Variational Multiscale error estimator for anisotropic adaptive fluid mechanic simulations: application to convection-diffusion problems

Alban Bazile, Elie Hachem, Juan-Carlos Larroya-Huguet, Youssef Mesri

Variational Multiscale error estimator for anisotropic adaptive fluid mechanic simulations: application to convection-diffusion problems

A. Bazile[1,2], E. Hachem[1], J.C. Larroya-Huguet[2], Y. Mesri[1]

[1] *MINES ParisTech, PSL - Research University, CEMEF - Centre for material forming, CNRS UMR 7635, CS 10207 rue Claude Daunesse, 06904 Sophia-Antipolis Cedex, France,*

[2] *Safran Aircraft Engines, Site de Villaroche Rond-Point Rene Ravaud-Reau, 77550 Moissy Cramayel, France,*

Abstract

In this work, we present a new a posteriori error estimator based on the Variational Multiscale method for anisotropic adaptive fluid mechanics problems. The general idea is to combine the large scale error based on the solved part of the solution with the sub-mesh scale error based on the unresolved part of the solution. We compute the latter with two different methods: one using the stabilizing parameters and the other using bubble functions. We propose two different metric tensors $\mathcal{H}_{iso}$ and $\mathcal{H}^{new}_{aniso}$. They are both defined by the recovered Hessian matrix of the solution and rely on the sub-grid scale error estimator. Thus, we develop a new anisotropic local error indicator and we test it for mesh adaptation on convection-dominated benchmarks in 2D and 3D. The results show that the proposed error indicator leads to enhanced and accurate solutions while using a drastically reduced number of elements.

Keywords: CFD; VMS; Error Estimator; Mesh Adaptation; Convection-Diffusion.

1. Introduction

The use of Computational Fluid Dynamics (CFD) for industrial applications has been in constant increase for the last decades. Researchers are

[1] *Email address:* alban.bazile@mines-paristech.fr

continuously developing new techniques to reach higher level of precision. Nevertheless, to comply with industrial expectations, a trade-off has to be found between high precision results and high computational costs [1]. Different strategies can be found in the literature. Most of them are related to high order elements (see [2, 3, 4]), parallel computing (see [5, 6, 7, 8]) or, in particular, adaptive methods (see [9, 10, 11, 12]).

Indeed, adaptive methods make it possible to improve the accuracy and the efficiency of numerical methods. In particular, anisotropic mesh adaptation has proved to be powerful in capturing dynamically the heterogeneities that can appear in numerous physical applications including those having boundary or inner layers [13, 14]. In these cases, gradients of the solution are highly directional and can be captured with a good accuracy using fewer additional elements. These mesh adaptation techniques are based on local modifications of an existing mesh. Usually, it consists in a local stretching of the elements which is defined by a metric field. This metric field is built from an error analysis on the mesh. In [15, 16, 17] for example, the error analysis is done on the edges of the elements.

In fact, theories of anisotropic error estimation have been well developed, leading to some standardization of the adaptation process. Error estimation of the discretization error and in particular, of the interpolation error have been performed in a number of works such as in [18, 19, 20]. From the interpolation error analysis, several recent results [21, 22, 20, 23, 24] have brought renewed focus on metric-based mesh adaptation where the underling metric is derived from a recovered Hessian. Indeed, the Hessian based metric mesh adaptation has several advantages from which we note: (i) the general computation framework, (ii) the relatively easy way of implementation and above all, (iii) the robustness.

Despite the practical construction of this kind of anisotropic error estimators, the information derived from them is only an indicator for the mesh adaptation mechanics. Indeed, the fine scale features related to some dynamic flow solutions are still difficult to capture without considering the PDE-dependent approximation error. The estimation of the latter involves the development of multiscale a posteriori error estimators that can be computationally prohibitive.

However, the Variational Multiscale method (VMS) that we use to stabilize our continuous finite element scheme provides, by construction, a cost free PDE-dependent a posteriori error estimator. The VMS approach introduced in [25, 26, 27, 28] consists in the splitting of the solution into a resolved part (i.e. coarse scales) and an unresolved part (i.e. subscales). The resolution of unresolved part gives a direct access to the sub-mesh scale information of the solution and allows us to compute an approximation error estimator without solving any additional equation. Recently, several works related to variational multiscale (or subscales) error estimators in the VMS framework have been published. Granzow et al. in [29] developed an error representation for output quantities based on a dual enrichment technique. Also, in [30], Baiges et al. proposed a general error estimator for the finite element solution of solid mechanics problems. John et al. proposed in [31] a robust residual-based a posteriori estimator for the SUPG finite element method applied, in particular, to stationary convection-diffusion-reaction equations.

In this work, we compute the subscales error estimator using two different methods. The first one uses stabilizing parameters derived from the VMS analysis and local norms defined on the elements. It has been developed by Hauke et al. in [32, 33] and provides an *element-wise* computation of the error . Referring to Irisarri et al., the second method uses a linear combination of bubble functions to establish a *pointwise* computation of the error. It has been developed for one-dimensional transport equation in [34] and for 2D transport equation in [35].

The first contribution of this paper is then to propose a new isotropic mesh adaptation technique based on the previous subscales error estimator. To do so, we define a new isotropic metric tensor $\mathcal{H}_{iso}$ and we solve an optimization problem under the constraint of a fixed number of elements. Until now, the variational multiscale error estimators were only derived for isotropic mesh adaptation. In our work, we also propose to use the sub-mesh scale information for anisotropic mesh adaptation.

The second contribution is therefore to combine both the coarse scales interpolation error indicator and the subscales error estimator for anisotropic mesh adaptation. To the best of our knowledge, such strategy has never been tempted. To do so, we derive a new anisotropic metric tensor $\mathcal{H}_{aniso}^{new}$ that allows to take into account the anisotropic variations of the solution on the

mesh and also relies upon the sub-grid information of the solution. This new anisotropic mesh adaptation technique takes into account the finite element VMS framework used for the numerical resolution of the convection-diffusion equation. In fact, the VMS approach allows to quantify the a posteriori subscales errors that can be used as weight to enrich the anisotropic error indicator of the conventional Hessian based approach. Unlike existing anisotropic a posteriori error estimators like [36, 37, 38] to mention a few, this hybrid approach takes advantages from both a priori and a posteriori methods. Indeed, we build an isotropic a posteriori error estimator targeting the subscales and we combine it with an anisotropic a priori interpolation based error indicator targeting the coarse scales.

The paper is structured as follows. In Section 2, we introduce the Variational Multi-Scale framework. From this mathematical background, we present, in Section 3, the two different computation methods of the subscales error estimator. Then, mesh adaptation is presented in Section 4. As a reminder, we introduce the Hessian based anisotropic mesh adaptation in Section 4.1. Then, we define the isotropic metric tensor $\mathcal{H}_{iso}$ built from the subscales error estimator in Section 4.2. Finally, we present the new combination of the interpolation error indicator and the subscales error estimator in Section 4.3. This combination is tested on 2D and 3D convection-diffusion benchmarks in Section 5. In this last section, an error analysis of the exact error is made to highlight the two contributions of the paper.

2. Convection-diffusion equation in the Variational Multi-Scale framework

To begin this paper, let's introduce some notations. Referring to Irissarri et al. in [35], we express the partial differential equation as:

$$\begin{cases} \mathcal{L}u = f & in\ \Omega \\ u = g & on\ \Gamma_g \\ \mathcal{B}u = h & on\ \Gamma_h \end{cases} \tag{1}$$

where $\mathcal{B}$ is an operator acting on the boundary that defines the natural boundary condition; and $\mathcal{L}$ is a generic differential operator. g is the value of the Dirichlet boundary condition and h determines the Neumann boundary condition. In this work, we consider only Dirichlet boundary conditions. For

the case of a convection-diffusion equation, we have:

$$\mathcal{L}u = -a\Delta u + \mathbf{v} \cdot \nabla u \tag{2}$$

where a is the diffusion coefficient and $\mathbf{v}$ the velocity field.

We define the solution and test function spaces as standard Sobolev spaces:

$$\begin{array}{l} \mathcal{S} = \{u \ \in \ H^1(\Omega) \mid u = g \ on \ \Gamma_g\} \\ \mathcal{V} = \{w \ \in \ H^1(\Omega) \mid w = 0 \ on \ \Gamma_g\} \end{array} \tag{3}$$

The variational formulation of Eq. 1 is:

$$\left\{ \begin{array}{l} \text{Find } u \in \mathcal{S} \text{ such that:} \\ \\ a(w,u) = (w,f), \quad \forall w \in \mathcal{V} \end{array} \right. \tag{4}$$

where $a(.,.)$ is a bilinear form, $(.,.)$ the $L^2(\Omega)$ inner product.

2.1. The Galerkin finite element formulation

Applying the Finite Element Method (FEM), we mesh the domain into n_{el} non-overlapping elements Ω_e. We write $\tilde{\Omega}$ and $\tilde{\Gamma}$ as:

$$\tilde{\Omega} = \bigcup_{e=1}^{n_{el}} \Omega_e \qquad \tilde{\Gamma} = \bigcup_{e=1}^{n_{el}} \Gamma_e \setminus \Gamma \tag{5}$$

We define the Finite Element solution and weighting function spaces $\mathcal{S}^h \subset \mathcal{S}$ and $\mathcal{V}^h \subset \mathcal{V}$ such that:

$$\begin{array}{l} \mathcal{S}^h = \{u_h \in H^1(\Omega) \mid u_{h|\Omega_e} \in \mathbb{P}_k, \ u_{h|\Gamma_g} = g \ \forall \Omega_e \in \tilde{\Omega}\} \\ \mathcal{V}^h = \{w_h \in H^1(\Omega) \mid w_{h|\Omega_e} \in \mathbb{P}_k, \ w_{h|\Gamma_g} = 0 \ \forall \Omega_e \in \tilde{\Omega}\} \end{array} \tag{6}$$

with $\mathbb{P}_k$ the space of polynomial of degree k. In this paper, we will only use linear elements, hence $k = 1$.

With these definitions, we apply the standard Galerkin method:

$$\left\{ \begin{array}{l} \text{Find } u_h \in \mathcal{S}^h \text{ such that,} \\ \\ a(w_h,u_h) = (w_h,f), \quad \forall w_h \in \mathcal{V}^h \end{array} \right. \tag{7}$$

It is well known that this formulation is unstable and leads to spurious oscillations when the convective term of the equation is dominant. For this reason, we stabilize the formulation using the Streamline Upwind Petrov-Galerkin (SUPG) numerical scheme.

