

ANR-MAIDESC-T3-D1 - Adaptive time-step with anisotropic meshing for unsteady convection-dominated problems

G. Jannoun^a, E. Hachem^{a,*}, J. Veysset^a, T. Coupez^a

^a MINES ParisTech, Center for Materials Forming (CEMEF), UMR CNRS 7635, BP 207, 06904 Sophia-Antipolis, France

Abstract

In this work, we develop a new anisotropic space and time adaptive method for convection dominated problems with inner and boundary layers. A new route to construct a metric field directly at the nodes of the mesh is highlighted using the length distribution tensor and an edge based error analysis. A Streamline Upwind Petrov-Galerkin (SUPG) finite element method is employed to solve the unsteady convection-diffusion equation. The numerical experiments show that the use of both space and time adaptivity generates optimal time stepping, allows the recovery of the global convergence order of the numerical schemes, reduces the computational time and cost and produces accurate and oscillation free numerical solutions.

Keywords: Anisotropic Meshing, Adaptive time stepping, Convection-dominated problem, Boundary Layers, Stabilized finite element method

1. INTRODUCTION

The numerical solution of the unsteady convection-diffusion equation using the Galerkin formulation normally exhibits global spurious oscillations in convection-dominated problems, especially in the vicinity of sharp gradients. Over the last two decades, a variety of finite element approaches have been proposed to deal with such situations. These methods increase stability while maintaining consistency by adding weighted residual terms to the weak formulation of the problem. They have grown in popularity, especially in applications to fluid dynamics, heat transfer and fluid-structure interactions. We can find the SUPG method (Streamline Upwind Petrov-Galerkin) in [1, 2, 3], the Galerkin/Least-Squares (GLS) method in [4, 5], the gradient Galerkin/least-squares (GGLS) method in [6, 7], the unusual stabilized method (USFEM) in [8, 9, 10], the enriched method with time interpolation in [11], the subgrid scale method in [12] and many others, each was used to optimize the performance of the finite element formulations for advection-diffusion equation with or without production.

At the same time, from a practical point of view, in order to perform long-time and large-scale industrial applications it is preferable to choose a number of nodes N based on the available computational resources and determine the most accurate possible solution. So the level of accuracy is not set apriori but it is highly desirable to have the best representation of the simulated phenomena.

This goal opens the door to the emergence of many numerical techniques that aim at optimizing both the spatial and the temporal discretizations [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. Indeed, combining stabilized finite element methods with these techniques allows the recovery of the global convergence order of the numerical schemes, reduces the computational time and cost and produces accurate and oscillation free numerical solutions.

One of the most popular technique in this domain is the anisotropic mesh adaptation. It allows the capture of scale heterogeneity in particular when simulating multi-components systems as the discontinuities and high gradients of the solution are captured with good accuracy for a reasonable number of elements.

* Corresponding author, Tel. +33 (0)4 92 95 74 58; fax: +33 (0)4 92 38 97 52
Email address: elie.hachem@mines-paristech.fr (E. Hachem)

Anisotropic mesh adaptation was first proposed in the late 1980s [24, 25, 26, 27]. Significant research effort has been devoted in the last few years to devise fruitful anisotropic mesh adaptation techniques with real applications. We distinguish four major error estimates for anisotropic adaptation: the hessian based relying on the solution hessian information to evaluate the linear interpolation error [28, 29, 30, 31], the a posteriori estimates approximating the discretization error using a theoretical analysis [32, 33, 34, 35, 36], the a priori error estimates [37, 38] and the goal oriented estimates that provide mathematical framework for assessing the quality of some functionals [39, 40, 41, 42]. Thanks to these technical and theoretical advances a considerable improvement was obtained in the accuracy and the efficiency of numerical simulations.

The challenging construction of the anisotropic mesh adaptation can be conveyed to that of constructing an appropriate mesh tensor by means of a discretization error analysis. For instance, during a re-meshing process, the elements of the original mesh are much more volatile than its nodes. Thus for practical reasons the meshing tools demand a nodal metric map. Other challenges are related to the possibility of constructing error estimators and adapting the mesh while taking into account different fields in particular for coupled systems (i.e. velocity, temperature, concentration,...).

In this work, we propose an alternative to these challenges. The main target of the approach is to produce extremely stretched and highly directional elements under the constraint of a fixed number of nodes. We intend to develop a metric based mesh adaptation that is capable of well capturing the anisotropy of a physical phenomena. The metric tensor would prescribe optimal mesh sizes and element orientations. It is a procedure that can be divided into three key steps. We start by constructing a length distribution tensor followed by a second order optimal nodal gradient reconstruction procedure as proposed in [43]. Then we determine the edge based error estimates that are driven entirely from an a posteriori analysis without any prior assumptions. The developed algorithm strives to improve the quality of the aforementioned error estimates by attempting to reduce and equidistribute the error over the edges of the mesh under the constraint of a fixed degree of freedom. The novelty of this paper resides in the combination of an edge-based error estimation with the equidistribution principle to derive a set of edge stretching factors resulting in an optimal anisotropic mesh adaptation. Unlike the hessian-based techniques for metric construction, the method that we propose avoids the reconstruction of this tensor and renders a reduction in the computational cost.

The second objective of this paper is to use the information gathered from the developed anisotropic mesh adaptation procedure together with the equidistribution principle in order to compute time step size that would significantly improve the computational cost and the global accuracy of the calculations. In general, a small time-step size produces more accurate solutions but with a high computational cost since more time-steps and hence more computations are required. On one hand, adopting a fixed time-stepping all over the simulation, requires an apriori knowledge of the user about the choice of this constant as it should satisfy a stability condition to guarantee the convergence of the numerical solution to the analytical one. On the other hand, determining the time-step sizes through a *CFL* argument might produce very small time-steps as their values depend on the smallest mesh-size. This choice of time-steps can be computation-wise very limiting especially when used with anisotropic meshing, as in the latter technique the mesh sizes can become very small at the locations of high solution gradients.

Using the error equidistribution in space and time, the developed mesh adaptation method, allows the control of the L^p norm of space-time interpolation error. It can be applied to any type of problems, allows a good and automatic capture of boundary and emerging inner layers in particular for high Peclet number. With such an advantage, the proposed approach recovers the orders of convergence and could be seen as a very useful tool for very stiff and convection-dominated problems. It is important to mention that the method tends to refine the mesh in the hierarchical importance of the solution's gradient. In other words, if new features (with high gradients) appear in the solution, the mesh will be automatically coarsened in regions with lower gradient and will be refined near the newly emerging features. In this case, if a small number of nodes was fixed by the user, the solution will still be well captured although not with the same degree of accuracy.

The combination of the space and time adaptive algorithms constitutes a tool to study with the available computational resources (a limitation on the number of degrees of freedom) and simulate real large unsteady convection dominated problems. The method is first demonstrated to produce optimal anisotropic meshes

minimizing the L^2 norm of the interpolation error. Note that the only parameter to be chosen by the user is a fixed number of nodes that will provide a control of both space and time.

The method demonstrated second order convergence in space and time. The results were also compared with ones presented in the literature and appear in general agreement with them. Finally 2D and 3D coupled problems were considered to prove the performance of the method, its scalability and its applicability to any type of equations.

This paper is organized as follows: we start in section 2 with the description of the developed anisotropic mesh adaptation method. Then we present in section 3 the time adaptation technique. The Streamline Upwind Petrov Galerkin method that is used to solve the unsteady convection-diffusion equation is presented in section 4. Section 5 contains several numerical experiments showing the efficiency and the accuracy of the proposed method. Finally, section 6 is dedicated to the conclusion and the upcoming work.

2. Mesh adaptation

In general, approaches to easily build unstructured anisotropically adapted meshes ([44], [45], [46], [47]) are based on local modifications of an existing mesh. Theories about anisotropic a posteriori error estimation (see [37, 43, 48, 31, 49, 50]) have been well developed in the literature, leading to some standardization of the adaptation process; development of metrics from the analysis of the discretization error and the steering of remeshing by these metrics.

In this work, we develop a new route for the metric construction. The latter is done directly at the nodes of the mesh without any direct information from the elements, neither considering any underlying interpolation. It is performed by introducing a statistical concept: the length distribution function. A second order tensor is employed to approximate the distribution of lengths. It is defined by gathering the information from the edges onto their common node. Using such a technique, the error along and in the direction of each edge is computed. Keystone characteristics of this adaptive technique are its automaticity and its independence from the problem at hand. Then the theory is extended to take into account multi-components fields (tensors, vectors, scalars) within a single metric thus avoiding additional computations of metrics intersection as in [49]. An additional feature of this method is that it works under the constraint of a fixed number of nodes and hence maintains computationally affordable meshes.

2.1. Construction of an anisotropic mesh

The main idea of anisotropic, metric based, mesh adaptation is to generate a unit mesh (with unit length edges and regular elements) in a prescribed Riemannian metric space. As it is almost impossible to achieve this property, we intend to construct a mesh that is optimal (very close to unity) up to a certain tolerance. Note that the specialty of this work is that the tolerance is implicitly assigned by the user through the assignment of a fixed number of nodes. The resulting mesh is uniform in the metric space but anisotropic in the Euclidean space.

2.2. Edge based error estimation

An *a posteriori* error estimate based on the length distribution tensor approach and the associated edge based error analysis is presented. It enables the calculation, for each node, of a metric tensor that prescribes a density of neighboring nodes, a set of anisotropic directions and stretching factors along these directions. The optimal stretching factor field is obtained by solving an optimization problem using the equidistribution principle under the constraint of a fixed number of nodes in the mesh.

We consider a variable $u \in \mathcal{C}^2(\Omega) = \mathcal{V}$ and \mathcal{V}_h a simple P^1 finite element approximation space: $\mathcal{V}_h = \{w_h \in \mathcal{C}^0(\Omega), w_h|_K \in P^1(K), K \in \mathcal{K}\}$ where $\Omega = \bigcup_{K \in \mathcal{K}} K$ and K is a simplex (segment, triangle, tetrahedral, ...).

