ELSEVIER

Comput. Methods Appl. Mech. Engrg. 182 (2000) 379–400

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Adaptive finite element computational fluid dynamics using an anisotropic error estimator

Regina C. Almeida [a], Raúl A. Feijóo [a,*], Augusto C. Galeão [a], Claudio Padra [b], Renato S. Silva [a]

[a] *Dept. de Mecanica Computacional, Laboratório Nacional de Computação Científica (LNCC/CNPq), Av. Getúlio Vargas 333, 25651-070 Quitandinha, Petrópolis, RJ, Brazil*

[b] *Instituto Balseiro, Centro Atómico de Bariloche (CAB), 8400 Bariloche, Argentina*

Received 1 September 1998

Abstract

The aim of this paper is to present the results on finite element adaptive strategies for computational fluid dynamics (CFD) problems with singularities arising from shock phenomena and/or discontinuous boundary data. The adaptive analysis is based on an optimal-mesh-adaptive strategy which is employed to refine the mesh, stretch and orient the elements in such a way that, along the adaptation process, the mesh becomes aligned with the discontinuities. This mesh adaptation process yields improved results in locating regions of rapid or abrupt variations of the variables, whose location is not known a priori. On the other hand, the proposed mesh adaptation process is generated by minimizing, for a given number of elements in the mesh, a new anisotropic error estimator based on local directional interpolation error and recovering of the second derivatives of the finite element solution. Several adaptive mesh-refinement solutions for interpolation problems are presented in order to show that the proposed optimal adaptive strategy using this anisotropic error estimator recovers optimal and/or superconvergent rates. Finally, applications of this approach to CFD problems are also presented in order to show the computational performance of the proposed optimal adaptive procedure.

Keywords: Error estimator; Adaptive analysis; Computational fluid dynamics; Finite elements; Mesh generation

1. Introduction

In several situations of computational fluid dynamics (CFD) we are faced with convection dominated phenomena. For such problems, stabilized finite element methods should be used in order to avoid numerical spurious oscillations which normally occur when the classical Galerkin's method is adopted. Among those stabilized models, the SUPG and CAU methods [1,9], previously derived for convection-dominated scalar problems and further generalized for compressible flows, had been successfully used to numerically describe localized singularities arising from shock phenomena and/or discontinuous boundary data. Nevertheless, the accuracy can be greatly improved by placing more grid-points where the solutions present those singularities. So, it is necessary not only to identify those regions, but also to obtain a good equilibrium between the refined and unrefined regions, such that the overall accuracy is optimal. In CFD, a priori error estimates, as the ones provided by the standard error analysis of finite elements, are often insufficient to assure reliable estimates of the accuracy of the computed numerical solution. This is so because they only yield information on the asymptotic error behavior and require regularity conditions of the solution which are not satisfied in the presence of singularities.

* Corresponding author. Tel.: +55-024-233-6017; fax: +55-024-233-6165.
E-mail address: feij@lncc.br (R.A. Feijóo).

0045-7825/00/$ - see front matter
PII: S0045-7825(99)00200-5

Those facts show the need for an estimator which can a posteriori be extracted from the computed numerical solution and the given data of the problem. Basically, a posteriori error estimators can be classified into two types. The first of these was introduced by Babuška and Rheinboldt [3,4] and it is based on evaluating the residuals of the approximate solution. Therefore, it strongly depends on the problem operator. The second approach, proposed by Zienkiewicz and Zhu [29], estimates the error in gradient-based norms, using some recovery process. In that work, simple averaging techniques were used yieding acceptable estimates, which makes this approach extremely popular.

The general standard error analysis is based on the *regularity assumption* on the meshes and on the exact solution. For the Dirichlet problem described by the Laplace equation defined on a general polygonal domain, Grisvard [17] showed that the solution u belongs to $W^{2,p}(\Omega)$, $1 < p < 4/3$. For this problem optimal error estimates for finite elements mesh refinements were obtained.

On the other hand, in two-dimensional problems it is known that the regularity assumption on the meshes (the ratio between outer and inner diameter of the elements remains bounded when $h \to 0$) can be replaced by a maximum angle condition for triangular elements. Moreover, for problems involving flows with shocks, boundary layers, contact discontinuites, etc., narrow elements are also needed in appropriate directions in order to represent those features economically because of the conflict of scales, an inseparable aspect of this kind of problem.

In [1], we introduced an h-adaptive mesh refinement strategy for unstructured meshes including direction of stretching and stretching ratio where the refinement and/or unrefinement is guided by an a posterior error indicator based on the norm used in [19] for the convergence analysis of the SUPG formulation for linear time-dependent multidimensional advective–diffusive systems. The direction of stretching was taken normal to the direction of the maximum change of the density variable and the stretching ratio was taken larger in regions where the flow is nearly one-dimensional. This strategy is very effective on generating finer grids near sharp layers and coarse meshes for regions of smooth solutions. However, since the stretching ratio was not determined freely by the flow solution because its range was set from 1 to 3, regions with almost regular meshes were generated by this methodology. Therefore, this strategy can probably be improved by considering an optimized approach to control the stretching allowing for more distorted meshes.

On the other hand, the adapted meshes of Peiró [22] and Peraire et al. [23,24], the highly stretched elements for solving viscous flows also reported by Mavriplis [20] and Marcum et al. [18] among others, show that anisotropic meshes produce highly accurate results for two- and three-dimensional problems while keeping the number of unknowns low. Incorporation of stretching through a local non-euclidean metric was reported in [22,23]. This approach can lead to elements with large (close to π) angles that researchers were able to preclude because of the maximum-angle condition [2]. However, if the mesh is adapted to the solution (i.e., if the elements are thin along the direction of the maximum absolute value of the second directional derivative [25]) it is possible to circumvent this condition. Several recent results [10–12,14] and papers therein have brought new attention to metric-based stretching where the underlying metric came from a recovered Hessian. In particular, Buscaglia and Dari [10,11] introduced a mesh optimization procedure where only operations that improve the mesh quality (which is a function of the recovered Hessian) are allowed.

