential Equations*, Radon Ser. on Computational and Applied Mathematics, 2008.

[22] H.G. Roos, M. Stynes, L. Tobiska, *Robust Numerical Methods for Singularly Perturbed Differential Equations*, Springer, Berlin, 2008.

[23] A. Russo, Bubble stabilization of finite element methods for the linearized incompressible Navier-Stokes equations, *Comput. Methods Appl. Mech. Engrg.*, **132** (1996), 335–343.

[24] A. Serghini Mounim, A stabilized finite element method for convection–diffusion problems, *Numer. Methods Partial Differential Eq.*, **28** (2012), 1916-1943.

[25] R. Verfürth, *A Review of a Posteriori Error Estimation and Adaptive Mesh–Refinement Techniques*, Wiley and Tubner, 1996.

[26] R. Verfürth, A posteriori error estimators for convection–diffusion equations, *Numer. Math.*, **80** (1998), 641–663.

[27] R. Verfürth, Robust a posteriori error estimates for stationary convection-diffusion equations, *SIAM J. Numer. Anal.*, **43** (2005), 1766–1782.

[28] O.C. Zienkiewicz, J.Z. Zhu, A simple error estimator and adaptive procedure for practical engineering analysis, *Internat. J. Numer. Meth. Engrg.*, **24** (1987), 337–357.