We define $\mathbf{X} = \{\mathbf{X}^i \in \mathbb{R}^d, i = 1, \dots, N\}$ as the set of nodes in the mesh, we denote by U^i the nodal value of u at \mathbf{X}^i and we let Π_h be the Lagrange interpolation operator from \mathcal{V} to \mathcal{V}_h such that: $\Pi_h u(\mathbf{X}^i) =$

$u(\mathbf{X}^i) = U^i, \forall i = 1, \dots, N$. As shown in figure 1, the set of nodes connected to node i is denoted by $\Gamma(i) = \{j, \exists K \in \mathcal{K}, \mathbf{X}^i, \mathbf{X}^j \text{ are nodes of } K\}$.

Starting from the assumption that for metric based adaptation methods, in order to master the global approximation error it is sufficient to control the interpolation error, the target can be summarized into finding the mesh, made up of at most N nodes, that minimizes the linear interpolation error in the L^p norm.

By introducing the notation: $\mathbf{X}^{ij} = \mathbf{X}^j - \mathbf{X}^i$ and using the analysis carried in [43], we can set:

$$\nabla u_h \cdot \mathbf{X}^{ij} = U^{ij}, \quad (1)$$

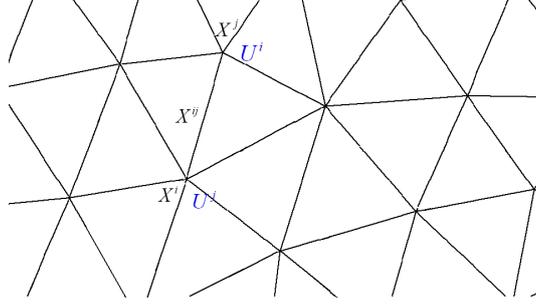


Figure 1: Length \mathbf{X}^{ij} of the edge joining nodes i and j .

$$\left\| \underbrace{\nabla u_h \cdot \mathbf{X}^{ij}}_{U^{ij}} - \nabla u(\mathbf{X}^i) \cdot \mathbf{X}^{ij} \right\| \leq \max_{\mathbf{Y} \in [\mathbf{X}^i, \mathbf{X}^j]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|, \quad (2)$$

where $\mathbb{H}(u) = \nabla^{(2)}u$ is the associated Hessian of u . Recall that taking $u \in \mathcal{C}^2(\Omega)$ we obtain $\nabla u \in \mathcal{C}^1(\Omega)$.

Applying the interpolation operator on ∇u together with (1) we obtain a definition of the projected second derivative of u in terms of only the values of the gradient at the extremities of the edge:

$$\nabla \mathbf{g}_h \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} = \mathbf{g}^{ij} \cdot \mathbf{X}^{ij}, \quad (3)$$

where $\mathbf{g}_h = \Pi_h \nabla u$, $\mathbf{g}^i = \nabla u(\mathbf{X}^i)$ and $\mathbf{g}^{ij} = \mathbf{g}^j - \mathbf{g}^i$.

Using a mean value argument, we set that: $\exists \mathbf{Y} \in [\mathbf{X}^i, \mathbf{X}^j] | \mathbf{g}^{ij} \cdot \mathbf{X}^{ij} = \mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}$.

We use this projection as an approximation of the error along the edge:

$$e_{ij} = \mathbf{g}^{ij} \cdot \mathbf{X}^{ij}. \quad (4)$$

However this equation cannot be evaluated exactly as it requires knowing the gradient of u and also its continuity at the nodes of the mesh. For that reason, we resort to a gradient recovery procedure.

2.3. Gradient Recovery

We use the method proposed in [43] for gradient recovery based on an optimization analysis. The latter will be defined by:

$$\mathbf{G}^i = (\mathbb{X}^i)^{-1} \sum_{j \in \Gamma(i)} U^{ij} \mathbf{X}^{ij}, \quad (5)$$

where $\mathbb{X}^i = \frac{d}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij}$ is what we call the length distribution tensor at node \mathbf{X}^i . Note that

this construction preserves the second order: $|(\mathbf{G}^i - \mathbf{g}^i) \cdot \mathbf{X}^{ij}| \sim (\mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij})$ where \mathbf{G}^i is the recovery gradient given by (5) and \mathbf{g}^i is the exact value of the gradient at node i .

Hence, the approximated error is evaluated by substituting \mathbf{G} by \mathbf{g} in (4):

$$e_{ij} = \mathbf{G}^{ij} \cdot \mathbf{X}^{ij}.$$

2.4. Metric construction from the edge distribution tensor

Taking into account this error analysis, we construct the metric for the unit mesh as follows:

$$\mathbb{M}^i = \left(\frac{d}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right)^{-1}.$$

For a complete justification of this result, the reader is referred to [43].

2.5. Error behavior due to varying the edge length

We examine now how the error behaves when the length of the edges changes by stretching coefficients $s_{ij} \in \mathcal{S}$ defined by :

$$\mathcal{S} = \{s_{ij} \in \mathbb{R}^+, i = 1, \dots, N, j = 1, \dots, N, \Gamma(i) \cap \Gamma(j) \neq \emptyset\}.$$

To obtain a new metric depending on the error analysis, a new length for each edge has to be calculated and then used for rebuilding the length distribution tensor. An interesting way of linking the error variations to the changes in edge lengths is by introducing a stretching factor $s \in \mathbb{R}^+$ would be

$$\begin{cases} \widetilde{\mathbf{X}}^{ij} = s \mathbf{X}^{ij} \\ \|\widetilde{e}_{ij}\| = s^2 \|e_{ij}\| = s^2 \|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}\| \end{cases} \quad (6)$$

where \widetilde{e}_{ij} and $\widetilde{\mathbf{X}}^{ij}$ are the target error at edge ij and its associated edge length respectively. Following the lines of [43] we can simply define the metric associated with \mathcal{S} by:

$$\widetilde{\mathbb{M}}^i = \frac{|\Gamma(i)|}{d} \left(\widetilde{\mathbb{X}}^i \right)^{-1}, \quad (7)$$

where $\widetilde{\mathbb{X}}^i = \frac{d}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} s_{ij}^2 \mathbf{X}^{ij} \otimes \mathbf{X}^{ij}$ is the length distribution tensor and $|\Gamma(i)|$ is the cardinality of $\Gamma(i)$. Let n_{ij} be the number of created nodes in relation with the stretching factor s_{ij} and along the edge ij . When scaling the edges by a factor s_{ij} , the error changes quadratically so that the number of created nodes along the edge ij is given by:

$$n_{ij} = \left(\frac{\widetilde{e}_{ij}}{e_{ij}} \right)^{-\frac{1}{2}} = s_{ij}^{-1}. \quad (8)$$

As per node i , the local mesh orientation and anisotropy is given by the following tensor:

$$N^i = \left(\frac{d}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} n_{ij} \frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|} \otimes \frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|} \right). \quad (9)$$

So that the local mesh density i.e. the total number of created nodes per node i is:

$$n^i = \det \left(\frac{d}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} n_{ij} \frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|} \otimes \frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|} \right). \quad (10)$$

By considering the averaging process of the number of nodes distribution function, the total number of nodes in the adapted mesh is given by:

$$N = \sum_i n^i. \quad (11)$$

A direct relation between N and e , assuming a uniform totally balanced error along the edge $\widetilde{e}_{ij} = e$ where e is a constant reads as:

$$n^{ij}(e) = s_{ij}^{-1}(e) = \left(\frac{e}{e_{ij}} \right)^{-\frac{1}{2}}. \quad (12)$$

Hence, for a node i , we have

$$n^i(e) = e^{-\frac{d}{2}} \det \left(\frac{d}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \left(\frac{1}{e_{ij}} \right)^{-\frac{1}{2}} \frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|} \otimes \frac{\mathbf{X}^{ij}}{|\mathbf{X}^{ij}|} \right) \Leftrightarrow n^i(e) = e^{\frac{d}{2}} n^i(1), \quad (13)$$

where $n^i(1)$ is the number of created nodes for a uniform error equal to 1. so that

$$N = e^{-\frac{d}{2}} \sum_i n^i(1). \quad (14)$$

Therefore, the global induced error for a given total number of nodes N can be determined by:

$$e(N) = \left(\frac{N}{\sum_i n^i(1)} \right)^{-\frac{2}{d}}. \quad (15)$$

Thus, the corresponding stretching factors under the constraint of a fixed number of nodes N are given by:

$$s_{ij} = \left(\frac{e(N)}{e_{ij}} \right)^{\frac{1}{2}} = \left(\frac{\sum_i n^i(1)}{N} \right)^{\frac{1}{d}} e_{ij}^{-1/2}. \quad (16)$$

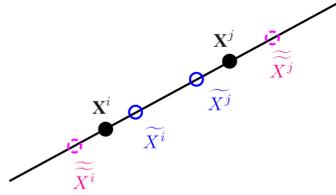


Figure 2: Varying the edge in its own direction.

2.6. Extension to multi-component field

Here we propose to construct a unique metric directly from a multi-component vectors field containing, for instance, all the components of the velocity field, its norm, the temperature field, a concentration scalar field. Consequently, we do not need to intersect several metrics but construct it using the following error vector $\vec{e}_{ij} = \{e_{ij}^1, e_{ij}^2, \dots, e_{ij}^n\}$ where n is the number of components. Let $u = \{u_1, u_2, \dots, u_n\}$ be the set of parameters based on which we want to adapt, and let $\mathcal{Z} = \mathcal{V}^n$ and $\mathcal{Z}_h = \mathcal{V}_h^n$. In the view of constructing a unique metric, the above theory is applied for each component of u . That way, we make important computational savings. It comes out immediately that the error is now a vector given by: $\vec{e}_{ij} = \{e_{ij}^1, e_{ij}^2, \dots, e_{ij}^n\}$ and then

$$s_{ij} = \left(\frac{\|e(N)\|}{\|\vec{e}_{ij}\|} \right)^{\frac{1}{2}}. \quad (17)$$

Here, the norm can be the discrete L_2 , L_1 or L_∞ norms.