In the present paper, we place particular emphasis on the anisotropic mesh adaptation process generated by a directional error estimator based on the recovering of the second derivatives of the finite element solution. The goal of this approach is to achieve a mesh-adaptive strategy minimizing the directional error estimator for a given number of elements in the mesh. As shown in this paper, this approach allows us to refine the mesh, stretch and orient the elements in such a way that, along the adaptation process, the mesh becomes aligned with the discontinuities whose locations are not known a priori. As a result, highly accurate solutions are obtained keeping the number of unknowns affordably low.

In our approach the recovered Hessian is carried out, but differs from the recovering process mentioned before in

(i) using the recovered Hessian, it is defined a new anisotropic error estimator;

(ii) minimizing this anisotropic error estimator for a given number of elements an optimal-mesh-adaptive procedure is built;

(iii) using a C^{++} implementation of the *advancing front technique* [15,21–23] the resulting meshes allow very large stretched elements.

Hence, in this paper we present an a posteriori anisotropic error estimator for the difference between a given function $u \in W^{2,p}$ and a discrete function $U \in V_h$ which is a "good approximation" of u in Ω, in the sense that,

$$\|u - U\|_{L^p(\Omega)} \leqslant C\|u - \Pi u\|_{L^p(\Omega)}. \tag{1}$$

In the above expression $\Pi : W^{2,p}(\Omega) \to V_h$ is an operator whose approximation properties are similar to the Clément interpolation operator [13], i.e.,

$$\|u - \Pi u\|_{L^p(T)} \leqslant C\left(\sum_{K \subset \tilde{T}} \|D_2 u(x)(x - x_K) \cdot (x - x_K)\|^p_{L^p(K)}\right)^{1/p}, \tag{2}$$

where $\tilde{T}$ is the patch formed by T and its neighboring elements and x_K is the centroid of element K.

The a posteriori error estimator is based on processing the function U with the hope of obtaining a better approximation to the Hessian matrix of u. This approximation will be used instead of the exact Hessian matrix to estimate the L^p-norm of the error $e = u - U$.

The process to obtain a *recovered* Hessian matrix from the function U is based on applying twice a technique to recover the gradient. Zienkiewicz and Zhu [29] were the first to use a recovered gradient to estimate the energy norm of the error of the finite element approximation (for more information about recovering first derivatives see also [6–8,16,26–28,30–34] and papers therein). Therefore, a natural question is in what manner we may recover the Hessian matrix to provide again the error in the principal direction of the Hessian in each element.

Peiró [22] obtained a recovered Hessian matrix solving, by finite elements, a variational equation for the Hessian (see also [6,11,16,28]). The matrix of this system is the so-called *mass matrix*. The computation would be as expensive as the approximate solution of the problem itself. As a general rule, the computational cost for a posteriori error evaluation must be inexpensive in comparison with the solution of the problem. Because of this, Peiró proposed in [22] to use the "lumped" form of the mass matrix.

Combining (1) and (2) we get

$$\begin{aligned}\|u - U\|_{L^p(\Omega)} &\leqslant C\|u - \Pi u\|_{L^p(\Omega)} \leqslant C\left(\sum_{T \in \mathscr{T}} \|\mathbf{H}(u)\|^p_{L^p(\tilde{T})}\right)^{1/p} \\ &\leqslant C\left(\sum_{T \in \mathscr{T}} \left(\|\mathbf{H}(u) - \mathbf{H}_{\mathrm{R}}(U)\|_{L^p(\tilde{T})} + \|\mathbf{H}_{\mathrm{R}}(U)\|_{L^p(\tilde{T})}\right)^p\right)^{1/p},\end{aligned} \tag{3}$$

where $\mathbf{H}(u)(x) = D_2 u(x)(x - x_K) \cdot (x - x_K)$, $\mathbf{H}_{\mathrm{R}}(U) = D_{2\mathrm{R}} U(x)(x - x_K) \cdot (x - x_K)$ and $D_{2\mathrm{R}} U$ is a recovery Hessian operator.

We assume (numerical evidences for that will be presented in this paper and a more mathematical justification will appear in a forthcoming paper) that there exists a constant C^* such that

$$\|\mathbf{H}(u) - \mathbf{H}_{\mathrm{R}}(U)\|_{L^p(\tilde{T})} \leqslant C^* N_{\mathscr{T}}^{-\beta} \|\mathbf{H}_{\mathrm{R}}(U)\|_{L^p(\tilde{T})}, \tag{4}$$

where $\beta \geqslant 0$ and $N_{\mathscr{T}}$ denotes the number of elements of the mesh $\mathscr{T}$. Now, using (3) and (4) we get

$$\|u - U\|_{L^p(\Omega)} \leqslant C(1 + N_{\mathscr{T}}^{-\beta}) \|\mathbf{H}_{\mathrm{R}}(U)\|_{L^p(\Omega)}. \tag{5}$$

This shows that the L^p-norm of the recovered Hessian can be used as an a posteriori error estimator.

The outline of this paper is as follows. The first main contribution, namely the anisotropic error estimator, together with the procedures to get the recovered Hessian matrix are described in Section 2. The second contribution, namely the optimal-mesh-adaptive procedure is presented in Section 3. In Section 4 we show numerical results which exhibit the advantageous feature of the procedure to get the recovered Hessian matrix. We also present numerical evidences that assumption (4) is satisfied for regular and non-regular functions with singularities and/or abrupt variations. Finally, we show that the optimal-mesh-adaptive algorithm together with our anisotropic error estimator are suitable for an efficient generation of meshes for several CFD problems (applications of the same approach to Limit Analysis Problems are also reported in [6]).