2.2. Streamline Upwind Petrov-Galerkin (SUPG) method

The Streamline Upwind Petrov-Galerkin (SUPG) method consists in introducing stabilizing terms in the standard Galerkin formulation. To do so, for all terms in Eq. (7), we replace the weighting function w_h by a new weighting function $w_h + \tau_e \mathbf{v} \cdot \nabla w_h$. This modification of the formulation is usually interpreted as adding more weight to the node upstream, reducing the weight on the node downstream. It adds an artificial weighted diffusion along the streamline direction. This stabilization is done locally and on each element we have:

$$a(w_h + \tau_e \mathbf{v} \cdot \nabla w_h, u_h) = (w_h + \tau_e \mathbf{v} \cdot \nabla w_h, f) \quad \forall w_h \in \mathcal{V}^h \tag{8}$$

Thanks to the bi-linearity of $a(.,.)$ and of the $L^2(\Omega)$ inner product, the formulation can be written as:

$$a(w_h, u_h) + a(\tau_e \mathbf{v} \cdot \nabla w_h, u_h) - (\tau_e \mathbf{v} \cdot \nabla w_h, f) = (w_h, f) \quad \forall w_h \in \mathcal{V}^h \tag{9}$$

And so,

$$a(w_h, u_h) + (\tau_e \mathbf{v} \cdot \nabla w_h, \mathcal{L}u_h - f) = (w_h, f) \quad \forall w_h \in \mathcal{V}^h \tag{10}$$

Finally, summing on each element, we have a new bi-linear form $a_\tau(.,.)$ and we can write the following formulation:

$$a_\tau(w_h, u_h) = a(w_h, u_h) + \sum_{\Omega_e \in \tilde{\Omega}} (\mathbf{v} \cdot \nabla w_h, \tau_e(\mathcal{L}u_h - f))_{\Omega_e} = (w_h, f) \quad \forall w_h \in \mathcal{V}^h \tag{11}$$

Concerning the choice of the stabilizing parameter τ_e, we refer to Hughes et al. in [39] and we choose:

$$\tau_e = \frac{h_e}{2||\mathbf{v}||_2} \left(coth(Pe_{\Omega_e}) - \frac{1}{Pe_{\Omega_e}} \right) \tag{12}$$

where h_e is the local mesh size, computed as described in [16]. Pe_{Ω_e} is the local Peclet number defined as:

$$Pe_{\Omega_e} = \frac{||\mathbf{v}||_2 h_e}{2a} \tag{13}$$

Remark 1. *In order to study the error of this numerical scheme, we use the Variational Multiscale analysis. In fact, it has to be noted that the SUPG scheme can be considered as a particular form of the generalized VMS formulation. Indeed, the stabilizing term of Eq. (11) can also be seen as the effect of the subscales on the coarse scales. Hughes gives more details about this concordance in [28].*

2.3. The Variational Multi-Scale formulation

The VMS formulation consists in decomposing the solution and test functions spaces into two sub-spaces: a mesh scale subspace (or coarse scales) $(\mathcal{S}^h, \mathcal{V}^h)$ and an under-mesh scale subspace (or subscales) $(\mathcal{S}', \mathcal{V}')$ such that $\mathcal{S} = \mathcal{S}^h \oplus \mathcal{S}'$ and $\mathcal{V} = \mathcal{V}^h \oplus \mathcal{V}'$. Therefore, we can decompose the solution and test functions as follow:

$$\begin{array}{l} u = u_h + u', \quad u_h \in \mathcal{S}_h, \quad u' \in \mathcal{S}' \\ w = w_h + w', \quad w_h \in \mathcal{W}_h, \quad w' \in \mathcal{S}' \end{array} \tag{14}$$

Thanks to the orthogonality between the coarse scales subspace and the subscales subspace, the variational form can be split into a coarse scales sub-problem and a subscales sub-problem [39]:

$$\begin{array}{l} a(w_h, u_h) + a(w_h, u') = (w_h, f) \quad \forall w_h \in \mathcal{V}^h \\ a(w', u_h) + a(w', u') = (w', f) \quad \forall w' \in \mathcal{V}' \end{array} \tag{15}$$

We start by solving the subscales sub-problem (second equation). For smooth functions on the element interior but rough across the inter-element boundaries, the integration by parts leads to the following equation:

$$\begin{array}{l} a(w', u') = -a(w', u_h) + (w', f) \\ a(w', u') = -(w', \mathcal{L}u_h - f) - (w', [\mathcal{B}u_h])_{\tilde{\Gamma}} - (w', \mathcal{B}u_h)_{\Gamma_h} \quad \forall w' \in \mathcal{V}' \end{array} \tag{16}$$

where the jump term [.] represents the difference of the fluxes on both sides of the element boundaries (see [40] for more details). An analytic solution of problem (16) can be found. This solution will be developed in the next subsection.

3. A posteriori error estimation on solution's subscales

In fact, the VMS analysis gives access to a solution of the subscales u'. Considering the relation $u = u_h + u'$, this part of the solution can also be

considered as the *subscales error*. However, u' is not explicitly computed during the simulation. To obtain it, we use an a posteriori computation.

In this paper, we use two different methods for the a posteriori computation of the subscales error estimator. Both methods are residual based and rely on convection-dominated regime assumptions:

- The first method consists in computing the error estimator as the multiplication of the stabilizing parameter's norm and the residual's norm. The computation is done *element-wise* because of the local definition of the stabilizing parameter and the local error norms used.
- The second method does not use the stabilizing parameter. The error estimator is computed thanks to the *pointwise* error estimation of Irisarri et al. in [35]. It uses a set of bubble functions as a substitution of the subscales Green's functions.

In this section, we give the key features of these two computation methods.

3.1. Computation of the error estimator with stabilizing parameters

Going back to Section 2.3, we had the following sub-problem for the subscales:

$$a(w', u') = -(w', \mathcal{L}u_h - f) - (w', [\mathcal{B}u_h])_{\tilde{\Gamma}} - (w', \mathcal{B}u_h)_{\Gamma_h} \quad \forall w' \in \mathcal{V}'$$

According to [39], this problem can be solved analytically using Green's function $g'(\mathbf{x}, \mathbf{y})$. We have the following paradigm:

$$u'(\mathbf{x}) = -\int_{\tilde{\Omega}_y} g'(\mathbf{x}, \mathbf{y})(\mathcal{L}u_h - f)(\mathbf{y}) d\Omega_y - \int_{\tilde{\Gamma}_y} g'(\mathbf{x}, \mathbf{y})([\mathcal{B}u_h])(\mathbf{y}) d\Gamma_y - \int_{\Gamma_{hy}} g'(\mathbf{x}, \mathbf{y})(\mathcal{B}u_h)(\mathbf{y}) d\Gamma_y \tag{17}$$

This error estimator includes:

- Interior residuals, $\mathcal{L}u_h - f$ in $\tilde{\Omega}$
- Inter-elements residuals, $[\mathcal{B}u_h]$ on $\tilde{\Gamma}$
- Natural boundary condition residual $\mathcal{B}u_h$ on Γ_h

From here, multiple cases can be considered. In this work, we use linear shape functions with a piecewise constant source term f. Therefore, we can make the assumption that the residual $\mathcal{L}u_h - f$ is P0. Furthermore, we only have Dirichlet boundary conditions. Thus, we neglect the effects of the boundary condition residual. Finally, as for all convection-dominated regimes, we will first consider the smooth case, i.e. that the discontinuity of the subscales is not preponderant and that the inter-elements residuals can be neglected. With these assumptions, we write:

$$u'(\mathbf{x}) \approx -\int_{\tilde{\Omega}_\mathbf{y}} g'(\mathbf{x},\mathbf{y})(\mathcal{L}u_h - f)(\mathbf{y})d\Omega_y \tag{18}$$

We will also make the assumption that the error has a local influence. Thus, we can replace the subscales Green's function by a local Green's function that vanishes at the element boundary, i.e., $g^e(\mathbf{x},\mathbf{y}) = 0$ on Γ^e. We can write:

$$g'(\mathbf{x},\mathbf{y}) \approx g^e(\mathbf{x},\mathbf{y}) \approx \tau_e \delta(\mathbf{y}-\mathbf{x}) \tag{19}$$

with,

$$\tau_e = \frac{1}{meas(\Omega_e)} \int_{\Omega_{e,\mathbf{x}}} \int_{\Omega_{e,\mathbf{y}}} g^e(\mathbf{x},\mathbf{y}) d\Omega_{e,\mathbf{x}} d\Omega_{e,\mathbf{y}} \tag{20}$$

In [40], the author shows that the above assumptions work well for convection-dominated regimes. In fact, numerical experiments show that for high Peclet number flows and stabilized solutions, the contribution of jump terms is negligible.

From here, we can therefore write a local error estimator like the following:

$$u'(\mathbf{x})|_{\Omega_e} \approx -\int_{\Omega_e} g^e(\mathbf{x},\mathbf{y})(\mathcal{L}u_h - f)(\mathbf{y})d\Omega_y = -\int_{\Omega_e} \tau_e \delta(\mathbf{y}-\mathbf{x})(\mathcal{L}u_h - f)(\mathbf{y})d\Omega_y = -\tau_e(\mathcal{L}u_h - f)(\mathbf{x}) \tag{21}$$

Taking the L^2 norm, we can finally write the following local error estimator for the solution subscales:

$$\boxed{||u'(\mathbf{x})||_{L^2,\Omega_e} = |\tau_e| \; ||\mathcal{L}u_h - f||_{L^2,\Omega_e}} \tag{22}$$

We know the expression of τ_e from the application of the SUPG method and we can *a posteriori* compute the P0 residual in our finite element computation domain. As said in the introduction, the sub-scale information computed here is *element-wise* and can be used as such in mesh adaptation.

3.2. Computation of the error estimator with high order bubbles functions

In this section, we use the work of Irisarri et al. in [35]. This time, we apply a *pointwise* computation of the error estimator. This computation method consists in decomposing the error into two components according to the nature of the residuals:

$$u'(\mathbf{x}) = u'_{bub}(\mathbf{x}) + u'_{poll}(\mathbf{x}). \tag{23}$$

The first term u'_{bub} is the internal residual error and it is related to the local internal residual, $f - \mathcal{L}u_h$, inside the elements. As we will see later, this part of the error is modeled locally thanks to a set of bubble functions. The second term u'_{poll} is the inter-element error. It represents the pollution error due to sources of errors outside the element. As said in the previous section, it is negligible when considering convection-dominated regime. Consequently, in this paper we will consider only the internal residual error and suppose that:

$$u'(\mathbf{x}) \approx u'_{bub}(\mathbf{x}). \tag{24}$$

3.2.1. Practical aspects of computation

As proposed by Irisarri et al. in [35], the way of obtaining a numerical solution of u'_{bub} is by solving the following discrete problem:

$$\begin{cases} \text{Find } u'_{bub} \in \mathcal{S}^h_{bub} \text{ such that} \\ a(w'_{bub}, u'_{bub}) = (w'_{bub}, f - \mathcal{L}u_h), \quad \forall w'_{bub} \in \mathcal{S}^h_{bub} \end{cases} \tag{25}$$

This time, the error component is expressed with a combination of bubble functions:

$$u'_{bub}(\mathbf{x}) = \sum_{i=1}^{n_{bub}} c^b_i b_i(\mathbf{x}) \tag{26}$$

Considering bubbles functions of order 3, we have:

$$u'_{bub}(\mathbf{x}) = c^b_1 b_1(\mathbf{x}) + c^b_2 b_2(\mathbf{x}) + c^b_3 b_3(\mathbf{x}) \tag{27}$$

with c^b_i unknown constant to be determined.