3. Time adaptation procedure

Many a posteriori error estimators that can be found in the literature deal with stationary convection-diffusion problems combined with stabilized finite elements methods (see for instance [51, 52]). However, fewer papers are available for treating unsteady problems [53]. In this work, we propose a new adaptation technique that relies on equidistributing the error in space and in time. Our objective is to extend the above developed analysis to unsteady problems and to test the accuracy of the new space-time error indicator in the particular case of convection dominated regime and in presence of inner and boundary layers.

Here we consider the time dimension as a $1D$ spatial problem and apply the whole theory in $1D$. Now that we have shown how to construct a metric field at each node i of the discretized domain that would lead to an adapted anisotropic mesh in \mathbb{R}^d we move on to use the information that we obtained from this construction in order to build an adaptive time marching technique. The latter should at the same time provide accurate estimations and not require a large computational cost. Based on the information given by the derived error estimator in space and the solutions at the previous times, the algorithm automatically calculate an appropriate time-step for the following computations regardless of the properties of the problem at hand.

Denote by $\mathcal{T} = \{t^0, \dots, t^{n-1}, t^n, t^{n+1}, \dots\}$ and let $t^{n,k} = |t^n - t^k|, n, k \in \mathcal{T}$ the temporal nodes and by $\Delta t^n = t^{n,n+1}$ the time increments as shown in figure 3. Assume that the solution is already computed on the whole domain up to time t^n . The aim is to choose an appropriate time-step Δt^n .

Without loss of generality, the analysis will be carried on an arbitrary spatial node i . Note that at a

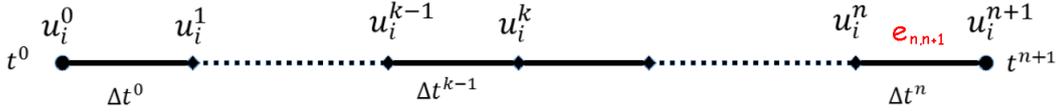


Figure 3: Temporal discretization at the spatial node i .

spatial node i , we only have one time edge to be determined (t^n, t^{n+1}) . Define $\{\tau_{n,n+1}\}$ to be the temporal edge scaling (stretching) factor such that:

$$\begin{aligned} \tilde{e}_{n+1,n} &= \tau_{n+1,n}^2 e_{n+1,n} \\ \tilde{t}^{n+1,n} &= \tau_{n+1,n} t^{n+1,n} \end{aligned} \quad (18)$$

where $e_{n+1,n}$ is an approximation of the interpolation error known as the edge based error (more details are defined later), \tilde{e} and \tilde{t} are the target error at the temporal edge $t^n t^{n+1}$ and its associated edge length.

Let u_{n-1}^i, u_n^i and u_{n+1}^i be the solutions at node i at times t^{n-1}, t^n , and t^{n+1} , respectively. Using a forward difference approximation, we have $u_{n+1} - u_n = \dot{u}_n \Delta t_n$ and $u_{n-1} - u_n = -\dot{u}_{n-1} \Delta t_{n-1}$. Then applying the $P1$ recovery gradient in $1D$, we get:

$$\dot{u}_n = \frac{u_{n,n+1}^i \Delta t_n + u_{n,n-1}^i \Delta t_{n-1}}{\Delta t_n^2 + \Delta t_{n-1}^2} \quad (19)$$

and the quadratic interpolation error is given by:

$$e_{n,n-1}^i = \dot{u}_{n,n-1}^i \Delta t_{n-1} \quad (20)$$

and

$$\tilde{e}_{n,n-1}^i = \tau_{n-1}^i \dot{u}_{n,n-1}^i \Delta t_{n-1} \quad (21)$$

where $\dot{u}_{n,n-1}^i = \dot{u}_n^i - \dot{u}_{n-1}^i$. Now using the equi-distribution error argument, we write

$$\tilde{e}_{n,n-1}^i = e_n(N) \quad (22)$$

with $e_n(N)$ being the maximal error in space for a total number of nodes N . Hence the stretching factor of the time-step size is given by:

$$\tau_{n-1}^i = \left(\frac{e(N, t_n)}{e_{n,n-1}^i} \right)^{\frac{1}{2}} \quad (23)$$

and the optimal time-step is determined by:

$$\widetilde{\Delta t}_n = \min_i \tau_{n-1}^i \Delta t_n \quad (24)$$

However if we consider this formula deeply we find out that it requires the solution at time t^{n+1} and the latter is not computed yet. Therefore instead of computing the optimal time-step Δt^n we calculate Δt^{n-1} and we let $\Delta t^n = \Delta t^{n-1}$. It is important to note that when frequency f of spatial adaptation is used, the temporal stretching factors will be modified to account for a new equidistribution. The new stretching factor is then defined by:

$$\tau_{n-1}^i = \left(\frac{e(N, t_n)/f}{e_{n,n-1}^i} \right)^{\frac{1}{2}} \quad (25)$$

Finally, it is worth mentioning that h-refinement is used, a classical linear interpolation from one mesh to the other is applied, and this work can be extended to maintain conservation as proposed in [54]. All these steps for space and time adaptation are summarized in the following algorithm:

Algorithm 1 Adaptive Algorithm

<i>Set</i> $\mathcal{T}_h^0, u_h^0, n = 1, t = 0$ and $\widetilde{\Delta t} = \Delta t$	▷ Initializations
while $t < T$ do	▷ Time loop
<i>Set</i> N	▷ Fixing the number of nodes
$t := t + \widetilde{\Delta t}_{n-1}$	▷ Time step incrementation
<i>Compute</i> u_h^n on mesh \mathcal{T}_h^{n-1}	
<i>Do for each nodes</i> X^i	
<i>Compute</i> \mathbb{X}^i	▷ The length distribution tensor
<i>Compute</i> \mathbf{G}^i using (5)	▷ Recovery gradient operator
<i>Do for all edges</i> \mathbf{X}^{ij}	
<i>Compute</i> \mathbf{G}^{ij}	▷ Recovery gradient operator
<i>Compute</i> e_{ij}	▷ Edge based error
<i>Compute</i> $e(N)$ using (15)	▷ The equidistributed error
<i>Compute</i> s_{ij}	▷ Stretching factor for each edge
<i>Compute the</i> L_2 <i>norm</i>	▷ For multi-component fields
<i>Compute</i> $\widetilde{\mathbb{X}}^i$	▷ The new length distribution tensor
<i>Compute</i> $\widetilde{\mathbb{M}}^i$ using (7)	▷ The new metric
<i>Compute</i> \mathcal{T}_h^n using MTC Mesher see [44]	▷ Mesh adaptation
<i>Interpolate</i> u_h^n on \mathcal{T}_h^n	▷ Linear Interpolation
<i>Do for each node</i> X^i	
<i>Compute</i> \dot{u}_n using (19)	▷ Temporal gradient
<i>Compute</i> $\widetilde{e}_{n,n-1}^i$ using (20)	▷ Temporal error
<i>Compute</i> $\widetilde{\Delta t}_n$ using (24)	▷ The optimal new time step
end while	

4. Convection diffusion equation

In this section the general equation of the unsteady convection diffusion equation is solved. A stabilized solution is obtained using the Streamline Upwind Petrov-Galerkin formulation.

We consider the following problem that models the transport of a quantity u through convection and diffusion:

$$\begin{cases} \partial_t u - a\Delta u + \mathbf{v} \cdot \nabla u = f, & \text{in } \Omega, \\ u(\cdot, 0) = u_0 & \text{in } \Omega, \\ u = g, & \text{on } \Gamma, \end{cases} \quad (26)$$

where $\Omega \subset \mathbb{R}^d$ is a bounded polyhedral domain with boundary Γ , for simplicity we have taken Dirichlet boundary conditions. Here a is the constant diffusion factor, $\mathbf{v}(\mathbf{x}) \in [W^{1,\infty}(\Omega)]^2$ is the divergence-free velocity field, $f(x) \in L^2(\Omega)$ is a given source term, u_0 is the initial data and g is a given boundary condition. We distinguish four types of boundary layers:

- Regular boundary layers: appear at the outflow boundary defined by $\Gamma^- = \{\mathbf{x} \in \Gamma : \mathbf{v} \cdot \mathbf{n} < 0\}$ where the velocity field \mathbf{v} is not parallel to the boundary. Their width is of $\mathcal{O}(a)$. This order results from the difference between the solution of the convection-diffusion problem (26) and the only diffusion one (when $a = 0$).
- Parabolic boundary layers: appear at the characteristic boundary defined by $\Gamma^0 = \{\mathbf{x} \in \Gamma : \mathbf{v} \cdot \mathbf{n} = 0\}$ where the velocity field \mathbf{v} is parallel to the boundary. Their width is thicker than that of the regular boundary layers and is of $\mathcal{O}(\sqrt{a})$.
- Corner boundary layers: appear at the neighborhood of a corner of the domain i.e. at the intersection of boundary layers.
- Interior layers: appear at the inflow boundary $\Gamma^+ = \{\mathbf{x} \in \Gamma : \mathbf{v} \cdot \mathbf{n} > 0\}$ due to the discontinuities in the data that propagate inside the domain according to the velocity field \mathbf{v} . Their width is of the same order of that of the parabolic boundary layers i.e. $\mathcal{O}(\sqrt{a})$.