2. The a posteriori error estimator

Standard assumptions and notation are introduced in this section. Given a polygon $\Omega \subset \mathbb{R}^2$, we consider a family of triangulations $\{\mathscr{T}_h\}$ of Ω such that any two triangles in $\mathscr{T}_h$ share at most a vertex or an edge. In the definition of the error estimator we will use the standard conforming space

$$V_h = \{v \in H_0^1(\Omega) : v|_T \in \mathscr{P}_s(T)\},$$

where $\mathscr{P}_s(T)$ denotes the space of polynomials of degree no greater than k. For each node N_i, $1 \leqslant i \leqslant M$ of the triangulation, we associate a basis function $\varphi_i \in V_h$ in the usual manner. For each i, we set $S_i = \operatorname{supp} \varphi_i$.

2.1. Some derivative recovery techniques

2.1.1. Gradient recovery operator

As a first step, we recall the *Clément* interpolation operator. Given a function $v \in L^2(\Omega)$, we let $P_i v$ denote the projection of the function $v|_{S_i} \in L^2(S_i)$ over the subspace $\mathscr{P}_k(S_i)$, i.e. one has

$$P_i v \in \mathscr{P}_k(S_i) \quad \text{and} \quad (v - P_i v, p)_{S_i} = 0 \quad \forall p \in \mathscr{P}_k(S_i), \tag{6}$$

where $(\cdot,\cdot)_{S_i}$ denotes de L_2-inner product in S_i, and we set

$$\Pi_{\mathrm{C}} v = \sum_{i=1}^{M} P_i v(N_i) \varphi_i.$$

We define the following gradient recovery operator $\nabla_{\mathrm{R}} : V_h \to (V_h)^2$ given by

$$\nabla_{\mathrm{R}} u_h = \Pi_{\mathrm{C}}(\nabla u_h) := \left[\Pi_{\mathrm{C}}\left(\frac{\partial u_h}{\partial x}\right), \Pi_{\mathrm{C}}\left(\frac{\partial u_h}{\partial y}\right) \right]$$

Remark. *The case $k = 0$ is the simplest algorithm known as the area weighted average (in Section* 4 *we use an index A to indicate this type of recovery technique); indeed, we have*

$$\nabla_{\mathrm{R}} u_h(N_i) = \left[P_i\left(\frac{\partial u_h}{\partial x}\right), P_i\left(\frac{\partial u_h}{\partial y}\right) \right] = \frac{\sum_{T \subset S_i} |T| \nabla(u_h|_T)}{|S_i|},$$

where $|T|$ and $|S_i|$ are the areas of element T and patch S_i respectively.

Remark. *In* (6) *the Gaussian and/or optimal integration points can be used for the numerical computation of the integral. In this case the algorithm is called* patch recovery technique [31]. *In Section* 4 *we use the index IPE for referring to this recovery technique called as* Integral Polynomial Expansion [16]. *If the L^2-inner product is replaced in* (6) *by a* discrete inner product *using appropriate sampling points in the patch, the technique is called* discrete patch recovery *or* Local Polynomial Expansion – LPE [16,31].

Remark. *If the L^2-inner product is replaced in* (6) *by a weighted inner product, i.e.,*

$$(v, w)_{S_i} = \int_{S_i} v(x)\, w(x)\, g_i(x)\, \mathrm{d}x,$$

where $g_i(x)|_T = (|T|\,|x - N_i|)^{-1}$ for all $T \subset S_i$, we obtain another family of methods called as inverse distance weighted average – (ID). *For the case $k = 0$ we obtain*

$$\nabla_{\mathrm{R}} u_h(N_i) = \frac{\sum_T \nabla(u_h|_T) \int_T g_i(x)\, \mathrm{d}x}{\int_{S_i} g_i(x)\, \mathrm{d}x}$$

and using the one point Gaussian rule we get

$$\nabla_{\mathrm{R}} u_h(N_i) = \frac{\sum_T \nabla(u_h|_T)(1/d_T)}{\sum_T (1/d_T)}, \tag{7}$$

where $d_T = |x_T - N_i|$ and x_T is the baricenter of T. It is interesting to note that for one-dimensional problems, technique (7) *recovers exact first-order derivatives for quadratic functions as shown by Babuška and Rodriguez* [5].

As was pointed out in [31], the computed recovered gradient performs better on nodes interior to the domain rather than on nodes along the boundaries. Trying to overcome this effect, recovery for boundary nodes is performed by using appropriate interior patches instead of the standard patch associated with the boundary node (see also [6,16]).

Many approaches exist to improve the accuracy of recovering techniques (see for example [7,8,26,27]). In all of them some additional knowledge on the behavior of the function is exploited. We will avoid those procedures aiming to use the minimum information about the function itself whose gradient we want to recover.

As will be shown later in Section 4, when the solution of a problem has strong gradients, the mesh become oriented and stretched in the direction of the maximal curvature of the function. For high stretches, the matrix associated with the recovery technique tends to become ill-conditioned. Moreover, for weighted average techniques, such as area and inverse distance, the stretches may yield poor accuracy in the calculations. To improve the accuracy of the recovering techniques we will then employ a *normalizing patch domain approach*. In this normalized domain we perform the computations needed in the recovery procedures. After that, we have to change, according to the chain rule, derivatives to the original domain. In the following, for simplicity, argument N is used to indicate the value of a function estimated in the node with coordinate $\mathbf{X}_N$. Since we will use the *advancing front technique* [15,21–23], it is useful to adopt a domain transformation in accordance with the mapping induced by the generator algorithm. Around an arbitrary point N of the mesh, the algorithm based on the *advancing front technique* tries to build equilateral triangles in the metric defined by the variable metric tensor $\mathbf{S}$, given by

$$\mathbf{S}(N) = \frac{1}{s(N) \ * \ h(N)} \mathbf{e_1} \otimes \mathbf{e_1} + \frac{1}{h(N)} \mathbf{e_2} \otimes \mathbf{e_2}, \tag{8}$$

where $\mathbf{e}_i(N),\ i = 1, 2$, are the eigenvectors of the recovered Hessian matrix $H_{\mathrm{R}}(u_h(N))$. $h(N)$ represents the height of the element (measured in the $\mathbf{e}_2$ direction), and $s(N) \geqslant 1$ is the stretching in the $\mathbf{e}_1$ direction of an element generated at node N. Those parameters are computed in the course of the adaptive procedure and the strategy adopted to estimate them is described in Section 3. Then, a point M in any element around node N, is transformed into a point $\mathbf{x}_M$ by

$$\mathbf{x}_M = \mathbf{S}(N)(\mathbf{X}_M - \mathbf{X}_N),$$

where $\mathbf{X}_M$ and $\mathbf{x}_M$ are the coordinates of M before and after the mapping, respectively. Therefore, the first-order gradients recovery algorithm can be summarized as indicated in Table 1.