Referring to [41], the definition of the first bubble function $b_1(\mathbf{x})$ is the following:

$$b_1(\mathbf{x}) = (d+1)^{d+1} \prod_{i=1}^{d+1} \hat{\lambda}_i \tag{28}$$

where d is the dimension of the problem and $\hat{\lambda}_i$ are the barycentric coordinates in the reference element.

The next bubble functions $b_2(\mathbf{x})$ and $b_3(\mathbf{x})$ are built by adding the monomials of the Pascal triangle with center in the barycenter $c_e = (\xi_e, \eta_e)$ of the element. For example, in 2D, in the reference element: $\Omega_{ref} = \{(\xi, \eta) : 0 \leq \xi \leq 1; 0 \leq \eta \leq 1 - \xi\}$, we choose the following bubble functions:

$$\begin{array}{l} b_1(\xi, \eta) = 27 \times \xi\eta(1 - \xi - \eta) \\ b_2(\xi, \eta) = 27 \times \xi\eta(1 - \xi - \eta)(\xi - \xi_b) \\ b_3(\xi, \eta) = 27 \times \xi\eta(1 - \xi - \eta)(\eta - \eta_b) \end{array} \tag{29}$$

with $c_e = (\xi_b, \eta_b)$, $\xi_b = 1/3$ and $\eta_b = 1/3$.

Approximating $u'_{bub}(\mathbf{x})$ by Taylor series and neglecting the second order terms, we have an expression of $u'_{bub}(\mathbf{x})$ close to the centroid $\mathbf{c}_i$ of the element [35]:

$$u'_{bub}(\mathbf{x}) \approx b_1(\mathbf{x})(f - \mathcal{L}u_h)(\mathbf{c}_i) + b^e_{y_1}(\mathbf{x}) \frac{d(f - \mathcal{L}u_h)}{dy_1}|_{y=c_i} + b^e_{y_2}(\mathbf{x}) \frac{d(f - \mathcal{L}u_h)}{dy_2}|_{y=c_i} + HO \tag{30}$$

where:

$$b^e_{y_1} = \int_{\Omega_e} g^e(\mathbf{x}, \mathbf{y})(y_1 - c_{i,1}) d\Omega_y \quad and \quad b^e_{y_2} = \int_{\Omega_e} g^e(\mathbf{x}, \mathbf{y})(y_2 - c_{i,2}) d\Omega_y \tag{31}$$

As said before, we make the assumption that the residual $f - \mathcal{L}u_h$ is P0. Therefore, we have:

$$\frac{d(f - \mathcal{L}u_h)}{dy_1}|_{y_1=c_i} = \frac{d(f - \mathcal{L}u_h)}{dy_2}|_{y_2=c_i} = 0 \tag{32}$$

Consequently, the internal residual can be simply expressed as follows:

$$u'_{bub}(\mathbf{x}) \quad = b_1(\mathbf{x})(f - \mathcal{L}u_h)(\mathbf{c}_i) \tag{33}$$

Developing the residual with the convection-diffusion generic operator $\mathcal{L}$, we have:

$$u'_{bub}(\mathbf{x}) \quad = b_1(\mathbf{x})(f(\mathbf{c}_i) + a\Delta u_h(\mathbf{c}_i) - \mathbf{v} \cdot \nabla u_h(\mathbf{c}_i)) \tag{34}$$

Then, u_h is P1, therefore, $\nabla u_h(\mathbf{c}_i)$ is a constant inside the element and $\Delta u_h(\mathbf{c}_i) = 0$. For this reason, we finally get:

$$u'_{bub}(\mathbf{x}) = b_1(\mathbf{x})(f(\mathbf{c}_i) - \mathbf{v} \cdot \nabla u_h(\mathbf{c}_i)) \tag{35}$$

The above computation of the error estimator is *pointwise*. In fact, the error estimator is given at each point $\mathbf{x}$ of the domain. However, to include the error information in the mesh adaptation, we need an information of the error inside each element. To get this information, we can compute 3 different types of norm: L^1, L^2 and L^∞. We recall here the definition of these 3 norms:

$$||u'_{bub}||_{L^1}(\Omega_e) = |\Omega_e| \times \sum_{1 \leqslant i \leqslant N_{interp}} |u'_{bub}(\mathbf{x}_i)| \tag{36}$$

$$||u'_{bub}||_{L^2}(\Omega_e) = \sqrt{|\Omega_e|} \times \sqrt{\sum_{1 \leqslant i \leqslant N_{interp}} |u'_{bub}(\mathbf{x}_i)|^2} \tag{37}$$

$$||u'_{bub}||_{L^\infty}(\Omega_e) = max \left\{ \ |u'_{bub}(\mathbf{x})| \quad | \quad \mathbf{x} = (\mathbf{x}_i)_{1 \leqslant i \leqslant N_{interp}} \right\} \tag{38}$$

where $(\mathbf{x}_i)_{1 \leqslant i \leqslant N_{interp}}$ are the interpolation points defined by the integration points of the bubble function $b_1(\mathbf{x})$ and N_{interp} is the number of interpolation points.

3.2.2. Validation of the error estimator computation with bubbles functions

Now that we have developed practical computation aspects, we validate the computation on a benchmark of a convection-diffusion equation. To do so, we take the example given in [35] of an error estimation where the FEM is stabilized by the SUPG scheme. We recall that the differential operator is given by: $\mathcal{L}u = -a\Delta u + \mathbf{v} \cdot \nabla u$ with $a = 0.03$, $f = 1$, $\mathbf{v} = (1, 1)$ and homogeneous Dirichlet boundary conditions.

We compute the SUPG solution on P1 triangular elements instead of the Q2 bilinear quadrilateral elements used in the referred paper. Therefore, we choose a number of elements in order to get approximately the same SUPG solution in comparison with Irisarri et al.. We choose to divide each square cells in 8 triangles and we obtain a mesh of 512 triangular elements. We compare our SUPG solution with the reference on Figure 1.

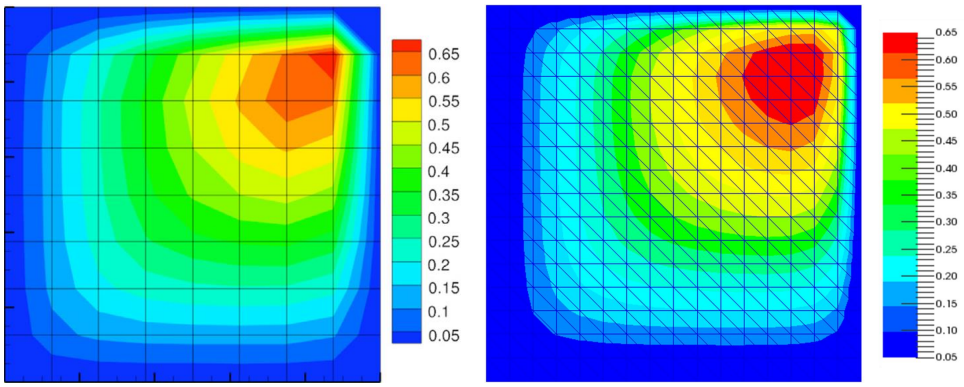


Figure 1: Comparison of the SUPG solution between Irisarri et al. [35] (left) and our solution (right)

Then, we compute the error estimate $||u'_{bub}||_{L^\infty}(\Omega_e)$ as explain in the previous subsection. We compare our results with the referred paper on Figure 2.

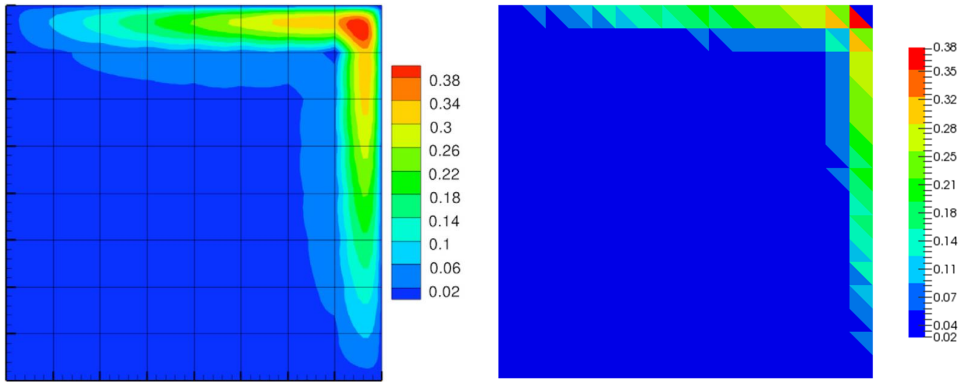


Figure 2: Comparison of error estimate between Irisarri et al. [35] (left) and our error estimation (right)

Remark 2. . *As the reader will notice, we don't have the same graphic representation between the two figures. In fact, the graphic representation of Irisarri et al. keep the pointwise definition of the error estimator. It is therefore a point cloud representation. On the contrary, we use an element-wise graphic representation. We use this representation because, as we will*

see, we only need the maximum value inside the element for our use of the error estimator in mesh adaptation.

Despite the difference in graphic representations, we observe a good concordance between the computation of Irisarri et al. and our computation. In fact, we reach approximately the same maximal values inside the elements and the error is located in the same area for both computations. Therefore, we validate the computation of the error estimator with bubble functions. We will see in the next section how to use this information in mesh adaptation.

4. Mesh adaptation with the subscales error estimator

In this section, we propose 3 different types of mesh adaptation:

1. An isotropic mesh adaptation: This technique uses the previously described subscales error estimator. From the latter, we build an isotropic metric tensor $\mathcal{H}_{iso}$ to adapt the mesh.
2. An anisotropic mesh adaptation (described in [5]): This technique uses the anisotropic local interpolation error indicator. From the latter, we derived an anisotropic metric tensor $\mathcal{H}_{aniso}$ to adapt the mesh.
3. A *new* anisotropic mesh adaptation: This technique uses a *new* anisotropic local error indicator that takes into account (i) the interpolation error indicator and (ii) the subscales error estimator. From the latter, we derive a new metric tensor $\mathcal{H}^{new}_{aniso}$.

As a reminder, we start by introducing the principles of anisotropic mesh adaptation and the construction of the anisotropic metric tensor $\mathcal{H}_{aniso}$. Then, the isotropic tensor $\mathcal{H}_{iso}$ based on the subscales error estimator is defined in Section 4.2. Finally, a combination between the anisotropic local error indicator and the subscales error estimator is proposed and the new metric tensor $\mathcal{H}^{new}_{aniso}$ is defined in Section 4.3.

4.1. Principles of anisotropic mesh adaptation

To discretize our computational domain, we use anisotropic and unstructured meshes. It implies that we can stretch the elements in certain directions according to the solution features. To do so, we start by performing an error analysis on the mesh. Then, to correlate the error with the geometry, a metric field is defined. From this metric field, an anisotropic error indicator is defined and used as a functional for a re-meshing optimization problem.