4.1. Galerkin finite element formulation

Let us consider the functional Sobolev space of functions having square integrable first order derivatives $H_S^1(\Omega)$ in which we are searching the solution in accordance with its regularity:

$$H_s^1 = \{w \in H^1(\Omega) \mid w = s \forall x \in \partial\Omega\} \quad (27)$$

with

$$H^1(\Omega) = \{w \in L^2(\Omega), \|\nabla w\| \in L^2(\Omega)\} \quad (28)$$

and L^2 is the Hilbert vector space given by:

$$L^2(\Omega) = \left\{ w(x) \mid \int_{\Omega} |w|^2 dx < \infty \right\}$$

The Galerkin finite element formulation is obtained by multiplying Eq. (26) by an appropriate test function $w \in H_0^1(\Omega)$ and by integrating over the computational domain. The discrete problem becomes: find $u \in H^1(\Omega)$ such that $u = g$ on Γ and

$$\left(\frac{\partial u}{\partial t}, w \right) + \mathcal{B}(u, w) = \mathcal{F}(w) \quad \forall w \in H_0^1(\Omega) \quad (29)$$

where $(u, w) = \int_{\Omega} uw \, d\Omega$ and $\begin{cases} \mathcal{B}(u, w) = a(\nabla u, \nabla w) + (\mathbf{v} \cdot \nabla u, w) \\ \mathcal{F}(w) = (f, w) \end{cases}$.

Challenge arises when convection dominates diffusion i.e. when $a \ll \|\mathbf{v}\|$. Note that this case is in accordance with a Peclet number ($Pe = \frac{\|\mathbf{v}\|h}{2a}$) much greater than one. In this case, the standard Galerkin finite element discretization, if the mesh is not well refined, gives rise to node-to-node oscillations in the solution. One way to eliminate these oscillations is the use of upwind techniques such as the stabilized finite element method.

4.2. Streamline Upwind Petrov-Galerkin method

For the spatial discretization, we consider the finite element partition \mathcal{T}_h of Ω into tetrahedral elements K . Using these representations, the above-defined functional spaces $H_s^1(\Omega)$ and $H_0^1(\Omega)$ are approached by discretized spaces $H_s^{1h}(\Omega)$ and $H_0^{1h}(\Omega)$. Let $V_h \subset H^1(\Omega)$ be the space of piecewise linear functions defined on the elements of \mathcal{T}_h and $g_h \in V_h$ be a piecewise linear interpolation of g on \mathcal{T}_h . The SUPG approximation of (29) is formulated on these finite-dimensional subspaces by modifying the standard Galerkin weighting functions w_h for all terms in the equation into $w_h + \tau_K \mathbf{v} \cdot \nabla w_h$. This modification is interpreted by allowing more weight to the node in the upstream direction and reducing the weight to the node in the downstream direction. The variational formulation becomes:
find $u_h \in V_h$ such that $u_h = g_h$ on Γ and

$$\left(\frac{\partial u_h}{\partial t}, w_h \right) + \mathcal{B}_\tau(u_h, w_h) = \mathcal{F}_\tau(w_h) \quad \forall w_h \in H_0^1(\Omega) \quad (30)$$

where τ_K is a stabilization parameter and
$$\begin{cases} \mathcal{B}_\tau(u_h, w_h) = \mathcal{B}(u_h, w_h) + \sum_{K \in \mathcal{T}} \tau_K (-a \Delta u_h + \mathbf{v} \cdot \nabla u_h, \mathbf{v} \cdot w_h)_K \\ \mathcal{F}_\tau(w_h) = \mathcal{F}(w_h) + \sum_{K \in \mathcal{K}} \tau_K (f, \mathbf{v} \cdot \nabla w_h)_K \end{cases}$$

Comparing equations (29) and (30) we notice that the SUPG formulation introduce a diffusion along the streamline direction weighted by a stabilization parameter. As the additional stabilizing terms vanish for the exact solution of the problem, the SUPG discretization is consistent.

Finally, the problem defined by equation (30) yields the system of first order differential equations:

$$C \frac{\partial U}{\partial t} + \mathcal{K}U = \mathcal{F} \quad (31)$$

where U is the vector of nodal unknown values of the solution.

4.3. Time integration scheme

The system of ordinary differential equations (31) has to be integrated in time. Using the θ time discretization schemes, the derivative of the solution with respect to time can be approximated at time t^n by:

$$C \frac{U^n - U^{n-1}}{\Delta t} + \mathcal{K}(\theta U^n + (1 - \theta)U^{n-1}) = \mathcal{F} \quad (32)$$

where Δt is the time-step, $n = 1, \dots, N$, and $0 \leq \theta \leq 1$. We remind that this family includes the backward Euler scheme ($\theta = 1$), the Crank-Nicolson scheme ($\theta = 0.5$) and the forward Euler scheme ($\theta = 0$).

4.4. Choice of the stabilizing parameter

The stabilizing parameter τ_K will govern the amplitude of the added artificial diffusion in the direction of the streamline as it determines and calibrates the amount of upwinding weights locally in each element. In the literature, we can find many references [1, 2, 55, 56, 57, 58] concerning the choice of the stabilizing parameters.

A theoretical analysis of convection diffusion problems, in one space dimension and on a uniform mesh, has shown in [51, 59] that the SUPG solution is nodally exact for continuous piecewise linear finite elements if

$$\tau_K = \frac{h_K}{2\|\mathbf{v}\|_2} \left(\coth(\text{Pe}_K) - \frac{1}{\text{Pe}_K} \right) \quad (33)$$

where $\text{Pe}_K = \frac{\|\mathbf{v}\|_2 h_K}{2|a|}$ is the mesh Peclet number and h_K is the element size.

Another possible suggestion for the stabilizing parameter taken from [1] is:

$$\tau_K = \frac{h_K}{2\|\mathbf{v}\|_2} \max \left\{ 0, 1 - \frac{1}{\text{Pe}_K} \right\} \quad (34)$$

Experimental studies have shown that in a convection dominated regime for large Peclet numbers this choice of the stabilizing parameters gives more stable numerical results. Note that τ_K is positive for both convection and diffusion dominated problems with:

$$\tau_K = \begin{cases} c_1 h_K & \text{for } \text{Pe}_K > 3, \\ c_2 \frac{h_K^2}{a} & \text{for } \text{Pe}_K \leq 3 \end{cases} \quad (35)$$

where c_1 and c_2 are positive constants.

Note that the calculation of h_K is one of the main subjects of this paper in particular when using anisotropic mesh adaptation. Indeed, the stability coefficients depend on the local mesh size h_K and weight the extra terms added to the weak formulation. In the case of strongly anisotropic meshes with highly stretched elements, the definition of h_K is still an open problem and plays a critical role in the design of the stabilizing coefficients. We followed the lines in [60] to compute h_K as the diameter of K in the direction of the velocity \mathbf{v} as follows:

$$h_K = \frac{2|\mathbf{v}|}{\sum_{i=1}^{N_K} |\mathbf{v} \cdot \nabla \varphi_i|} \quad (36)$$

where N_K is the number of vertices of K and $\varphi_1, \dots, \varphi_{N_K}$ are the usual basis functions of $P_1(K)$ mapped onto K .

4.5. Existence, Uniqueness and order of convergence

The existence and the uniqueness of the solution obtained using the SUPG method can be studied using the Lax-Milgram theorem. Note that in convection dominated problems, the standard Galerkin formulation fails to satisfy the coercivity condition leading to instabilities in the solution. We define the energy norm as $|||w|||_2^2 = a||\nabla w||_2^2 + \tau||w||_2^2$. For this norm, the linear functional $F_\tau(\cdot)$ is continuous and the bilinear form $B(\cdot, \cdot)$ satisfies:

$$B(u, u) \geq |||u|||^2 \quad \forall u \in H_0^1(\Omega) \quad (37)$$

$$B(u, w) \leq |||u||| \cdot \left(a^{-1/2} \|v\|_\infty \|w\| \right) \quad (38)$$

and thus is coercive and continuous with respect to the energy norm.

Theorem: If linear approximation is used on a finite element triangulation, then there exists a constant C , asymptotically proportional to the mesh Peclet number with:

$$|||\nabla(u - u_h)||| \leq C h_K^{\max} |||u|||_{H^2(\Omega)} \quad (39)$$

Theorem: If piecewise linear approximations is used on a uniform finite element triangulation with $h > 2a$, then there exists a bounded constant C , independent of the diffusion coefficient with:

$$|||u - u_h||| \leq C (h_K^{\max})^{3/2} |||u|||_{H^2(\Omega)} \quad (40)$$

The error bounds in the L^2 norm show a loss in the order of convergence by half-an-order for the SUPG method which reflects that in the presence of steep layers and convection dominated regimes it is difficult to achieve optimal convergence rates. Accuracy and convergence improvements can be obtained by resorting to anisotropic mesh adaptations.

5. Numerical examples

The main objective of this paper is to provide a method that can be used to simulate diverse industrial applications. For practical reasons and as noted above, it is highly desirable to find the most accurate

solution using the available computational resources. In this section, we assess the performance of the newly developed space and time adaptive algorithms on several time dependent problems. We first examine the order of convergence of the numerical method using a well-known inner layers test case. The following two examples deal with dynamic properties of the space-time adaptive procedure. In these examples, we present the results in problems with boundary layers and internal layers that move through the domain and compare them to [53]. Finally, the last two examples present coupled problems showing the ability of the proposed method to automatically detect and reflect evolving/emerging layers, to produce oscillating free solutions and to reduce the computational costs of a 3D large scale simulation.