2.1.2. Hessian recovery operator

Now, we introduce the recovered Hessian $H_{\mathrm{R}}(u_h) \in V_h^{2\times 2}$ given by

$$H_{\mathrm{R}}(u_h) = \frac{H^*(u_h) + (H^*(u_h))^t}{2}, \tag{9}$$

Table 1
First order gradients recovery algorithm

(i)	Define the patch associated with node N.
(ii)	Build the metric tensor $\mathbf{S}(\mathbf{N})$, from information about the mesh shape around node N, defined by the known parameters $s(N)$, $h(N)$, $\mathbf{e_1}$ and $\mathbf{e_2}$.
(iii)	Transform all elements in the patch.
(iv)	Compute the gradients $\mathrm{grad}\, u_h$ in each one of the transformed elements.
(v)	Use any recovery algorithm to compute $\mathrm{grad}_{\mathrm{R}}\, u_h(N)$.
(vi)	From $\nabla_{\mathrm{R}}\, u_h(N) = \mathbf{S}^{\mathbf{T}}(N\ )\mathrm{grad}_{\mathrm{R}}\, u_h(N)$ transform the gradient $\mathrm{grad}_{\mathrm{R}}\, u_h(N)$ to the original domain.

where $H^*(u_h)$ is the matrix defined by

$$H^*(u_h) = \begin{pmatrix} \nabla_R(\Pi_C(\frac{\partial u_h}{\partial x})) \\ \nabla_R(\Pi_C(\frac{\partial u_h}{\partial y})) \end{pmatrix}. \tag{10}$$

Other approaches for recovery of second-order derivatives are described in [11,22] and [10,16].

2.2. A posteriori anisotropic error estimator

Assuming that $u_h \in V_h$ is a good approximation for the function u, hence from (4) and (5) we get

$$\|u - u_h\|_{L^p(\Omega)} \simeq C\|H_R(u_h(x))(x - x_0) \cdot (x - x_0)\|_{L^p(\Omega)}. \tag{11}$$

This shows that the interpolation error in one point x such that $|x - x_0|$ is small enough, is governed by the behavior of the second-order derivative in such point. Thus, the interpolation error is not distributed in an isotropic way around point x_0; i.e., the error depends on *direction* $x - x_0$ and on $H_R(u(x))$, the recovered Hessian matrix value in this point. Therefore, (11) suggests the use of this expression as a *directional local error estimator*. However, the non-positive-definiteness of $H_R(u(x))$ precludes its use as a metric. Following the approach suggested by Peiró [22], we introduce the tensor

$$\mathbf{G} = \mathbf{Q}\Lambda\mathbf{Q}^T \tag{12}$$

for which the columns entries of the matrix $\mathbf{Q}$ are the eigenvectors of the recovered Hessian matrix, and $\Lambda = \mathrm{diag}\{|\lambda_1|, |\lambda_2|\}$ where $|\lambda_i|$, $i = 1, 2$, are the absolute values of the eigenvalues of $H_R(u_h(x))$ ($|\lambda_1| \leqslant |\lambda_2|$).

Now, given a partition $\mathscr{T}_h$ of Ω, we define

- the anisotropic error estimator of the element $T \in \mathscr{T}_h$ by

$$\eta_T = \left\{ \int_{\Omega_T} (\mathbf{G}(u_h(x_0))(x - x_0) \cdot (x - x_0))^p \, d\Omega \right\}^{1/p}, \tag{13}$$

  where x_0 is the baricenter of T;
- the global error estimator η by

$$\eta = \left\{ \sum_{T \in \mathscr{T}_h} (\eta_T)^p \right\}^{1/p}. \tag{14}$$

From definition (13) we get

$$\begin{aligned} (\eta_T)^p &= \int_{\Omega_T} (\mathbf{G}(u_h(x_0))(x - x_0) \cdot (x - x_0))^p \, d\Omega \\ &= \int_{\Omega_T} \left(\sum_{i=1,2} |\lambda_i(x_0)| [e_i(x_0) \otimes e_i(x_0)](x - x_0) \cdot (x - x_0) \right)^p d\Omega \\ &= \int_{\Omega_T} \left\{ \sum_{i=1,2} |\lambda_i(x_0)| [e_i(x_0) \cdot (x - x_0)]^2 \right\}^p d\Omega \\ &\leqslant \int_{\Omega_T} \left\{ \sum_{i=1,2} |\lambda_i(x_0)| h_i^2 \right\}^p d\Omega, \end{aligned} \tag{15}$$

which gives an upper bound for the proposed anisotropic error estimator η_T.

Now, we can introduce an *optimal local error constraint* requiring the *shape* of the element to be such that the estimated error yields the same value in any direction. This is equivalent to imposing a constant error along the principal directions of curvature of each element, i.e.,

$$|\lambda_1| h_1^2 = |\lambda_2| h_2^2 = \mathrm{cte}. \tag{16}$$

Using this requirement we can therefore state the *stretching of the element T* as

$$s_T := \frac{h_1}{h_2} = \sqrt{\frac{|\lambda_2|}{|\lambda_1|}}. \tag{17}$$

Combining (16) and (15) we get

$$(\eta_T)^p \leqslant \mathrm{area}_T \{2\,|\lambda_2(x_0)|\,h_2^2\}^p, \tag{18}$$

which leads to

$$\eta_T \leqslant 2\,\mathrm{area}_T^{1/p}\,|\lambda_2(x_0)|\,h_2^2. \tag{19}$$

This shows that the local error estimator is bounded by the maximum value of the second-order derivative in the baricenter of the element times the square of the element height.