Let us consider a certain triangulation Ω_h. We can derive an upper bound of the approximation error using an interpolation error analysis in the L^p norm. Referring to Almeida et al. in [18], this upper bound is expressed thanks to the recovered Hessian of the approximated solution u_h. In fact, using P1 linear elements, we usually cannot compute directly the Hessian of the solution. Instead, we compute an approximation called the *recovered* Hessian matrix:

$$||u - u_h||_{L^p(\Omega)} \leq C' \mathcal{N}_{\Omega_h}^{-\alpha} ||H_R(u_h)(x)||^p_{L^p(\Omega_e)} \tag{39}$$

where $\alpha \geq 0$, $\mathcal{N}_{\Omega_h}$ the number of elements of the mesh, $H_R(u_h)(x)$ the *recovered* Hessian matrix and C' is independent of the element size.

To apply the re-meshing strategy, we build an equilateral tetrahedron in the metric space. As in [5], it is defined at an arbitrary point P by the local metric field $\mathcal{M}$:

$$\mathcal{M}(P) = \frac{1}{h_1(P)} e_1 \otimes e_1 + ... + \frac{1}{h_d(P)} e_d \otimes e_d \tag{40}$$

with $(e_i)_{i=1,d}$ the eigenvectors of $H_R(u_h(x))$ and $h_i(P)$ the mesh sizes in the e_i directions.

However, the *recovered* Hessian matrix is not a metric because it is not positive definite. Therefore, we define the following metric tensor:

$$\mathcal{H}_{aniso} = \mathcal{R} \Lambda \mathcal{R}^T \tag{41}$$

where $\mathcal{R}$ is the orthogonal matrix built with the eigenvectors $(e_i)_{i=1,d}$ of $H_R(u_h(x))$ and $\Lambda = diag(|\lambda_1|, ..., |\lambda_d|)$ is the diagonal matrix of absolute value of the eigenvalues of $H_R(u_h(x))$. This metric tensor can also be written as follow:

$$\mathcal{H}_{aniso} = \mathcal{R} \Lambda \mathcal{R}^T = |\lambda_1| e_1 \otimes e_1 + ... + |\lambda_d| e_d \otimes e_d \tag{42}$$

Here, we want to align the mesh with the solution field u. It means that we want the error to be equi-distributed on each direction of the domain. To do so, the shape of each element have to be such that the local error is equal in any direction. It is equivalent to say that the local error is constant per element in the principal directions of curvature. Therefore, we have:

$$|\lambda_1| h_1^2 = ... = |\lambda_d| h_d^2 = cte \tag{43}$$

Following the work of Mesri et al., we introduce the following local error indicator in the L^p norm:

$$\eta_{\Omega_e} = d|\Omega_e|^{\frac{1}{p}}|\lambda_d(x_0)|h_d^2 \tag{44}$$

where $|\lambda_d(x_0)|$ is the maximum eigenvalues of $H_R(u_h(x))$ corresponding to direction d, $|\Omega_e|$ is the volume of the element and h_d is the length of the element in direction d.

In [5], the authors define a minimization problem where the functional is the error indicator of Eq. (44). This optimization problem is expressed as follow:

$$\begin{cases} \text{Find } h_{\Omega_e} = \{h_{1,\Omega_e}, ..., h_{d,\Omega_e}\},\ \Omega_e \in \Omega_h \text{ that minimizes the cost function:} \\ F(h_{\Omega_e}) = \displaystyle\sum_{\Omega_e \in \Omega_h} (\eta_{\Omega_e})^p \\ \text{under the constraint:} \\ \\ \mathcal{N}_{\Omega'_h} = C_0^{-1} \displaystyle\sum_{\Omega_e \in \Omega_h} \int_{\Omega_e} \prod_{i=1}^{d} \frac{1}{h_{i,\Omega_e}} d\Omega_e \\ \text{where } C_0 \text{ is the volume of a regular tetrahedron and } \Omega'_h \text{ is the new triangulation.} \end{cases} \tag{45}$$

Then, this multidimensional optimization problem is replaced by a one-dimensional optimization problem. The unknown is no more h_{Ω_e} but the mesh size corresponding to the max of the $(h_{i,\Omega_e})_{1 \leqslant i \leqslant d}$. The way of dealing with anisotropy is via stretching factors $(s_{i,\Omega_e})_{1 \leqslant i \leqslant d-1}$ defined between the mesh sizes $(h_{i,\Omega_e})_{1 \leqslant i \leqslant d}$. The solution is given by Mesri et al. in [5]. In 3D and L^p norm, it is expressed thanks to the following theorem:

Theorem 1. *For $d = 3$, the optimization problem (45) has a unique solution and is given by:*

$$\begin{cases} h_{3,\Omega_e} = \left[\dfrac{\beta}{\frac{(2p+3)}{3} C_{1,\Omega_e}} \displaystyle\int_{\Omega_e} C_{2,\Omega_e} d\Omega_e \right]^{\frac{1}{2(p+3)}} \\ h_{2,\Omega_e} = s_{2,\Omega_e} h_{3,\Omega_e} \\ \\ h_{1,\Omega_e} = s_{1,\Omega_e} s_{2,\Omega_e} h_{3,\Omega_e} \end{cases}$$

with

$$\begin{cases} s_{1,\Omega_e} = \dfrac{h_1}{h_2} = \left(\dfrac{|\lambda_2|}{|\lambda_1|}\right)^{\frac{1}{2}} \\ s_{2,\Omega_e} = \dfrac{h_2}{h_3} = \left(\dfrac{|\lambda_3|}{|\lambda_2|}\right)^{\frac{1}{2}} \end{cases}$$

where

$$C_{1,\Omega_e} = 3^p C_0 s_{1,\Omega_e} s_{2,\Omega_e}^2 \times \left(\frac{||u'||_{L^\infty}(\Omega_e)}{u'_{TOL}}\right)^p \times |\lambda_3|^p, \quad C_{2,\Omega_e} = C_0^{-1} \frac{1}{s_{1,\Omega_e} s_{2,\Omega_e}^2} \quad \textit{and}$$

$$\beta^{\frac{1}{\frac{2}{3}(p+3)}} = \mathcal{N}_{\Omega'_h}^{-1} \sum_{\Omega_e \in \Omega_h} \left\{ \left(\frac{1}{\int_{\Omega_e} C_{2,\Omega_e} d\Omega_e}\right)^{\frac{1}{\frac{2}{3}(p+3)}} \int_{\Omega_e} C_{2,\Omega_e} \left[\frac{2p+3}{3} C_{1,\Omega_e}\right]^{\frac{1}{\frac{2}{3}(p+3)}} d\Omega_e \right\}$$

Finally, the above solution gives the mesh sizes in the 3 directions that define the metric field on each element. These mesh sizes are computed with respect to a fixed number of elements $\mathcal{N}_{\Omega'_h}$.

4.2. Isotropic mesh adaptation with the subscales error estimator

The first use of the subscales error estimator computed in Section 3 is for an *isotropic* mesh adaptation. Referring to [42] and from a theoretical convergence point of view, we know that the error is linked to the local mesh size h according to the following relation:

$$||u'||_{L^\infty}(\Omega) \approx C.h^{k+1} \tag{46}$$

where C is independent of the element size. In our case, using linear elements, we have:

$$||u'||_{L^\infty}(\Omega) \approx C.h^2 \tag{47}$$

To control the mesh size of our adaptive process, the user has to give a tolerance value $||u'_{tol}||_{L^\infty}$ so that the local mesh size stays above an acceptable bound. In fact, this tolerance corresponds to the desired error that the user wants to obtain on the mesh. For example, in a *pointwise* error estimation, $||u'_{tol}||_{L^\infty}$ is the tolerated error at each control point inside the element. Referring to [43], we can write the tolerated pointwise error as a scalar positive value u'_{TOL}:

$$||u'_{tol}||_{L^\infty} = u'_{TOL} \tag{48}$$

Starting from here, we can consider three strategies to adapt the mesh. Indeed, we can decide to either:

1. uniformly distribute the desired local error norm over the *old* mesh or,
2. uniformly distribute the desired local error norm over the *new* mesh or,
3. uniformly distribute the desired pointwise error.

In this paper, we will use the 2^{nd} mesh adaptation strategy for the *element-wise* computation of the error estimator (i.e. the one using stabilizing parameters); and the 3^{rd} mesh adaptation strategy for the *pointwise* computation of the error estimator (i.e. the one using bubble functions).

For the 3^{rd} strategy, still referring to [43], we write the relationship between the error and the tolerance in a uniform pointwise error distribution:

$$\left(\frac{h_{new}}{h}\right)^2 = \frac{u'_{TOL}}{u'} \tag{49}$$

where h_{new} is the size of the new mesh (after mesh adaptation), u' is the estimated error.

Then, we build the isotropic metric tensor:

$$\mathcal{H}_{iso} = \mathcal{R}\Lambda\mathcal{R}^T = |\lambda|e_1 \otimes e_1 + ... + |\lambda|e_d \otimes e_d \tag{50}$$

with

$$|\lambda| = \frac{1}{h_{new}^2} = \frac{||u'||_{L^\infty(\Omega_e)}}{u'_{TOL}} \times \frac{1}{h^2} \tag{51}$$

where $\mathcal{R}$ is the orthogonal matrix built with the eigenvectors $(e_i)_{i=1,d}$ of $H_R(u_h(x))$.

Here, the eigenvalues λ of the metric are equal in all directions. Therefore, the mesh will be adapted *isotropically*. However, we keep the eigenvectors of $H_R(u_h(x))$ in the definition of $\mathcal{H}_{iso}$. Thus, we keep the element's orientation prescribed by the *recovered* Hessian matrix. Finally, the re-meshing strategy is the same that the one described in Section 4.1. In particular, as we apply an isotropic re-meshing, the stretching factors $(s_{i,\Omega_e})_{1\leqslant i\leqslant d-1}$ between the mesh sizes $(h_{i,\Omega_e})_{1\leqslant i\leqslant d}$ are equal to 1.

4.3. Combination of subscales error estimator with anisotropic mesh adaptation

The second use of the subscales error estimator computed in Section 3 is for an *anisotropic* mesh adaptation. In this section, we propose to combine

both the coarse scales error indicator and the subscales error estimator. To do so, we derive a new metric that allows to take into account the anisotropic variations of the solution on the mesh but also relies on the subscales error estimator previously computed.