5.1. Application to the steady convection-diffusion equation

We first consider a well-known steady test case [52, 38] with interior layer and variable convective field. We set the computational domain $\Omega = (0, 1)^2$ and a diffusion coefficient $a = 0.005$. We choose the exact solution to be continuous all over the domain but to develop an interior layer along the line $y = -x + 0.85$:

$$u(x, y) = \left(1 + e^{\frac{x+y-0.85}{2a}}\right)^{-1}$$

The boundary conditions and the source term are derived from the exact solution. We consider a variable velocity field $\mathbf{v} = (u(x, y), u(x, y))^T$. Figure 4(a) presents the numerical solution obtained on a mesh made up of around 4,000 elements. The obtained anisotropic mesh is shown in figure 4(b). One can notice the high concentration of the elements along the line $y = -x + 0.85$ in order to accurately capture the inner layer. Taking a closer look at the mesh in the zoomed version near the left boundary we can observe the good orientation of the elements with the stretching in the direction of the sharp layer. This demonstrates the ability of the algorithm to work under the constraint of a fixed number of nodes and to effectively control the elements' sizes, orientations and locations. This example aims at emphasizing the spatial order

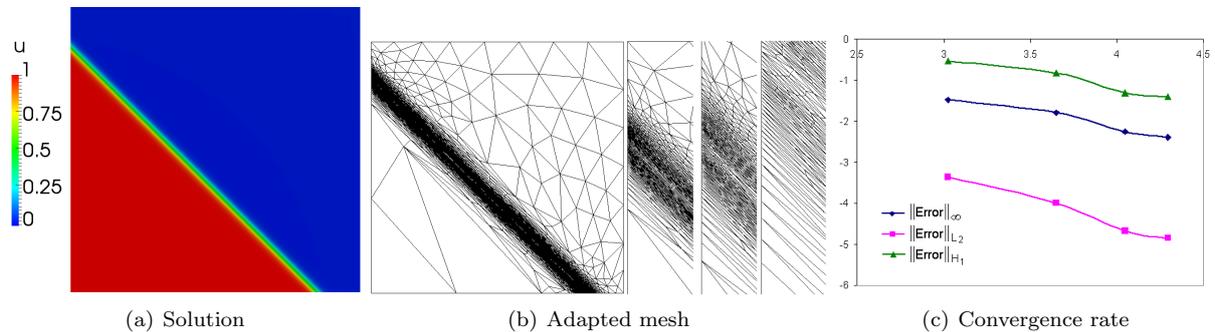


Figure 4: Numerical solution, corresponding mesh and L_{∞} , L^2 and H^1 norms of the error versus the number of elements in the mesh for $a = 0.005$.

of convergence when using the mesh adaptation technique. The global convergence order is computed in the L_{∞} , L^2 and H^1 norms. As can be seen in figure 4(c) the convergence is of first order in the H^1 , L^{∞} norms and of second order in the L_2 norm recovering the theoretical predictions. The very small errors in the L_2 norm reflect the advantage of the adapted mesh as it is well aligned with the solution that has strong anisotropic features.

5.2. Convection-diffusion in a plane shear flow

We consider a problem proposed in [61] and revisited by [53] using a space-time adaptive algorithm. It models the transport of a small source in a plane shear flow. The computational domain is set to $\Omega = (0, 24000) \times (-34000, 34000)$. We consider zero source term, a small diffusion coefficient $k = 1$, a final

time $T = 9000$ and a convective field $v = (v_0 + \lambda y, 0)^t$ where $a_0 = 0.5$ and $\lambda = 1e^{-3}$. The initial condition is a point source with mass m at $(x_0, y_0) = (7200, 0)$. The analytical solution of the problem is given by:

$$u(x, y, t) = \frac{m}{4\pi\epsilon t \left(1 + \lambda^2 \frac{t^2}{12}\right)^{1/2}} \exp^{-\zeta}$$

where

$$\zeta = \frac{(x - \bar{x} - \lambda y t / 2)^2}{4\epsilon t \left(1 + \lambda^2 \frac{t^2}{12}\right)} + \frac{y^2}{4\epsilon t} \quad \text{and } \bar{x} = x_0 + a_0 t.$$

We start the numerical resolution at time $t = 2400$ with

$$m = 4\pi\epsilon t_0 \left(1 + \lambda^2 \frac{t_0^2}{12}\right)^{1/2}$$

The objective of this test case is to assess and compare the performance of the developed adaptation technique with the one proposed in [53]. Therefore, we repeated the simulations using the same number of nodes.

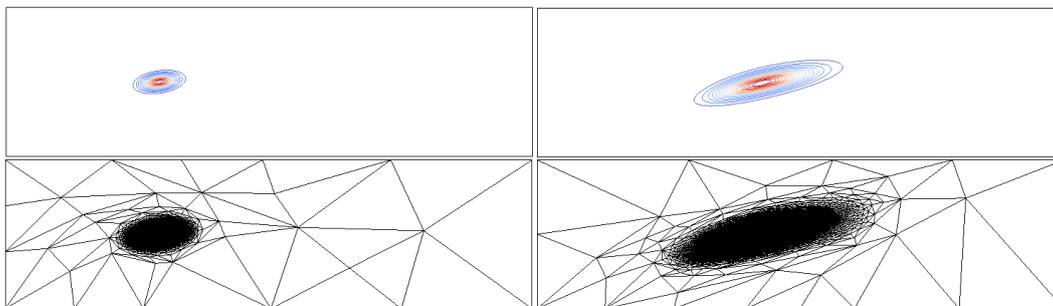


Figure 5: The obtained isovalues and the adapted meshes at two different time step.

Figure 5 presents the isovalues of the temperature field and the adapted mesh for a fixed number of nodes 8000. As shown in [53], we point out how well the mesh is stretched in the direction of high gradients of the solution. The corresponding history of the L_2 norm of the approximation error and the time-step size evolution are provided in Table 1.

Number of nodes	$\epsilon_{\text{present work}}$	$\epsilon_{\text{reference}}$	$N_{\tau \text{ present work}}$	$N_{\tau \text{ reference}}$
1679	0.04556	0.116	27	159
3728	0.02303	0.0551	37	206
11990	0.00970	0.0265	58	281
40525	0.00149	0.0129	160	400

Table 1: The error and the number of time steps of the adapted solution at final time.

We can see how the time-step size increases as the solution is advected and diffused. We note that the profile of the time-step variations is similar to the one presented in [53] with a better control of the L_2 norm of the error for almost the same number of nodes. Finally, for almost the same number of nodes, we achieve a better control of the approximation error within a fewer number of time-steps.

Finally, we emphasize in Figure 6 that the optimal second order convergence is reached with respect to the number of nodes and to the number time-steps. In accordance with [53], we have the same order of convergence with fewer time-steps and a lower error which proves the efficiency and accuracy of the proposed technique.

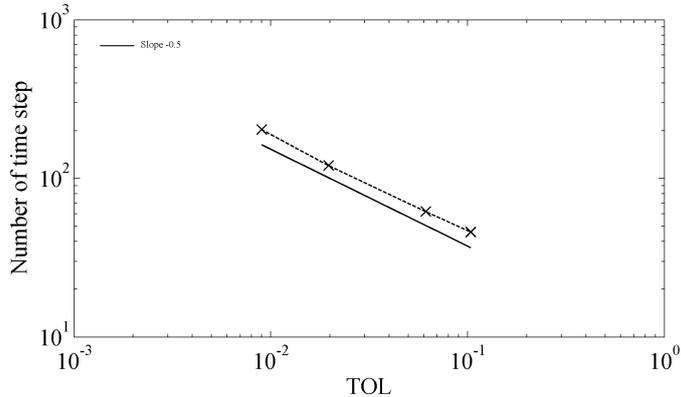


Figure 6: Total number of time steps with respect to the tolerance.

5.3. Internal and boundary layers

We present a second test case, taken from [53], with challenging anisotropic features, exhibiting internal and boundary layers. The convective field is considered to be constant $v = (2, 1)^t$ in the whole computational domain $\Omega = (0, 1)^2$. The diffusion coefficient is set to $a = 10^{-3}$ and zero source term is assumed. The initial condition is zero everywhere except at the boundary where the boundary condition is defined by:

$$u(x, y) = \begin{cases} 1 & \text{if } \{x = 0, 0 \leq y \leq 1\} \cup \{0 \leq x \leq 1, y = 1\}, \\ \frac{\delta - x}{\delta} & \text{if } \{x \leq \delta, y = 0\}, \\ \frac{y - 1 + \delta}{\delta} & \text{if } \{x = 1, y \geq 1 - \delta\}, \\ 0 & \text{if } \{x > \delta, y = 0\} \cup \{x = 1, y \leq 1 - \delta\}. \end{cases}$$

As time advances, the gradient of temperature at the left boundary spreads into the domain creating an internal layer that reaches the right wall resulting in a boundary layer. On the other hand, the discontinuity at the top wall reduces with time. Figure 7 shows the evolution of the temperature over time together with the corresponding anisotropically adapted mesh. A localized refinement of the mesh where the solution exhibits steep layers can be clearly observed. To keep up with the fixed number of nodes, the mesh is automatically coarsened at the locations of lower solution gradients. The plots also reflect how well the mesh follows the propagation of the solution with time.

Note the importance of this test case to reflect the potential of the developed algorithm. If non adaptive mesh is used, a very fine mesh size would be necessary everywhere inside the domain and a very small time-step would be required to correctly capture the layers' propagation along the whole domain.

Figure 8 shows the history of the time-step sizes that responds very well to the solution's profile. We notice that the algorithm starts by generating small time-steps in order to allow a good capture of the solution's gradients. As the temperature diffuses, the size is progressively increased then as it stabilizes (around the time $t = 0.5s$), significant increase of the time-steps is observed.

We have reported in Table 2 the obtained error and the total number of time steps for several number of nodes N . As observed in the previous example, we achieve a better control of the approximation error with fewer time-steps. Figure 9 shows that the optimal second order of convergence with respect to the time step is again recovered as the number of time step is multiplied by $\sqrt{2}$ when the user's specified tolerance (TOL) is divided by two.