The above upper bound allows us to introduce another local error estimator as

$$\eta_T = 2\,\mathrm{area}_T^{1/p}\,|\lambda_2(x_0)|\,h_2^2 \tag{20}$$

and the global estimator

$$\eta = \left\{ \sum_{T\in\mathscr{T}_h} (\eta_T)^p \right\}^{1/p}. \tag{21}$$

In Section 3, we will propose an adaptive strategy for which the resulting adapted meshes aim to minimize the estimated error with the minimum computational cost (*optimal adaptive analysis*).

3. The optimal mesh adaptive procedure

Let $\mathscr{T}_n$ denote the finite element partition at the nth-step of the adaptive analysis, u_n denotes the approximate solution associated with this mesh and $h_{\mathrm{old}}(N)$ describes the height of the elements at node N in the direction of the maximum absolute value of the directional recovered second order derivative at the nodal point N. Then, given the number of desired elements in the new adapted mesh, the optimal-mesh-adaptive procedure generates a *new* mesh, $\mathscr{T}_{n+1}$, such that the distribution $h_{\mathrm{new}}(N)$, for all $N \in \mathscr{T}_n$ minimizes the global estimator error (21). Hence, the optimal-mesh-adaptive procedure can be formally cast as a constrained optimization problem

$$(\mathscr{P}) \quad \begin{cases} \text{Find } h_T,\ T \in \mathscr{T}_n \text{ such that minimizes the functional} \\ \mathscr{F}(h_T) = \sum_{T\in\mathscr{T}_n} 2^{p-1} s_T\,|\lambda_{2T}|^p\,h_{2T}^{2(p+1)} \text{ with the subsidiary condition} \\ N_{\mathscr{T}_{n+1}} = (4/\sqrt{3}) \sum_{T\in\mathscr{T}_n} \int_T \det \mathbf{S}\,\mathrm{d}\Omega = (4/\sqrt{3}) \sum_{T\in\mathscr{T}_n} \int_T (1/(s_T h_{2T}^2))\,\mathrm{d}\Omega. \end{cases} \tag{22}$$

The functional $\mathscr{F}(h_T)$ is derived from (20) and (21) and from the fact that during the adaptive procedure, expression (17) must be satisfied, i.e.

$$\mathrm{area}_T = \frac{1}{2} s_T(x_0) h_{2T}^2, \tag{23}$$

where $s_T(x_0)$ is the stretch of the element $T \in \mathscr{T}_n$ at the baricenter point x_0.

Introducing the following notations

$$C_{1T} = 2^{p-1} s_T\,|\lambda_{2T}|^p, \quad C_{2T} = \frac{4}{\sqrt{3}} \frac{1}{s_T}, \quad \gamma_T = \frac{1}{h_{2T}^2} \tag{24}$$

the minimization problem $(\mathscr{P})$ can be rewritten as follows:

$$(\mathscr{P}) \quad \begin{cases} \text{Find } h_{2T},\ T \in \mathscr{T}_n \text{ such that minimizes the functional} \\ \mathscr{F}(\gamma_T) = \sum_{T\in\mathscr{T}_n} \frac{C_{1T}}{\gamma_T^{p+1}} \text{ with the subsidiary condition} \\ N_{\mathscr{T}_{n+1}} = \sum_{T\in\mathscr{T}_n} \int_T C_{2T}\,\gamma_T\,\mathrm{d}\Omega \end{cases} \tag{25}$$

which is equivalent to the following min–max problem

$$\min_{\gamma_T}\max_{\beta}\mathscr{L}(\gamma_T,\beta)=\mathscr{F}(\gamma_T)-\beta\left[N_{\mathscr{T}_{n+1}}-\sum_{T\in\mathscr{T}_n}\int_T C_{2T}\gamma_T\,\mathrm{d}\Omega\right]. \tag{26}$$

The solution of this min–max problem is given by

$$\frac{\partial\mathscr{F}}{\partial\gamma_T}=0\ \Rightarrow\ -\frac{(p+1)\,C_{1T}}{\gamma_T^{p+2}}+\beta\int_T C_{2T}\,\mathrm{d}\Omega=0, \tag{27}$$

$$\frac{\partial\mathscr{F}}{\partial\beta}=0\ \Rightarrow\ N_{\mathscr{T}_{n+1}}-\sum_{T\in\mathscr{T}_n}\int_T C_{2T}\gamma_T\,\mathrm{d}\Omega=0. \tag{28}$$

From (27) we get

$$\gamma_T=\left[\frac{(p+1)\,C_{1T}}{\beta\int_T C_{2T}\,\mathrm{d}\Omega}\right]^{1/(p+2)} \tag{29}$$

and substituting this expression in (28) we obtain

$$\beta^{1/(p+2)}=N_{\mathscr{T}_{n+1}}^{-1}\sum_{T\in\mathscr{T}_n}\left\{\left(\frac{1}{\int_T C_{2T}\,\mathrm{d}\Omega}\right)^{1/(p+2)}\int_T C_{2T}[(p+1)C_{1T}]^{1/(p+2)}\,\mathrm{d}\Omega\right\}. \tag{30}$$

From definition the third equation of (24)

$$h_{2T}=\sqrt{\frac{1}{\gamma_T}} \tag{31}$$

and γ_T and β as given by (29) and (30), respectively, we finally arrive at the optimal value of h_{2T}.

Up to this point we know the optimal distribution of the elements shapes for the new mesh (given by (31), the stretch s_T and the direction of this stretching given by the first eigenvector of the recovered Hessian matrix). Using this information and the advancing front technique, the new (optimal) mesh can be generated. Here, it is interesting to note that since the product $h_{2T}\cdot s_T$ must be limited by the characteristic length of the domain Ω, the optimal values of $h_{2T},\ T\in\mathscr{T}_n$, must be modified prior to generating the new mesh. Due to this limitation, the number of elements in the new adapted mesh may be different from the expected one, $N_{\mathscr{T}_{n+1}}$. To force the equality between these two numbers, the elements sizes h_{2T} at nodal level, must be scaled (see [6,16]). Therefore, the optimal-mesh-adaptive algorithm can be summarized as indicated in Table 2.