We first consider the previous anisotropic local error indicator define in Eq. (44) and recalled here:

$$\eta_{\Omega_e} = d|\Omega_e|^{\frac{1}{p}}|\lambda_d(x_0)|h_d^2$$

Then, going back to the previous section, we write:

$$\frac{h_{d,new}^2}{h_d^2} = \frac{u'_{TOL}}{||u'||_{L^\infty}(\Omega_e)}$$

Now, the unknown of the re-meshing problem is $h_{d,new}$. In fact, we want the new mesh size to take into account the error estimator of the subscales. Thus, we propose a new anisotropic local error indicator:

$$\eta_{\Omega_e,new} = d|\Omega_e|^{\frac{1}{p}} \times |\lambda_d(x_0)| \times \frac{||u'||_{L^\infty}(\Omega_e)}{u'_{TOL}} \times h_{d,new}^2 \tag{52}$$

From here, we can define the new anisotropic metric tensor as:

$$\mathcal{H}_{aniso}^{new} = \mathcal{R}\Lambda\mathcal{R}^T = \frac{||u'||_{L^\infty}(\Omega_e)}{u'_{TOL}}|\lambda_1|e_1 \otimes e_1 + ... + \frac{||u'||_{L^\infty}(\Omega_e)}{u'_{TOL}}|\lambda_d|e_d \otimes e_d \tag{53}$$

Doing so, we keep the anisotropic effects from the solution variations but we isotropically scale this effect by the subscales error estimator.

With this new error indicator, we solve the same optimization problem of Eq. 45 with the unknown $h_{d,new}$. The optimal mesh is obtained exactly in the same way but using $\eta_{\Omega_e,new}$ as functional. Consequently, the optimization

problem becomes:

$$\begin{cases} \text{Find } h_{\Omega_e,new} = \{h_{1,\Omega_e,new}, ..., h_{d,\Omega_e,new}\},\ \Omega_e \in \Omega_h \text{ that minimizes the cost function:} \\ F(h_{\Omega_e,new}) = \sum_{\Omega_e \in \Omega_h} (\eta_{\Omega_e,new})^p \\ \text{under the constraint:} \\ \mathcal{N}_{\Omega'_h} = C_0^{-1} \sum_{\Omega_e \in \Omega_h} \int_{\Omega_e} \prod_{i=1}^{d} \frac{1}{h_{i,\Omega_e,new}} d\Omega_e \\ \text{where } C_0 \text{ is the volume of a regular tetrahedron and } \Omega'_h \text{ is the new triangulation.} \end{cases} \tag{54}$$

Following the same proof that Mesri et al. in [5], the solution of this new optimization problem becomes:

Theorem 2. *For $d = 3$, the optimization problem (54) has a unique solution and it is given by:*

$$\begin{cases} h_{3,\Omega_e,new} = \left[\dfrac{\beta}{\frac{(2p+3)}{3} C_{1,\Omega_e,new}} \displaystyle\int_{\Omega_e} C_{2,\Omega_e} d\Omega_e \right]^{\frac{1}{2(p+3)}} \\ h_{2,\Omega_e,new} = s_{2,\Omega_e} h_{3,\Omega_e,new} \\ h_{1,\Omega_e,new} = s_{1,\Omega_e} s_{2,\Omega_e} h_{3,\Omega_e,new} \end{cases}$$

with

$$\begin{cases} s_{1,\Omega_e} = \dfrac{h_1}{h_2} = \left(\dfrac{|\lambda_2|}{|\lambda_1|} \right)^{\frac{1}{2}} \\ s_{2,\Omega_e} = \dfrac{h_2}{h_3} = \left(\dfrac{|\lambda_3|}{|\lambda_2|} \right)^{\frac{1}{2}} \end{cases}$$

where

$$C_{1,\Omega_e,new} = 3^p C_0 s_{1,\Omega_e} s_{2,\Omega_e}^2 \times \left(\frac{||u'||_{L^\infty(\Omega_e)}}{u'_{TOL}} \right)^p \times |\lambda_3|^p, \quad C_{2,\Omega_e} = C_0^{-1} \frac{1}{s_{1,\Omega_e} s_{2,\Omega_e}^2} \quad \textit{and}$$

$$\beta^{\frac{1}{\frac{2}{3}(p+3)}} = \mathcal{N}_{\Omega'_h}^{-1} \sum_{\Omega_e \in \Omega_h} \left\{ \left(\frac{1}{\int_{\Omega_e} C_{2,\Omega_e} d\Omega_e} \right)^{\frac{1}{\frac{2}{3}(p+3)}} \int_{\Omega_e} C_{2,\Omega_e} \left[\frac{2p+3}{3} C_{1,\Omega_e,new} \right]^{\frac{1}{\frac{2}{3}(p+3)}} d\Omega_e \right\}$$

By re-defining the optimization problem, we integrate the effect of the subscales error estimator with respect to the constraint on the fixed number of

elements. This new re-meshing strategy is tested on 2D and 3D benchmarks in the next section.

5. Numerical examples

In this section, we test the effectivity and the accuracy of our new local error indicator on 2D and 3D convection-diffusion benchmark cases in convection-dominated regime. To do so, we start by computing the Finite Element solution u_h thanks to the SUPG scheme described in this paper. The solution is computed on 6 different types of mesh:

- ***Fixed mesh:*** Using no mesh adaptation,
- ***Isotropic mesh 1:*** Using the isotropic metric tensor $\mathcal{H}_{iso}$ computed with the subscales error estimator using the stabilizing parameter (see Section 3.1),
- ***Isotropic mesh 2:*** Using the isotropic metric tensor $\mathcal{H}_{iso}$ computed with the subscales error estimator using the bubble functions (see Section 3.2),
- ***Anisotropic mesh 1:*** Using the anisotropic metric tensor $\mathcal{H}_{aniso}$ computed *without* the subscales error estimator,
- ***Anisotropic mesh 2:*** Using the new anisotropic metric tensor $\mathcal{H}^{new}_{aniso}$ computed with the subscales error estimator using the stabilizing parameter,
- ***Anisotropic mesh 3:*** Using the new anisotropic metric tensor $\mathcal{H}^{new}_{aniso}$ computed with the subscales error estimator using the bubble functions.

For the benchmark cases 5.1 and 5.3, we have a known analytic solution u. Thus, we can compute the exact discrete error $e_h = u - u_h$ on each element of the mesh. With this local discrete error, we compute the global L^2 norm of the error such that:

$$||e_h||_{L^2} = \left(\int_\Omega e_h^2 \, d\Omega \right)^{1/2} \tag{55}$$

For the case 5.1, we define an efficiency index I_{eff} for our subscales error estimator as follow:

$$I_{eff} = \frac{||u'||_{L^2}}{||e_h||_{L^2}} \tag{56}$$

5.1. Case 1: regular boundary layers in 2D

This first case has a continuous solution and regular boundary layers. It has been studied by several authors like Zhang et al. in [44] or Hachem et al. in [16]. We consider the domain $\Omega = (0,1)^2$ and the velocity field $\mathbf{v}(x,y) = (1,1)^T$. Instead of a source term $f = 1$ like in Section 3.2.2, we choose the source term corresponding to the following analytic solution:

$$u(x,y) = xy\left(1 - e^{-\frac{1-x}{a}}\right)\left(1 - e^{-\frac{1-y}{a}}\right) \tag{57}$$

Thus, we have the following source term:

$$f(x,y) = (x+y)\left(1 - e^{-\frac{1-x}{a}}e^{-\frac{1-y}{a}}\right) + (x-y)\left(e^{-\frac{1-y}{a}} - e^{-\frac{1-x}{a}}\right) \tag{58}$$

We compute the Finite Element SUPG solution u_h for two different diffusion coefficients: $a = \{10^{-3}; 10^{-4}\}$. In addition, for practical purpose, we define the following stretching factor:

$$S_{\Omega_e} = \frac{h_{new}^2}{h^2} = \frac{u'_{TOL}}{||u'||_{L^\infty}(\Omega_e)} \tag{59}$$

To keep an acceptable mesh even where $||u'||_{L^\infty}(\Omega_e)$ is close to zero, we have to impose a lower bound for the stretching factor S_{Ω_e}. We decide that, from an iteration to the other, the size of the element should not increase of more than 10 times the previous one. Therefore, the condition: $S_{\Omega_e} \geqslant 0.01$ is applied.

The effectivity index for the two ways of computation of the error estimator is given on Figure 3.

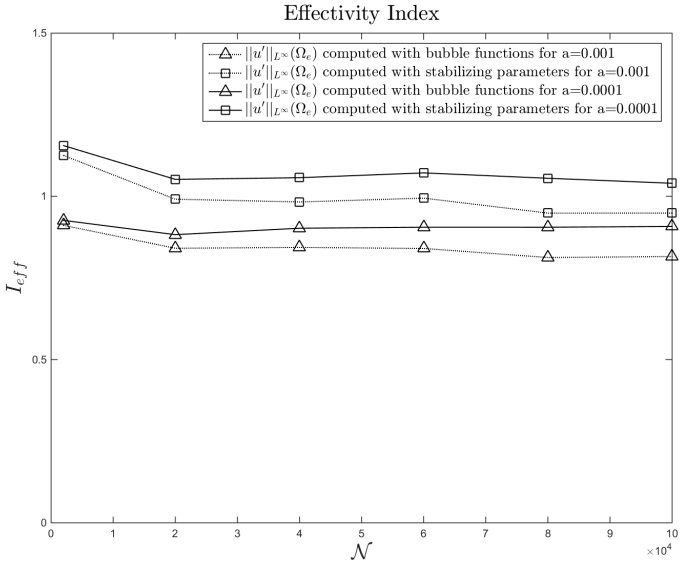


Figure 3: Effectivity indexes for $a = 10^{-3}$ and $a = 10^{-4}$

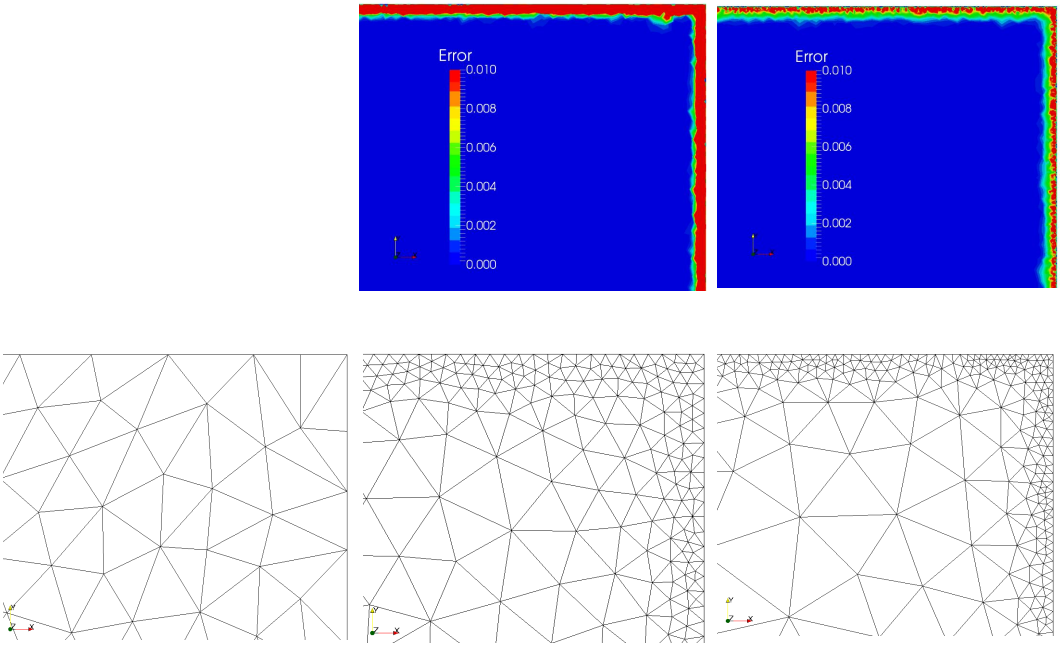
The first observation that can be made concerning the efficiency index I_{eff} is that for the two different Peclet numbers and for the different mesh sizes considered, our efficiency index I_{eff} stays close to 1. Therefore, we can say that our error estimator is effective for all configurations. Secondly, we observe that the error estimation computed with bubble functions is more effective than the one computed with stabilizing parameters which overestimates the error. Finally, we observe that the effectivity index I_{eff} gets closer to 1 when the diffusion coefficient is reduced (i.e. when the element Peclet number increases). This finding is in accordance with the convection-dominated regime assumption.