5.4. Application to coupled heat transfer and fluid flows

The last two examples deal with 2D and 3D coupled heat transfer and fluid flows problems. We solved simultaneously the Navier-Stokes equations along with the heat transfer convection-diffusion equation. Therefore, the adaptivity takes into account different fields: $\frac{T}{T_\infty}$, $\frac{v}{\|v\|_2}$, $\frac{\|v\|_2}{\|v\|_\infty}$. For the sake of simplicity, we will

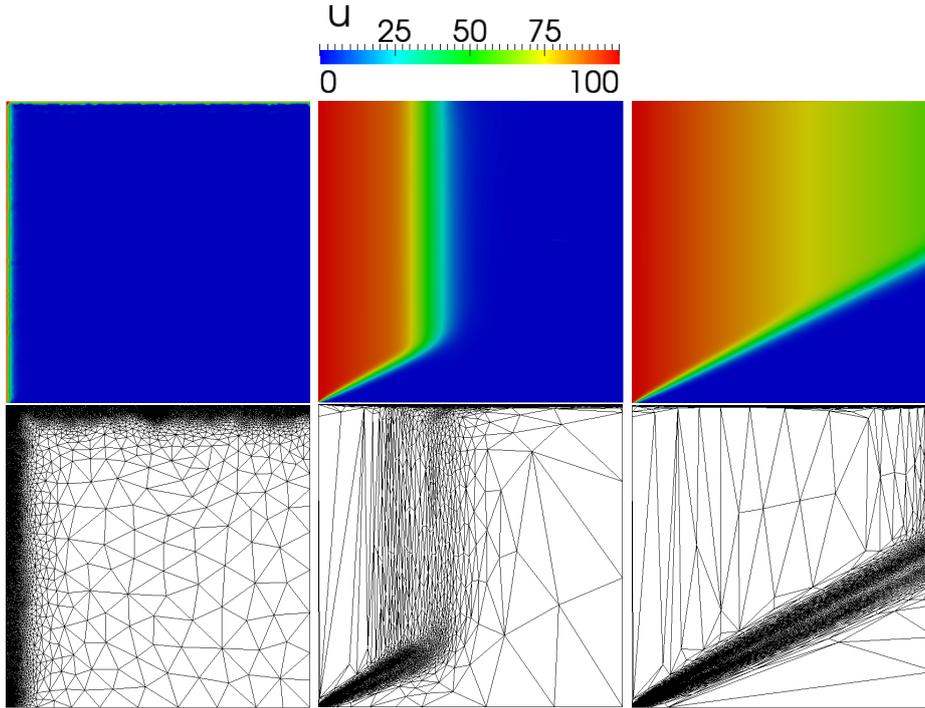


Figure 7: Numerical solutions at different time-steps with their corresponding anisotropically adapted meshes.

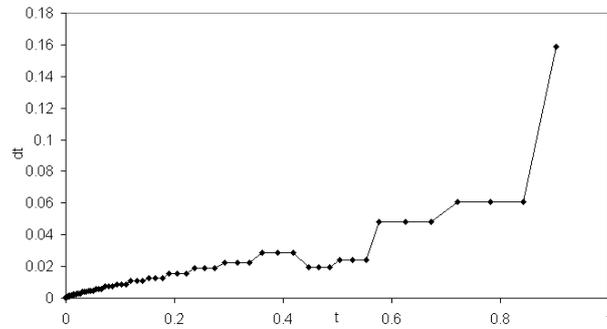


Figure 8: Evolution of the time-step size over time.

Number of nodes	$\epsilon_{\text{present work}}$	$\epsilon_{\text{reference}}$	$N_{\mathcal{T}}$ present work	$N_{\mathcal{T}}$ reference
3987	0.10369	0.25	46	163
12222	0.06134	0.125	62	247
38874	0.01964	0.0625	121	353
140057	0.00900	0.03125	204	502

Table 2: The error and the number of time steps of the adapted solution at final time.

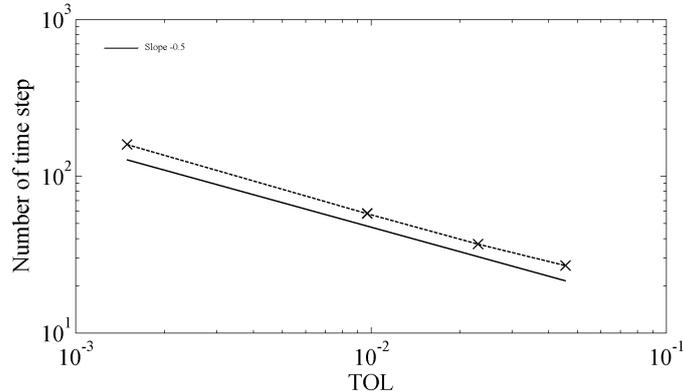


Figure 9: Total number of time steps with respect to the tolerance.

not discuss here the discretization of the Navier-Stokes equations and we kindly ask the reader to refer to [62] for more details.

5.4.1. Forced convection inside a cavity

We consider a 2D forced convection problem. This test case can be regarded as a simplified model of a gas-fired furnace or a temperature distribution inside a room.

Heated air is pumped into the enclosure from an inlet at the left wall with a velocity of magnitude $1m/s$ and a temperature of $100K$. Adiabatic condition for the temperature is applied at all other boundaries. The air is vented out of the enclosure through the outlet positioned at the right wall. The computations were run on a period of $10,000s$.

We have considered two different viscosity values ($\mu = 0.01$ and $\mu = 0.0005$) to test the performance of the adaptive algorithms for quasi-static and unsteady flows. We have also varied the parametric components of the vector field used to adapt the mesh. The aim of this test case is first to select the best set of parameters for meshing and second to validate the efficiency of the space and time adaptive techniques. The numerical results have shown that the vector field taking into account the velocity and the temperature i.e. $\frac{T}{T_\infty}, \frac{v}{\|v\|_2}, \frac{\|v\|_2}{\|v\|_\infty}$ gives the best results. Figures 10 and 11 show the obtained approximated temperature and velocity's magnitude and their corresponding mesh. Note the concentration of the elements not only along the boundary layers but also around the emerging vortices and the detachment regions. This reflects how, for a fixed number of nodes equal to $20,000$, the mesh is automatically coarsened in regions with low gradient of the solution and is refined in regions of high gradients. Also note the alignment between the direction of the gradient and the direction of refinement which allows an accurate capture of the anisotropy of the physical phenomena.

We have tested the time-step algorithm with the different possible combinations for the adaptation vector field. Figures 12(a) and 12(b) provide the evolution of the time-step sizes. The profiles of the time-step, for the different combinations, are as expected in the case where the solution stabilizes after a certain time. However, it can be noticed from the plot corresponding to the unsteady case that when adapting with respect to the velocity field only, the time-step grows significantly and in turn the resulting numerical solution is never stable. However, when adapting on the temperature alone and on the combination of both the temperature and the velocity field, the time-step size oscillates around a small value allowing a good capture of the solution's details. This highlights once more the capability of the algorithm in producing automatic and controlled time steps.

5.5. Application to 3D heat transfer and turbulent flow inside an industrial furnace

This test case is dedicated to the simulation of a 3D industrial furnace. The idea is to assess the capability of the proposed algorithms to deal with a complex 3D geometry and to handle simulating long time heating

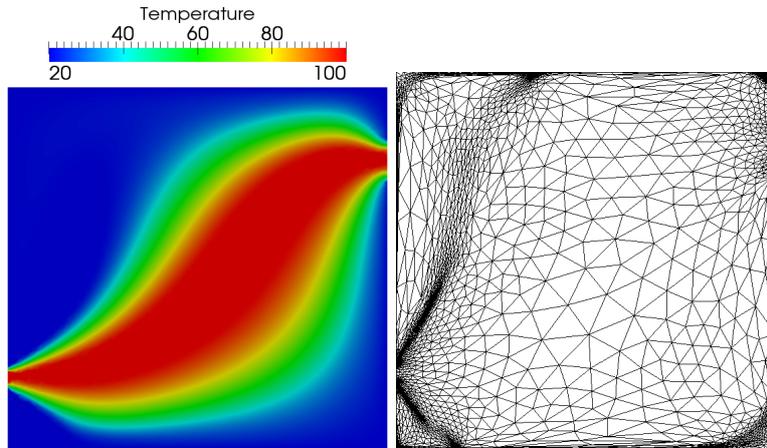


Figure 10: Temperature and velocity profiles with the corresponding adapted mesh for $\mu = 0.01$.