Table 2
Optimal-mesh-adaptive algorithm

(i)	For each node N evaluate: (a) $H_{\mathrm{R}}(u_h)$ according to Box 1 and expressions (9) and (10), (b) using (12) evaluate the metric $\mathbf{G}$.
(ii)	According to expression (13) and (14) evaluate $\eta_T,\ T\in\mathscr{T}$ and η respectively.
(iii)	If $\eta\leqslant\eta_{\mathrm{adm}}$ STOP.
(iv)	For a given number of elements and from (31), evaluate $h_{2T},\ T\in\mathscr{T}$.
(v)	Project h_{2T} in $[\underline{\alpha}\cdot h_{2T\mathrm{old}},\overline{\alpha}\cdot h_{2T\mathrm{old}}]$($\underline{\alpha}$ and $\overline{\alpha}$ control the mesh generation).
(vi)	Using any one of the weighted average algorithms, compute $h_{2T}(N)$.
(vii)	Project $s(N)$ in $[1,L/h_{2T}(N)]$, where L is the characteristic length of the domain.
(viii)	Perform the $h_{2T}(N),\ N\in\mathscr{T}$, scaling.
(ix)	Using the variable metric $\mathbf{S}(N)\ (N\in\mathscr{T})$ defined in (8), and with the *advancing front* mesh generation algorithm, compute the *new* optimal-mesh. (Other unstructured mesh generators can also be used. They should take into account the variable metric $\mathbf{S}(N),\ N\in\mathscr{T}$, utilized as a background mesh during the mesh generation process).

4. Numerical results

In this section we show several numerical results which allow us to compare the numerical performance of the different recovery procedures for first and second order derivatives. Moreover, we also present numerical evidences that the assumption (4) is satisfied for regular as well as non-regular functions with abrupt variations. Finally, we show that the optimal-mesh-adaptive algorithm together with our anisotropic error estimator are suitable for generating reliable meshes for convection–diffusion and CFD problems.

4.1. Gradient and second order derivatives recovering

Let us take the following functions:
(i) $u(x,y) = (1-x^{40})(1-y^{40}),\ (x,y) \in \Omega = [0,1] \times [0,1]$,
(ii) $u(x,y) = (1-x^{40}),\ (x,y) \in \Omega = [0,1] \times [0,1]$.

As a "good approximation" for each of these functions, we adopt the finite element linear interpolant $u_h \in V_h = \{v \in H_0^1(\Omega) : v|_T \in \mathscr{P}_1(T)\}$ where $\mathscr{P}_1(T)$ denotes the space of polynomials of degree no greater than 1 defined in the triangle $T \in \mathscr{T}$, i.e., $u_h(x_N, y_N) = u(x_N, y_N)$, where (x_N, y_N) are the coordinates of the nodal point N in the finite element mesh $\mathscr{T}$.

For all of these functions we plot, for uniform and adapted meshes obtained with our optimal mesh adaptive procedure, the following norms: $\|u-u_h\|_{L^2(\Omega)}$ ($\|u-uh\|$), $\|\nabla(u-u_h)\|_{L^2(\Omega)}$ ($\|G(u-uh)\|$), $\|\nabla u - \nabla_R u_h\|_{L^2(\Omega)}$ ($\|Gu - G_R u\|$), $\|D_2(u) - D_{2R}(u_h)\|_{L^2(\Omega)}$ ($\|D2 - D2R\|$), $\|\mathbf{H}_R(u_h)\|_{L^2(\Omega)}$ (HR), vs. the number of elements $N_{\mathscr{T}}$ in log scale. Observe that HR corresponds to our estimator (13) and (14). The following notations were also used in the figures we will show: H-HR means the term associated with the assumption (4), MaxHHRVHR is the maximum over all patches in the mesh of the ratio $(\text{H-HR})_{\text{patch}}/(\text{HR})_{\text{patch}}$, and HHR/HR is the ratio between the term H-HR and HR. Also, in the following figures θ is the effectivity index defined as $\text{HR}/\|u-uh\|$.

4.1.1. Uniform meshes

The coarse uniform mesh has 50 elements, $h_{2T} = 0.2$, and the finest uniform mesh has 51,200 elements, $h_{2T} = 0.00625$.

For function $u(x,y) = (1-x^{40})(1-y^{40})$, Figs. 1–4 show the performance of the different recovery techniques described in this paper and Fig. 5 presents the comparison between them.

The results obtained with the integral polynomial expansion recovery technique for function $u(x,y) = (1-x^{40})$, are shown in Fig. 6. Since the problem is one-dimensional, we take $N_{\mathscr{T}}$ as the number of elements over the x-axis.

The effectivity indices for both problems are depicted in Fig. 7, showing the quality of the proposed error estimator.

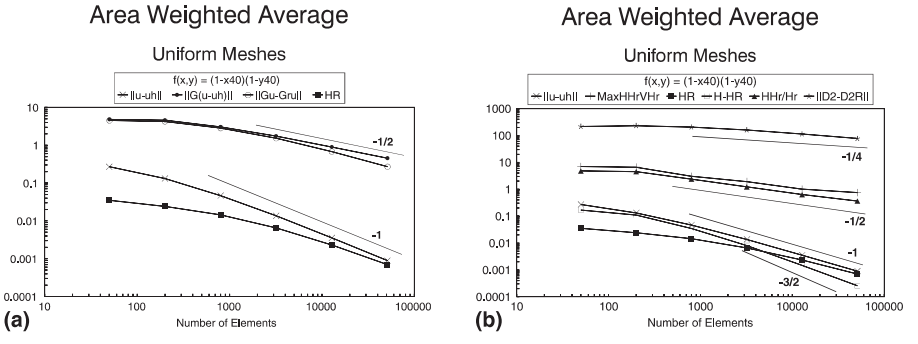

Fig. 1. Performance of the area weighted average recovery technique.