Now that we have confirmed the effectivity of our error estimation, we can use this estimation in the mesh adaptation. To analyze the mesh convergence of u_h, we compute u_h for 6 different mesh sizes: 2 000, 20 000, 40 000, 60 000, 80 000 and 100 000 elements.

The distributions of the discrete error e_h are presented on Figure 4. The corresponding meshes for each error distribution are given below. On this figure, the case corresponds to a diffusion coefficient of $a = 10^{-3}$. Mesh

adaptation is under a constraint of 20 000 elements and the figures are respectively:

- Figure 4.(a) corresponding to the *Fixed mesh*,
- Figure 4.(b) corresponding to the *Isotropic mesh 1*,
- Figure 4.(c) corresponding to the *Isotropic mesh 2*,
- Figure 4.(d) corresponding to the *Anisotropic mesh 1*,
- Figure 4.(e) corresponding to the *Anisotropic mesh 2*,
- Figure 4.(f) corresponding to the *Anisotropic mesh 3*.



(a) $N = 21\,000$ (b) $N = 21\,927$ (c) $N = 20\,798$

(d) $N = 20\,932$ (e) $N = 23\,834$ (f) $N = 23\,182$

Figure 4: Error distributions and meshes of 2D regular boundary layers for different mesh adaptation techniques

To begin the analysis of the results on the error distributions, we compare the Figure 4.(a), (b) and (c). We notice that the *isotropic* mesh adaptation according to our subscales error estimator reduces the error in the boundary layer. In fact, we notice from Figure 4.(b) and (c) that the mesh is isotropically adapted in the boundary layer. However, we observe that the error stays above the prescribed tolerance $u'_{TOL} = 0.01$ when we get close to the wall. To counter this effect, *anisotropic* mesh adaptation is a really efficient technique. On Figure 4.(d), we observe that the anisotropic mesh adaptation makes it possible for the error to go under the tolerance. However, we still keep an error approximately equals to the tolerance on the right top angle.

To eliminate this last part of the error, the new local error estimator proposed in Section 4.3 is well adapted as one can see on Figure 4.(f). In fact, we notice from the comparison of these 6 figures that our new local anisotropic error indicator $\eta_{\Omega_e,new}$ is the best driver of mesh adaptation. We observe on Figure 4.(e) and 4.(f) a drastic reduction of the error in the boundary layers compared to the other mesh adaptation techniques. Therefore, the use of the sub-scale information into the new anisotropic error indicator allows to improve the locality of the mesh nodes with respect to the equi-distribution of different error scales. Furthermore, we see that the equi-distribution is optimal when the error estimator is computed with the bubble functions (see Figure 4.(f)). To finish, we notice that the constraint of a fixed number of elements is well respected with a deviation of less than 19% with the targeted number of elements $\mathcal{N}$.

The reduction of the error can also be observed on the results for the global L^2 norm of the error $||e_h||_{L^2}$. This is true for the case $a = 10^{-3}$ on Figure 5 and the case $a = 10^{-4}$ on Figure 6. As expected, the *isotropic* mesh adaptation reduces the error for both cases of a. The error estimator computed with bubble functions shows to be the best driver of isotropic mesh adaptation; in particular when the number of element increases.

To increase the slope of convergence, we use *anisotropic* mesh adaptation. In fact, it has been shown in a number of works (for example in [5]) that the Hessian based anisotropic mesh adaptation leads to a convergence of order 2. This fact is highlighted in our results as we observe the increase of the slope going from isotropic to anisotropic mesh adaptation. On Figure 5 and 6, we observe that our new anisotropic local error indicator $\eta_{\Omega_e,new}$ behaves

as expected. This is the first important result of the present work. In fact, we observe that for coarse meshes (i.e. $\mathcal{N} = 2\ 000$ elements), the effect of the subscales error estimator is important. This reduction is even slightly better for the case with $a = 10^{-4}$. On the latter figure, we observe that, with only 20 000 elements, we are below the error of an interpolation based anisotropic mesh adaptation of 100 000 elements.

As the number of elements increases, we notice that the effect of the subscales error estimator becomes less important. Indeed, the slopes of the *Anisotropic mesh 2* and *Anisotropic mesh 3* cases decreases. This behavior is expected when we acknowledge that the subscales error estimator can also be considered as an estimation of the subscales error. Thus, when the number of element increases, the local mesh size decreases and, for this reason, the modeled part of the solution becomes smaller.

The second important result of the present work concerns the last (*Anisotropic mesh 3*) case corresponding to anisotropic mesh adaptation taking into account the error estimator computed with bubble functions. Again, we can state that this error estimator is the best choice to drive anisotropic mesh adaptation in comparison with the one computed with stabilizing parameters. In addition, we notice that the subscales error estimator computed with stabilizing terms has no effect when we reach an important number of elements (for example: $\mathcal{N} = 100\ 000$).

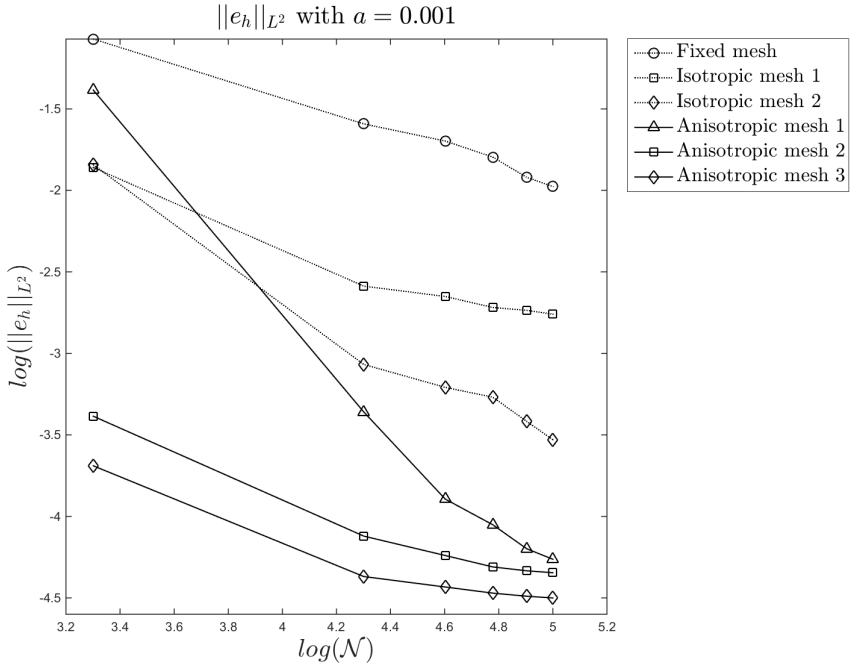


Figure 5: Error in L_2 norm with $a = 10^{-3}$ for regular boundary layers in 2D

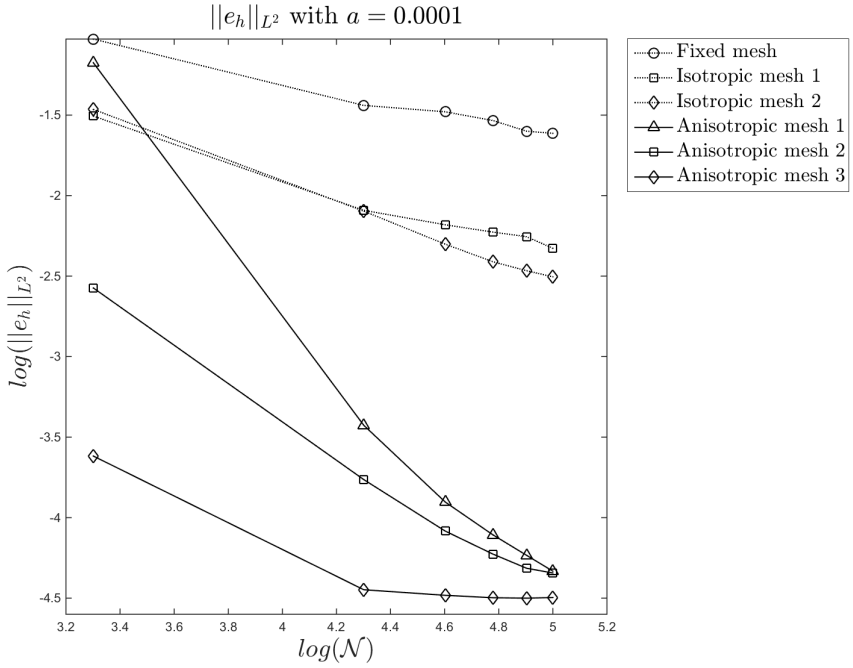


Figure 6: Error in L_2 norm with $a = 10^{-4}$ for regular boundary layers in 2D

5.2. Case 2: parabolic boundary layers with recirculating convective field

For this benchmark, we consider a domain $\Omega = (-1, 1)^2$ with a non-constant flow. The diffusion coefficient is $a = 0.001$. We apply a zero source term inside the domain. The right side wall has a Dirichlet boundary condition equals to 1. The other walls have a homogeneous Dirichlet boundary conditions. We apply a recirculating convective field defined by $\mathbf{v} = (2y(1 - x^2), -2x(1 - y^2))$ that will form boundary layers in the domain. The SUPG solution is given on Figure 7.

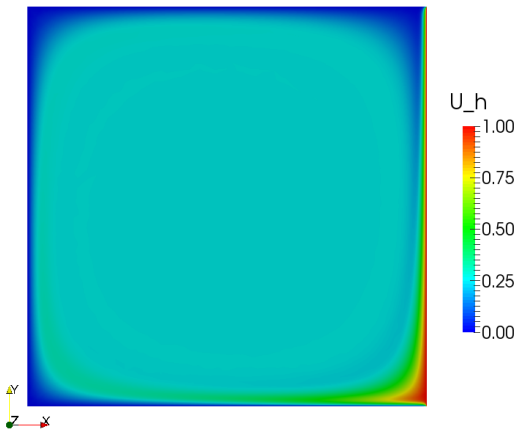


Figure 7: Numerical SUPG solution for parabolic layers with $a = 10^{-3}$

This example represents the propagation of the right wall heat inside the domain through the convective field. For this complex problem, it is difficult to capture the structure of the boundary layer. Our goal here is to show the efficiency of our new anisotropic error indicator in the capture of the parabolic layer. To illustrate the latter, we plot the two different error estimators (one computed with stabilizing parameter and the other with bubble functions) on Figure 8. The aim of this figure is to compare the two isotropic components of the two different error estimators. Therefore, it is sufficient to plot only the subscales part of the error estimation. As a result, we observe that the localization of the error is improved by the bubble functions method.