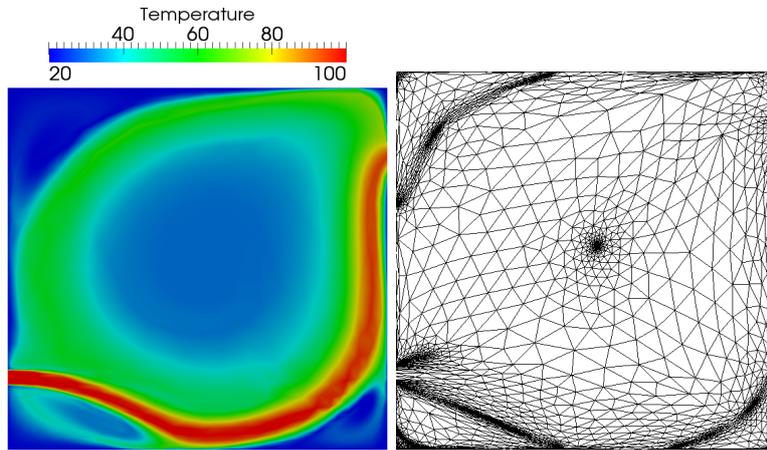
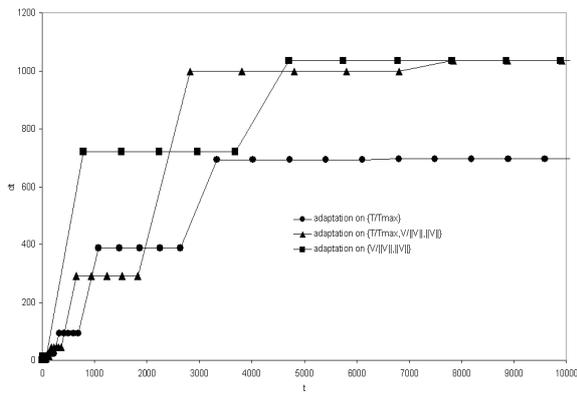
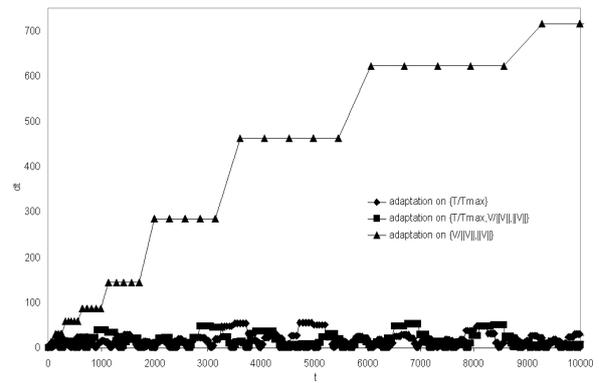


Figure 11: Temperature and velocity profiles with the corresponding adapted mesh for $\mu = 0.0005$.



(a) $\mu = 0.01$



(b) $\mu = 0.0005$

Figure 12: Time-step evolution for the different possible combinations of the adaptation vector field.

inside large scale furnaces.

The furnace is modeled as a hexagonal section duct of $2.7 \times 8.1 \times 5.3m^3$ forming one heat transfer zone. All computations have been conducted by starting with a gas at rest with a constant temperature of $1000C$. Adiabatic temperature is considered at all other boundaries for sake of simplicity. The heated air is pumped into the furnace at constant speed of $14m/s$ by a circular burner and located at the left vertical wall. The air is vented out of the furnace through two outlets positioned at the bottom of the left vertical wall. The 3D computations aim at simulating an hour of heating and have been conducted in parallel on 32, $2.4Ghz$ Opteron cores.

Figure 13 (top) shows the isothermal distribution at different time-steps. When the hot fluid spreads along the volume of the furnace, it induces a turbulent motion within the geometry. This forced convection is caused by the interaction of the moving stream and the stationary fluid inside the furnace. The numerically obtained temperature distribution clearly reflects the expected flow pattern. A number of small vortexes inside different buffer zones can be observed. The latter are due to the turbulence dissipation and mixing of the hot and cold air.

Figure 13 (bottom) highlights how well the mesh is adapted to capture the gradients of the solution, the boundary layers and the emerging vortexes. All the boundary layers as well as the vortexes are sharply captured and identified. The obtained meshes show the stretching of the elements and the high resolution near the walls, in the corners and at the location of high gradients. The anisotropic adaptive procedure modifies the mesh, according to the fixed number of nodes (100,000), in such a way that the local density is adequately distributed in all directions. Recall again that the algorithm builds up the mesh in a way to maximize the accuracy of the numerical solution. Note that for such forced convection, the mesh is adapted only according to the velocity components.

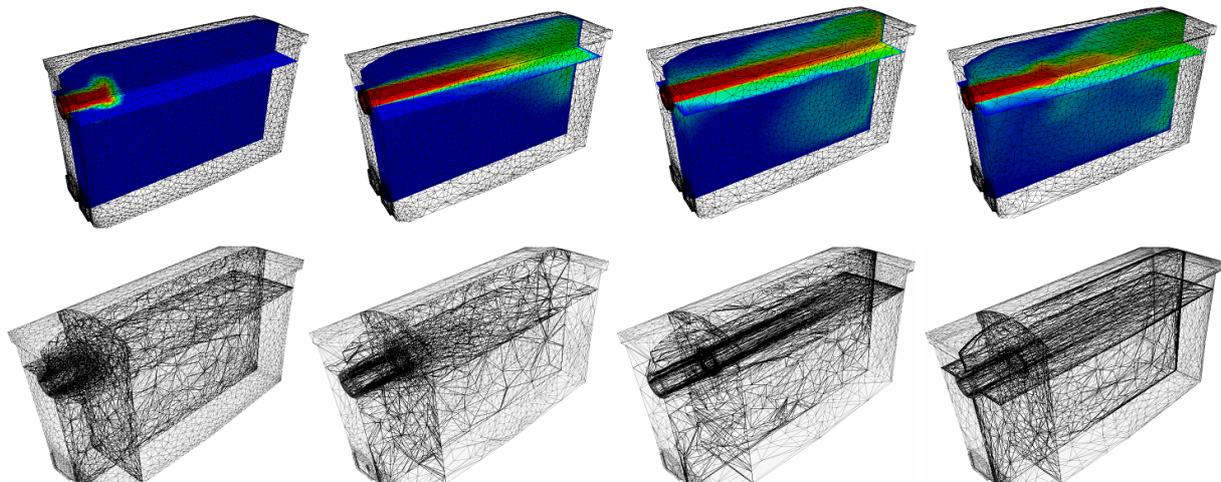


Figure 13: Isotherms inside the furnace (Top) and corresponding adapted meshes (bottom) at three different time-steps.

The results describing only one hour of the heating process required 100 days of computations on 32 cores with a fixed time-step equal to $0.005s$. This points out the necessity to supply fast algorithms capable of reproducing the full heating sequences in a reasonable time. Significant CPU time and computational cost were saved by applying the time adaptive procedure as it required only 2 days of calculations (see table 3).

6. Conclusion

We proposed in this paper a new adaptive time-step technique combined with anisotropic mesh adaptation for the solution of unsteady convection-dominated problems. A Streamline Upwind Petrov-Galerkin

	CPU Time (s)
Non-Adaptive $\Delta t = 0.005s$	8,640,000
Adaptive Δt	172,800
Ratio	50

Table 3: CPU time required for computing the solution with non-adaptive and a space/time adaptive methods

(SUPG) finite element method is presented and used to solve the unsteady convection-diffusion equation. An a posteriori error estimation based on the length distribution tensor approach and the associated edge based error estimator is applied to the stabilized finite element solution detecting automatically all sharp gradients, inner and boundary layers. A new time error estimator is derived and combined with the space adaptivity to provide automatically an appropriate time-step. Combining a stabilized finite element method of the SUPG type with the space and time adaptivity has shown to be favourable for simulating convection-dominated problems. All the meshes are obtained by solving an optimization problem under the constraint of a fixed number of nodes in the mesh. Several numerical 2D and 3D test cases are presented and show that the use of both space and time adaptivity allows the recovery of the global convergence order of the numerical schemes, reduces the computational time and cost and produces accurate and oscillation free numerical solutions.

Acknowledgment

This work was partly done in the MAIDESC ANR project which is supported by the French ministry of Research under contract ANR-13-MONU-0010.

References

- [1] A. N. Brooks, T. J. 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 (1-3) (1982) 199 – 259.
- [2] T. J. Hughes, M. Mallet, A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems, *Computer Methods in Applied Mechanics and Engineering* 58 (3) (1986) 305 – 328.
- [3] G. Scovazzi, A discourse on Galilean invariance, SUPG stabilization, and the variational multiscale framework, *Computer Methods in Applied Mechanics and Engineering* 196 (4-6) (2007) 1108–1132.
- [4] I. Harari, T. Hughes, Stabilized finite element methods for steady advection-diffusion with production, *Computer Methods in Applied Mechanics and Engineering* 115 (1994) 165–191.
- [5] I. Harari, T. J. Hughes, Galerkin/least-squares finite element methods for the reduced wave equation with non-reflecting boundary conditions in unbounded domains, *Computer Methods in Applied Mechanics and Engineering* 98 (3) (1992) 411 – 454.
- [6] I. Harari, T. Hughes, Stabilized finite element methods for steady advection-diffusion with production, *Computer Methods in Applied Mechanics and Engineering* 115 (1994) 165–191.
- [7] L. F. T. Hughes, G. Hulbert, A new finite element formulation for fluid dynamics: VIII. The Galerkin/least-squares method for the advection-diffusive equations., *Comput. Methods Appl. Mech. Engrg.* 73 (1989) 173–189.
- [8] L. Franca, F. C., Bubble functions prompt unusual stabilized finite element methods, *Computer Methods in Applied Mechanics and Engineering* 123 (1995) 229–308.
- [9] L. Franca, C. Farhat, A. Macedo, M. Lesoinne, Residual-Free Bubbles for the Helmholtz Equation, *International Journal for Numerical Methods in Engineering* 40 (1997) 4003–4009.
- [10] L. Franca, C. Farhat, M. Lesoinne, A. Russo, Unusual Stabilized Finite Element Methods and Residual-Free Bubbles, *International Journal for Numerical Methods in Fluids* 27 (1998) 159–168.
- [11] E. Hachem, H. Digonnet, E. Massoni, T. Coupez, Enriched finite element spaces for transient conduction heat transfer, *Applied Mathematics & Computations* 217 (2010) 3929–3943.
- [12] R. Codina, Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods, *Computer Methods in Applied Mechanics and Engineering* 190 (13-14) (2000) 1579–1599.
- [13] D. Kavetski, P. Binning, S. W. Sloan, Adaptive backward Euler time stepping with truncation error control for numerical modelling of unsaturated fluid flow, *International Journal for Numerical Methods in Engineering* 53 (6) (2002) 1301–1322.
- [14] S. W. Sloan, A. J. Abbo, Biot consolidation analysis with automatic time stepping and error control Part 1: theory and implementation, *International Journal for Numerical and Analytical Methods in Geomechanics* 23 (6) (1999) 467–492.