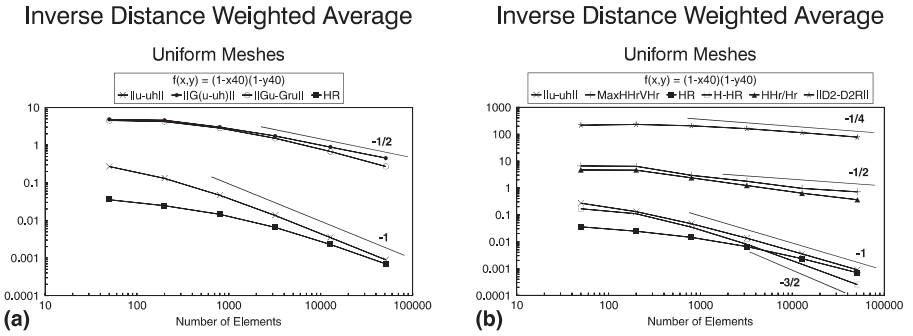

Fig. 2. Performance of the inverse distance weighted average recovery technique.

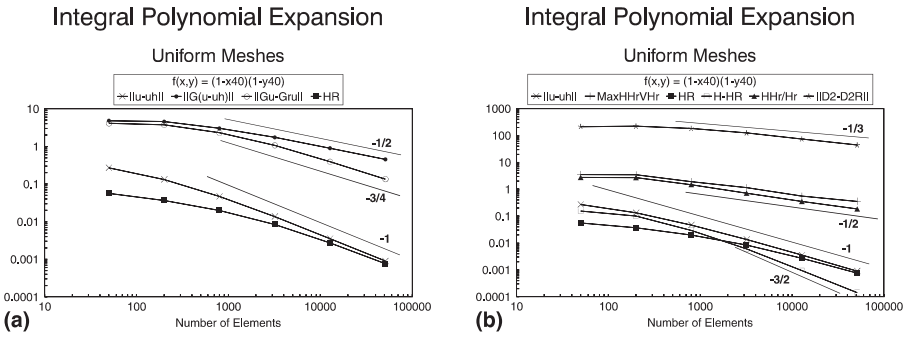

Fig. 3. Performance of the integral polynomial expansion recovery technique.

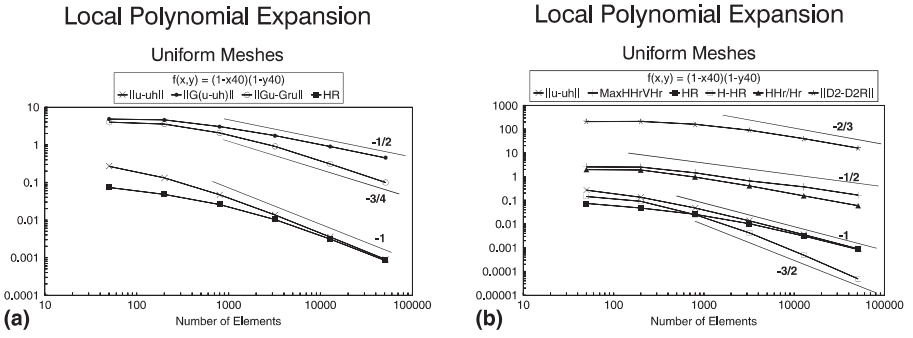

Fig. 4. Performance of the local polynomial expansion recovery technique.

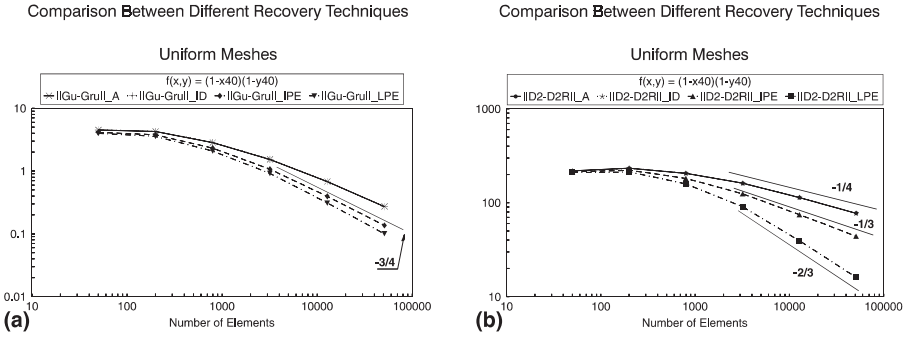

Fig. 5. Comparison between different recovery techniques.

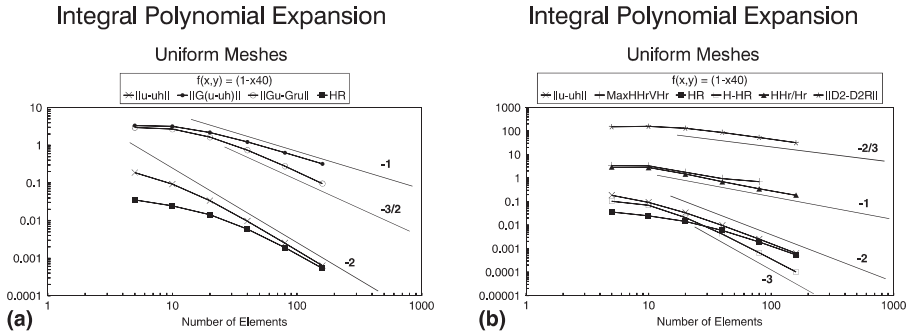

Fig. 6. Integral polynomial expansion recovery results for function $u(x,y) = (1 - x^{40})$.