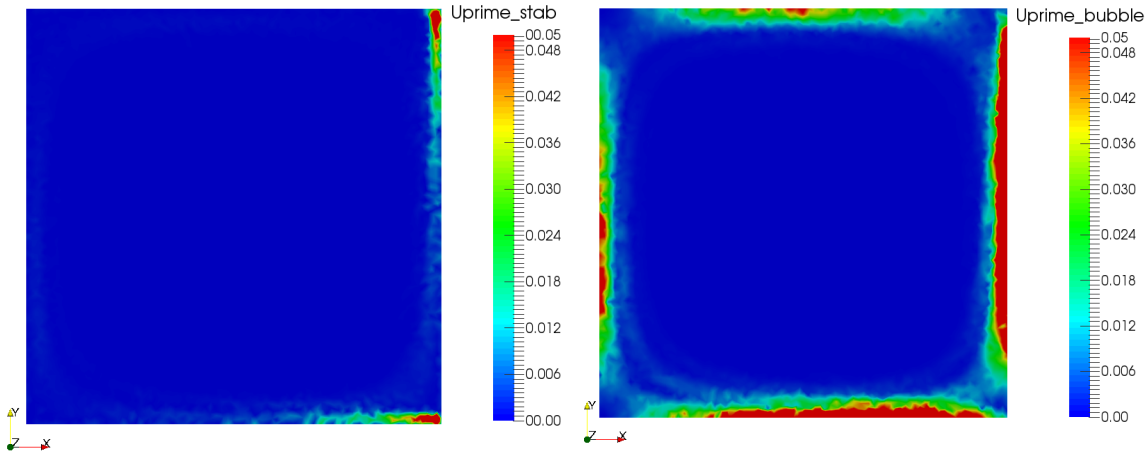


Figure 8: Subscales error estimator computed with stabilizing parameters (left) and bubble functions (right)

The resulting adapted meshes are presented on Figure 9. The cases (a), (b), (c), (d), (e) and (f) correspond to the same mesh adaptation techniques than in the previous example (Case 1). On Figure 9 (b) and (e) we observe that the isotropic and anisotropic mesh adaptation techniques follow the localization prescribed by the error estimator based on the stabilizing parameter (see Figure 8). Consequently, the mesh is adapted only on a restraint region of the parabolic boundary layer. On Figure 9(d), we observe that the conventional Hessian based anisotropic mesh adaptation already capture the 4 boundary layers. However, it does not take into account the sub-grid information that can improve the anisotropic mesh adaptation. In fact, the mesh adaptation technique using the *new* anisotropic local error indicator taking into account the subscales error indicator based on bubble functions (see Figure 9(f)) shows to be the best choice to accurately capture the 4 parabolic boundary layers.

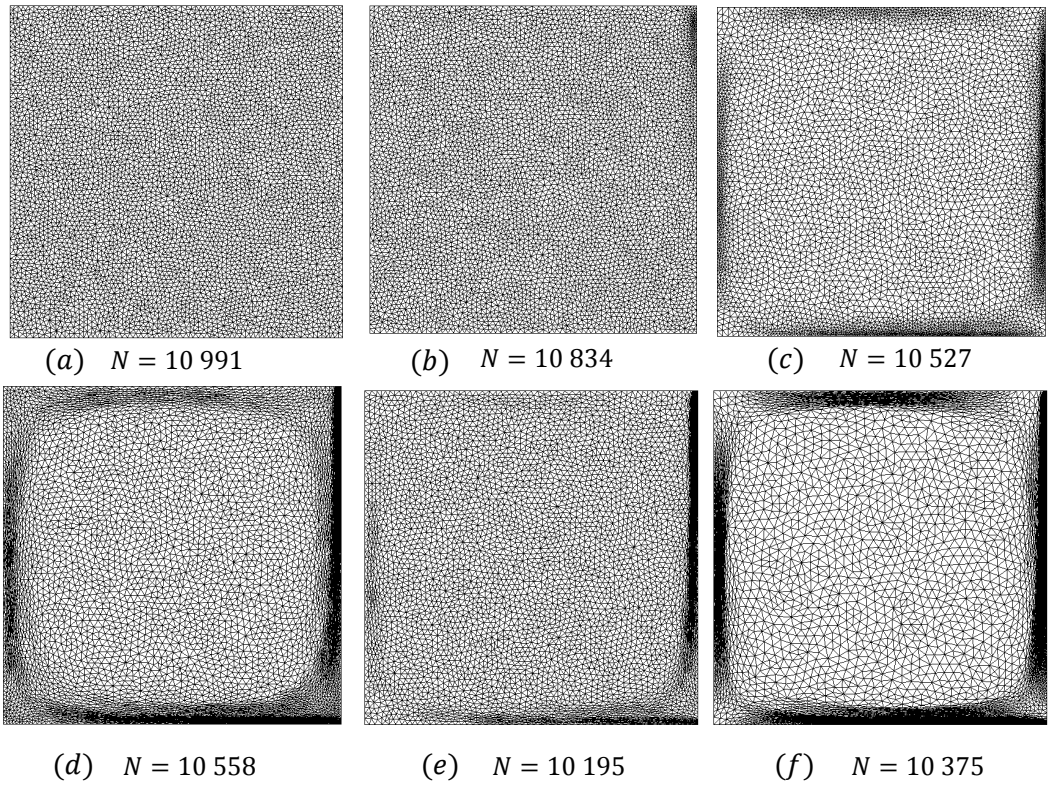


Figure 9: Resulting meshes for different types of mesh adaptation for the parabolic boundary layers.

5.3. Case 3: regular boundary layers in 3D

This 3D benchmark case is inspired by the one in Section 5.1. It takes the same configuration concerning velocity field, source term, boundary conditions and analytic solution with an extrusion in the z direction. However, the convection-diffusion equation is now solved on a cubic domain $\Omega = (0,1) \times (0,1) \times (0,1)$. The analytic solution is given on Figure 10. We compute the Finite Element SUPG solution u_h with the 6 mesh adaptation techniques previously describe for 9 mesh sizes: $\mathcal{N} = 40\,000,\ 80\,000,\ 120\,000,\ 160\,000,\ 200\,000,\ 300\,000,\ 400\,000,\ 50$ elements.

As for the 2D benchmark, the error distribution corresponding of each mesh is given on Figure 11.

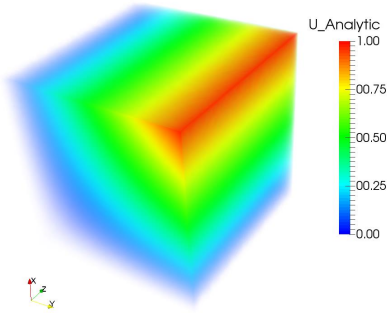
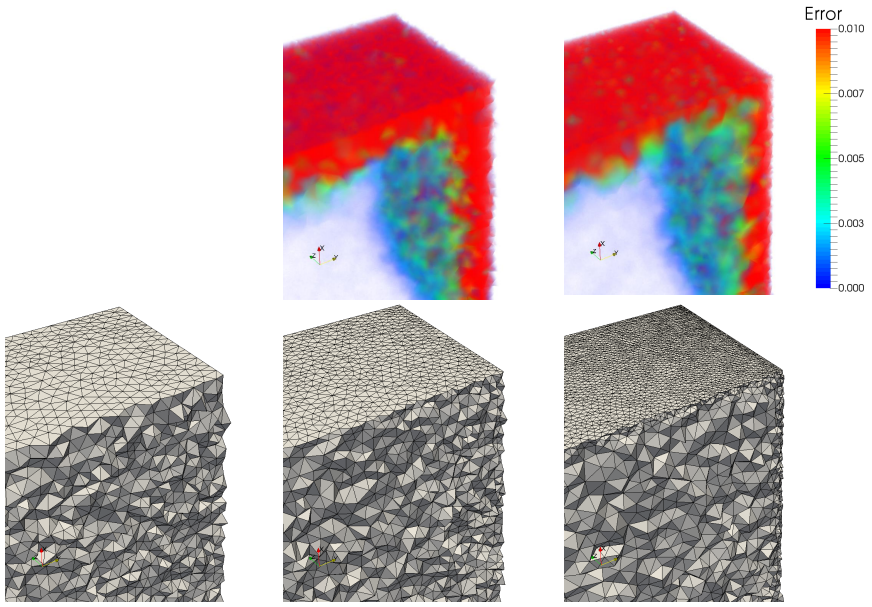


Figure 10: Analytic solution for 3D convection-diffusion equation with $a = 10^{-3}$



(a) $N = 200\ 360$ (b) $N = 194\ 045$ (c) $N = 198\ 946$

(d) $N = 212\ 242$ (e) $N = 200\ 933$ (f) $N = 199\ 659$

On Figure 11.(a), (b) and (c), we observe the instabilities due to the highly convective regime. Having isotropic elements, the local mesh size in the direction of the velocity field is too important. As said in the introduction of this paper, we observe that anisotropic mesh adaptation is a really efficient technique to eliminate the spurious oscillations in the boundary layer. This is highlighted on Figure 11.(d). However, there is still a part of the error that is not reduced on this figure. To counter this effect, we see that our new anisotropic local error indicator $\eta_{\Omega_e,new}$ is, again, efficient. Also, we observe that the constraint of a fixed number of elements is again well respected with a deviation of less than 6% with the targeted $\mathcal{N}$. Finally, the analysis on error distribution in 2D still holds in 3D and moreover, it seems to be enforce by adding a new dimension to the problem.