- [15] Y. T. Feng, D. Peric, A time-adaptive space-time finite element method for incompressible Lagrangian flows with free surfaces: computational issues, *Computer Methods in Applied Mechanics and Engineering* 190 (5-7) (2000) 499 – 518.
- [16] Z. Chen, J. Feng, An Adaptive Finite Element Algorithm with Reliable and Efficient Error Control for Linear Parabolic Problems, *Mathematics of Computation* 73 (247) (2004) 1167–1193.
- [17] S. E. Minkoff, N. M. Kridler, A comparison of adaptive time stepping methods for coupled flow and deformation modeling, *Applied Mathematical Modelling* 30 (9) (2006) 993 – 1009.
- [18] S. Berrone, M. Marro, Space-time adaptive simulations for unsteady Navier-Stokes problems, *Computers & Fluids* 38 (6) (2009) 1132 – 1144.
- [19] K. Georgiev, N. Kosturski, S. Margenov, J. Star, On adaptive time stepping for large-scale parabolic problems: Computer simulation of heat and mass transfer in vacuum freeze-drying, *Journal of Computational and Applied Mathematics* 226 (2) (2009) 268 – 274.
- [20] S. E. Minkoff, N. M. Kridler, A comparison of adaptive time stepping methods for coupled flow and deformation modeling, *Applied Mathematical Modelling* 30 (2006) 993–1009.
- [21] S. Micheletti, S. Perotto, Space-time adaptation simulation for purely diffusive problems in an anisotropic framework, *International Journal of Numerical Analysis and Modeling* 7 (1) (2010) 125 – 155.
- [22] K. J. Fidkowski, Y. Luo, Output-based space-time mesh adaptation for the compressible Navier-Stokes equations, *Journal of Computational Physics* 230 (14) (2011) 5753 – 5773.
- [23] Y. Zhang, Y. Hou, H. Zuo, A posteriori error estimation and adaptive computation of conduction convection problems, *Applied Mathematical Modelling* 35 (2011) 2336–2347.
- [24] J. Peraire, M. Vahdati, K. Morgan, O. Zienkiewicz, Adaptive remeshing for compressible flow computations, *Journal of Computational Physics* 72 (2) (1987) 449 – 466.
- [25] V. Selmin, L. Formaggia, Simulation of hypersonic flows on unstructured grids, *International Journal for Numerical Methods in Engineering* 34 (2) (1992) 569–606.
- [26] R. Lohner, Adaptive remeshing for transient problems, *Computer Methods in Applied Mechanics and Engineering* 75 (13) (1989) 195 – 214.
- [27] O. C. Zienkiewicz, J. Wu, Automatic directional refinement in adaptive analysis of compressible flows, *International Journal for Numerical Methods in Engineering* 37 (13) (1994) 2189–2210.
- [28] A. Tam, D. Ait-Ali-Yahia, M. Robichaud, M. Moore, V. Kozel, W. Habashi, Anisotropic mesh adaptation for 3D flows on structured and unstructured grids, *Computer Methods in Applied Mechanics and Engineering* 189 (4) (2000) 1205 – 1230.
- [29] C. Pain, A. Umpleby, C. de Oliveira, A. Goddard, Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations, *Computer Methods in Applied Mechanics and Engineering* 190 (2930) (2001) 3771 – 3796.
- [30] C. Gruau, T. Coupez, 3D Tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric, *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 4951–4976.
- [31] P. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 5068–5082.
- [32] G. Kunert, An a posteriori residual error estimator for the finite element method on anisotropic tetrahedral meshes, *Numerische Mathematik* 86 (3) (2000) 471–490.
- [33] L. Formaggia, S. Micheletti, S. Perotto, Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection-diffusion-reaction and the Stokes problems, *Applied Numerical Mathematics* 51 (4) (2004) 511 – 533.
- [34] S. Micheletti, S. Perotto, Reliability and efficiency of an anisotropic ZienkiewiczZhu error estimator, *Computer Methods in Applied Mechanics and Engineering* 195 (912) (2006) 799 – 835.
- [35] 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 (13) (2006) 14 – 23.
- [36] F. Hecht, R. Kuate, An Approximation of Anisotropic Metrics from Higher Order Interpolation Error for Triangular Mesh Adaptation, *J. Comput. Appl. Math.* 258 (2014) 99–115, ISSN 0377-0427.
- [37] L. Formaggia, S. Perotto, New anisotropic a priori error estimate, *Numer. Math.* 89 (2001) 641–667.
- [38] W. Huang, Metric tensors for anisotropic mesh generation, *J. Comput. Phys.* 204 (2) (2005) 633–665.
- [39] D. Venditti, D. L. Darmofal, Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows, *Journal of Computational Physics* 187 (1) (2003) 22 – 46.
- [40] S. Micheletti, S. Perotto, Output Functional Control for Nonlinear Equations Driven by Anisotropic Mesh Adaption: The Navier-Stokes Equations, *SIAM Journal on Scientific Computing* 30 (6) (2008) 2817–2854.
- [41] F. Alauzet, W. Hassan, M. Picasso, Goal oriented, anisotropic, A posteriori error estimates for the Laplace equation, in: *Numerical Mathematics and Advanced Applications 2009*, Springer Berlin Heidelberg, 47–58, 2010.
- [42] J. Peter, M. Nguyen-Dinh, P. Trontin, Goal oriented mesh adaptation using total derivative of aerodynamic functions with respect to mesh coordinates With applications to Euler flows, *Computers & Fluids* 66 (0) (2012) 194 – 214.
- [43] 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.
- [44] T. Coupez, A mesh improvement method for 3D automatic remeshing, *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Pineridge Press, 615–626, 1994.
- [45] X. Li, M. S. Shephard, M. W. Beall, 3D anisotropic mesh adaptation by mesh modification, *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 4915–4950.
- [46] M. S. Shephard, J. E. Flaherty, K. E. Jansen, X. Li, X. Luo, N. Chevaugeon, J. F. Remacle, M. W. Beall, R. M. Obara, Adaptive mesh generation for curved-domains, *Applied Numerical Mathematics* 52 (2005) 251–271.
- [47] J. F. Remacle, X. Li, M. S. Shephard, J. E. Flaherty, Anisotropic Adaptive Simulation of Transient Flows using Discon-

- tinuous Galerkin Methods, *International Journal for Numerical Methods in Engineering* 52 (2005) 899–923.
- [48] F. Alauzet, Adaptation de maillage anisotrope en trois dimension. Applications aux simulations instationnaires en Mécanique des Fluides, Ph.D. thesis, Université de Montpellier II, Montpellier, 2003.
 - [49] F. Alauzet, P. Frey, P.-L. George, B. Mohammadi, 3D transient fixed point mesh adaptation for time-dependent problems: Application to CFD simulations, *Journal of Computational Physics* 222 (2007) 592–623.
 - [50] C. Gruau, T. Coupez, 3D Tetrahedral, Unstructured and Anisotropic Mesh Generation with Adaptation to Natural and Multidomain Metric, *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 4951–4976.
 - [51] L. Formaggia, S. Micheletti, S. Perotto, Anisotropic mesh adaptation in computational fluid dynamics: application to the advection-diffusion-reaction and the Stokes problems, *Appl. Numer. Math.* 51 (4) (2004) 511–533, ISSN 0168-9274.
 - [52] H. Nguyen, M. Gunzburger, L. Ju, J. Burkardt, Adaptive anisotropic meshing for steady convection-dominated problems, *Computer Methods in Applied Mechanics and Engineering* 198 (37–40) (2009) 2964 – 2981.
 - [53] M. Picasso, V. Prachittham, An adaptive algorithm for the Crank-Nicolson scheme applied to a time-dependent convection-diffusion problem, *Journal of Computational and Applied Mathematics* 233 (4) (2009) 1139 – 1154.
 - [54] F. Alauzet, M. Mehrenberger, P1-conservative solution interpolation on unstructured triangular meshes, *International Journal for Numerical Methods in Engineering* 84 (13) (2010) 1552–1588.
 - [55] L. P. Franca, S. L. Frey, T. J. Hughes, Stabilized finite element methods: I. Application to the advective-diffusive model, *Computer Methods in Applied Mechanics and Engineering* 95 (2) (1992) 253 – 276.
 - [56] R. Codina, Comparison of some finite element methods for solving the diffusion-convection-reaction equation, *Computer Methods in Applied Mechanics and Engineering* 156 (1–4) (1998) 185 – 210.
 - [57] R. Codina, On stabilized finite element methods for linear systems of convection-diffusion-reaction equations, *Computer Methods in Applied Mechanics and Engineering* 188 (1–3) (2000) 61 – 82.
 - [58] L. Franca, F. Valentin, On an improved unusual stabilized finite element method for the advective-reactive-diffusive equation, *Computer Methods in Applied Mechanics and Engineering* 190 (13–14) (2000) 1785 – 1800.
 - [59] A. Mitchell, D. Griffiths, Generalised Galerkin methods for second order equations with significant first derivative terms, in: G. Watson (Ed.), *Numerical Analysis*, vol. 630 of *Lecture Notes in Mathematics*, Springer Berlin / Heidelberg, 90–104, 1978.
 - [60] T. E. Tezduyar, Y. Osawa, Finite element stabilization parameters computed from element matrices and vectors, *Computer Methods in Applied Mechanics and Engineering* 190 (3-4) (2000) 411–430.
 - [61] A. M. Baptista, E. E. Adams, P. Gresho, Benchmarks for the Transport Equation: The Convection-Diffusion Forum and Beyond, vol. 47, American Geophysical Union, 1995.
 - [62] E. Hachem, B. Rivaux, T. Klozcko, H. Dignonnet, T. Coupez, Stabilized finite element method for incompressible flows with high Reynolds number, *Journal of Computational Physics* 229 (2010) 8643–8665.