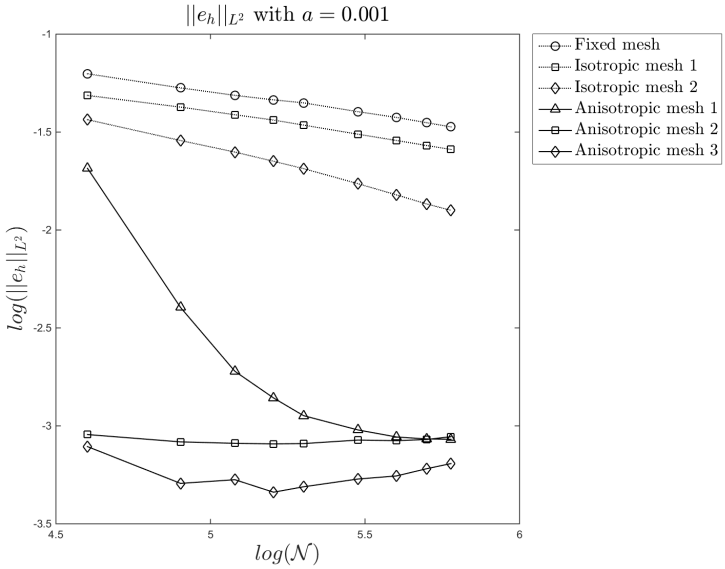


Figure 12: Error in L_2 norm with $a = 10^{-3}$ for regular boundary layers in 3D

The above statements on error distribution are also to be found in the global L^2 norm of the error $||e_h||_{L^2}$ on Figure 12. Moreover, the effects of the subscales error estimator on anisotropic mesh adaptation is enforced in 3D. The important result here is that, with only 40 000 elements in 3D, the error is below an interpolation based anisotropic mesh adaptation on 600 000 elements with our new anisotropic local error estimator $\eta_{\Omega_e,new}$. Again, taking

into account the sub-scale information in mesh adaptation makes it possible to reduce drastically the error on the 3D coarse meshes. With about 20 times less elements, we obtain the same global error $||e_h||_{L^2}$.

6. Conclusions

We proposed a new a posteriori error estimator based on the Variational Multiscale method for anisotropic adaptive fluid mechanics problems. This new error estimator is defined with the combination of both (i) an interpolation based anisotropic error indicator and (ii) a subscales error estimator. The results show that this combination allows to obtain highly precise solutions with much less elements in comparison with other mesh adaptation techniques. Furthermore, the definition of a new re-meshing optimization problem allows us to include the sub-grid information in mesh adaptation with respect to the constraint of a fixed number of elements.

The subscales error estimator is computed with two different methods and the results show that one using bubble functions is better suited to drive both isotropic and anisotropic mesh adaptation. In fact, the localization of the residual subscales error is better established with the latter.

This work shows that this new anisotropic mesh adaptation strategy is capable of dealing with boundary layers of convection-diffusion problems associated with high Peclet number flows in 2D and 3D. It is therefore a promising idea to simulate complex CFD aerothermal problems.

Acknowledgments

This work was supported by Safran Aircraft Engines [CIFRE 2015/1348].

References

[1] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, D. Mavriplis, CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences, Tech. rep. (2014).

[2] S. Brogniez, C. Farhat, E. Hachem, A high-order discontinuous Galerkin method with Lagrange multipliers for advection–diffusion problems,

Computer Methods in Applied Mechanics and Engineering 264 (2013) 49–66.

[3] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order CFD methods: current status and perspective, International Journal for Numerical Methods in Fluids 72 (8) (2013) 811–845.

[4] R. Sevilla, A. J. Gil, M. Weberstadt, A high-order stabilised ALE finite element formulation for the Euler equations on deformable domains, Computers & Structures 181 (2017) 89–102.

[5] Y. Mesri, W. Zerguine, H. Digonnet, L. Silva, T. Coupez, Dynamic parallel adaption for three dimensional unstructured meshes: Application to interface tracking, in: Proceedings of the 17th International Meshing Roundtable, Springer, 2008, pp. 195–212.

[6] Q. Meng, M. Berzins, Scalable large-scale fluid-structure interaction solvers in the Uintah framework via hybrid task-based parallelism algorithms: Uintah hybrid task-based parallelism algorithms, Concurrency and Computation: Practice and Experience 26 (7) (2014) 1388–1407.

[7] D. Ibanez, I. Dunn, M. S. Shephard, Hybrid MPI-thread parallelization of adaptive mesh operations, Parallel Computing 52 (2016) 133–143.

[8] T. Panitanarak, S. M. Shontz, A parallel log barrier-based mesh warping algorithm for distributed memory machines, Engineering with Computers (2017) 1–18.

[9] Y. Mesri, F. Alauzet, A. Loseille, L. Hascoët, B. Koobus, A. Dervieux, Continuous mesh adaptation models for CFD, CFD Journal 16 (4) (2008) 346–355.

[10] G. Jannoun, E. Hachem, J. Veysset, J.-F. Zaragoci, T. Coupez, Fully space-time metric based anisotropic mesh adaptation for unsteady problems, in: 6th International Conference on Adaptive Modeling and Simulation, ADMOS 2013, International Center for Numerical Methods in Engineering, Lisbon, Portugal, 2013, pp. 350–361.

[11] W. Hassan, M. Picasso, An anisotropic adaptive finite element algorithm for transonic viscous flows around a wing, Computers & Fluids 111 (2015) 33–45.

[12] E. Hachem, M. Khalloufi, J. Bruchon, R. Valette, Y. Mesri, Unified adaptive Variational MultiScale method for two phase compressible–incompressible flows, Computer Methods in Applied Mechanics and Engineering 308 (2016) 238–255.

[13] P. J. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, Computer Methods in Applied Mechanics and Engineering 194 (48–49) (2005) 5068–5082.

[14] R. Aubry, S. Dey, E. Mestreau, B. Karamete, Boundary layer mesh generation on arbitrary geometries, dOI: 10.1002/nme.5514 (2017).

[15] T. Coupez, Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing, Journal of Computational Physics 230 (7) (2011) 2391–2405.

[16] E. Hachem, G. Jannoun, J. Veysset, T. Coupez, On the stabilized finite element method for steady convection-dominated problems with anisotropic mesh adaptation, Applied Mathematics and Computation 232 (2014) 581–594.

[17] G. Jannoun, E. Hachem, J. Veysset, T. Coupez, Anisotropic meshing with time-stepping control for unsteady convection-dominated problems, Applied Mathematical Modelling 39 (7) (2015) 1899–1916.

[18] R. C. Almeida, R. A. Feijóo, A. C. Galeao, C. Padra, R. S. Silva, Adaptive finite element computational fluid dynamics using an anisotropic error estimator, Computer Methods in Applied Mechanics and Engineering 182 (3) (2000) 379–400.

[19] Y. Mesri, M. Khalloufi, E. Hachem, On optimal simplicial 3d meshes for minimizing the Hessian-based errors, Applied Numerical Mathematics.

[20] A. Loseille, F. Alauzet, Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error, SIAM Journal on Numerical Analysis 49 (1) (2011) 38–60.

[21] A. Agouzal, Y. V. Vassilevski, Minimization of gradient errors of piecewise linear interpolation on simplicial meshes, Computer Methods in Applied Mechanics and Engineering 199 (33–36) (2010) 2195–2203.

[22] P. Labbé, J. Dompierre, M.-G. Vallet, F. Guibault, Verification of three-dimensional anisotropic adaptive processes, International Journal for Numerical Methods in Engineering 88 (4) (2011) 350–369.

[23] Y. Mesri, H. Guillard, T. Coupez, Automatic coarsening of three dimensional anisotropic unstructured meshes for multigrid applications, Applied Mathematics and Computation 218 (21) (2012) 10500–10519.

[24] L. Kamenski, W. Huang, How a Nonconvergent Recovered Hessian Works in Mesh Adaptation, SIAM Journal on Numerical Analysis 52 (4) (2014) 1692–1708.

[25] A. N. Brooks, T. J. R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Computer Methods in Applied Mechanics and Engineering 32 (1) (1982) 199–259.

[26] T. J. R. Hughes, L. P. Franca, M. Mallet, A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics, Computer Methods in Applied Mechanics and Engineering 54 (2) (1986) 223–234.

[27] T. J. R. Hughes, L. P. Franca, G. M. Hulbert, A new finite element formulation for computational fluid dynamics: VIII. The galerkin/least-squares method for advective-diffusive equations, Computer Methods in Applied Mechanics and Engineering 73 (2) (1989) 173–189.

[28] T. J. R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Computer Methods in Applied Mechanics and Engineering 127 (1) (1995) 387–401.

[29] B. N. Granzow, M. S. Shephard, A. A. Oberai, Output-based error estimation and mesh adaptation for variational multiscale methods, Computer Methods in Applied Mechanics and Engineering 322 (2017) 441–459.

[30] J. Baiges, R. Codina, Variational Multiscale error estimators for solid mechanics adaptive simulations: An Orthogonal Subgrid Scale approach, Computer Methods in Applied Mechanics and Engineering.

[31] V. John, J. Novo, A robust SUPG norm a posteriori error estimator for stationary convection–diffusion equations, Computer Methods in Applied Mechanics and Engineering 255 (2013) 289–305.

[32] Multiscale multi-dimensional explicit a-posteriori error estimation for fluid dynamics | TU Delft Repositories.

[33] G. Hauke, D. Fuster, M. H. Doweidar, Variational multiscale a-posteriori error estimation for multi-dimensional transport problems, Computer Methods in Applied Mechanics and Engineering 197 (33–40) (2008) 2701–2718.

[34] D. Irisarri, G. Hauke, Pointwise Error Estimation for the One-Dimensional Transport Equation Based on the Variational Multiscale Method, International Journal of Computational Methods (2016) 1750040.

[35] D. Irisarri, G. Hauke, A posteriori pointwise error computation for 2-D transport equations based on the variational multiscale method, Computer Methods in Applied Mechanics and Engineering 311 (2016) 648–670.

[36] A. Papastavrou, R. Verfürth, A posteriori error estimators for stationary convection–diffusion problems: a computational comparison, Computer Methods in Applied Mechanics and Engineering 189 (2) (2000) 449–462.

[37] L. Formaggia, S. Perotto, P. Zunino, An anisotropic a-posteriori error estimate for a convection-diffusion problem, Computing and Visualization in Science 4 (2) (2001) 99–104.

[38] M. Picasso, Adaptive finite elements with large aspect ratio based on an anisotropic error estimator involving first order derivatives, Computer Methods in Applied Mechanics and Engineering 196 (1–3) (2006) 14–23.

[39] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, J.-B. Quincy, Advances in Stabilized Methods in Computational MechanicsThe variational multiscale method—a paradigm for computational mechanics, Computer Methods in Applied Mechanics and Engineering 166 (1) (1998) 3–24.

[40] G. Hauke, M. H. Doweidar, M. Miana, The multiscale approach to error estimation and adaptivity, Computer Methods in Applied Mechanics and Engineering 195 (13–16) (2006) 1573–1593.

[41] A. Ern, J.-L. Guermond, Éléments finis: théorie, applications, mise en oeuvre, Springer Science & Business Media, 2002, google-Books-ID: SHhnLz0EMywC.

[42] I. Babuška, T. Strouboulis, The Finite Element Method and Its Reliability, Clarendon Press, 2001, google-Books-ID: _Gv9KMOrA7sC.

[43] G. Hauke, M. H. Doweidar, S. Fuentes, Mesh adaptivity for the transport equation led by variational multiscale error estimators, International Journal for Numerical Methods in Fluids 69 (12) (2012) 1835–1850.

[44] Z. Zhang, Finite element superconvergence on Shishkin mesh for 2-D convection-diffusion problems, Mathematics of Computation 72 (243) (2003) 1147–1177.