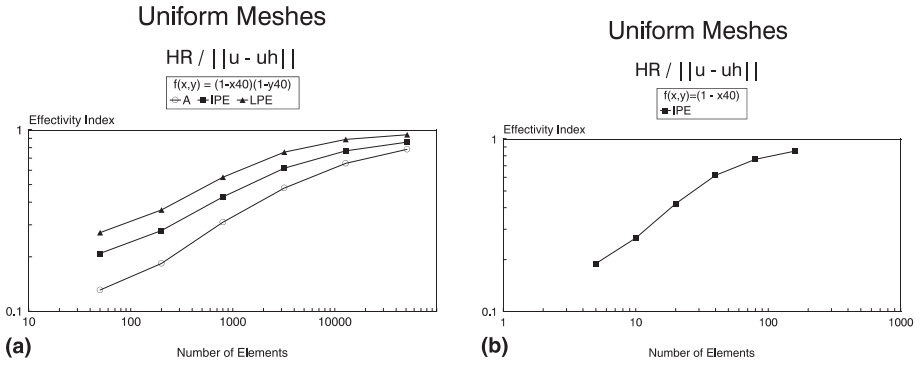

Fig. 7. Effectivity indices for functions $u(x,y) = (1 - x^{40})(1 - y^{40})$ and $u(x,y) = (1 - x^{40})$.

4.1.2. Adapted meshes

As a first application of our optimal-mesh-adaptive procedure, we proceed to apply this algorithm to the function $u(x,y) = (1 - x^{40})(1 - y^{40})$. For the integral polynomial expansion recovery technique, the results are presented in Fig. 8 for meshes with 50 (uniform mesh), 124 (first adapted mesh), 256, 555, 1156 and 2441 elements. For the local polynomial expansion recovery technique the number of elements in each step are 50, 124, 255, 544, 1146 and 2491. The first and the last adapted meshes are shown in Fig. 9 for the former recovery technique. The effectivity indices are presented with each mesh and in both we have $s_T \in [1; 1724]$.

For the last optimal mesh (2441 elements), the result obtained with the integral polynomial expansion recovery technique for the x-component of the recovered gradient $\nabla_R(u_h)$ is depicted in Fig. 10. The same figure plots the result obtained with the local polynomial expansion recovery technique for the

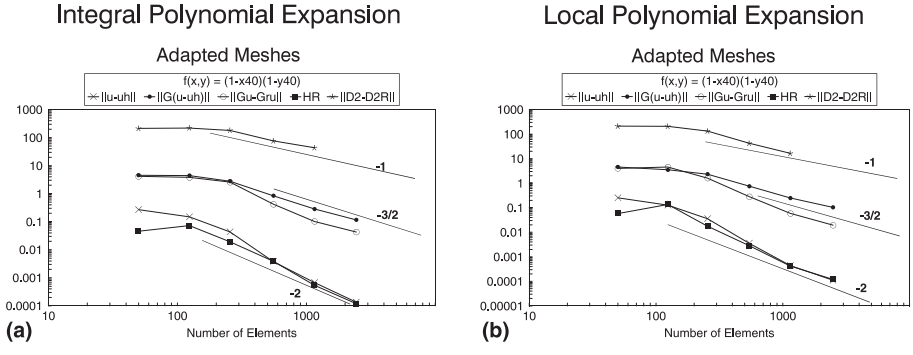

Fig. 8. Optimal mesh adaptive process for function $u(x,y) = (1 - x^{40})(1 - y^{40})$.

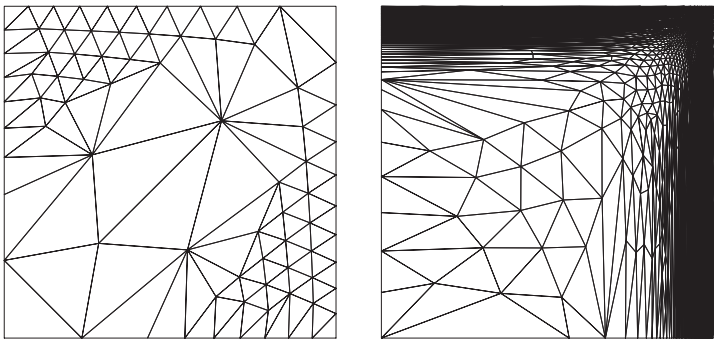

Fig. 9. First ($\theta = 0.1713$) and last ($\theta = 0.8815$) optimal meshes for function $u(x,y) = (1-x^{40})(1-y^{40})$.

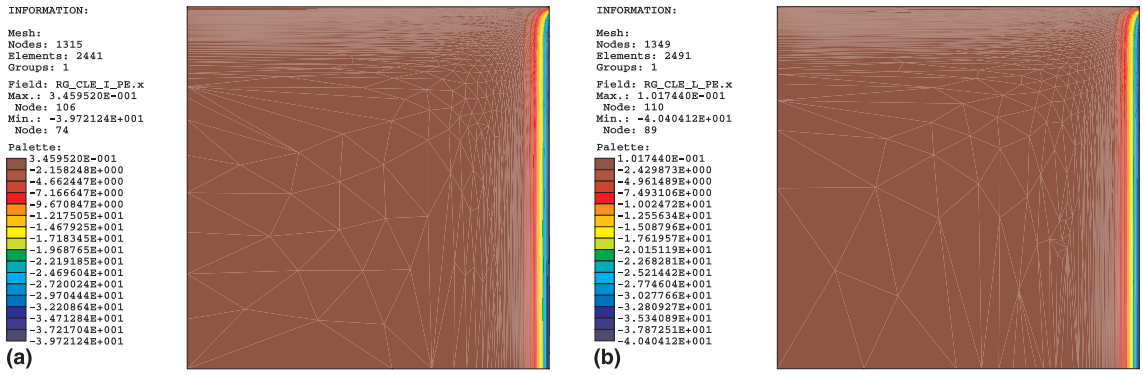

Fig. 10. Function $u(x,y) = (1-x^{40})(1-y^{40})$. x-component of the recovered gradient using the: (a) integral, and (b) local polynomial expansion technique.

corresponding last optimal mesh (2491 elements). Finally, the xx-component of the recovered Hessian $H_R(u_h)$ is depicted in Fig. 11 for the last optimal mesh (2441 elements) obtained with the integral polynomial expansion recovery algorithm.

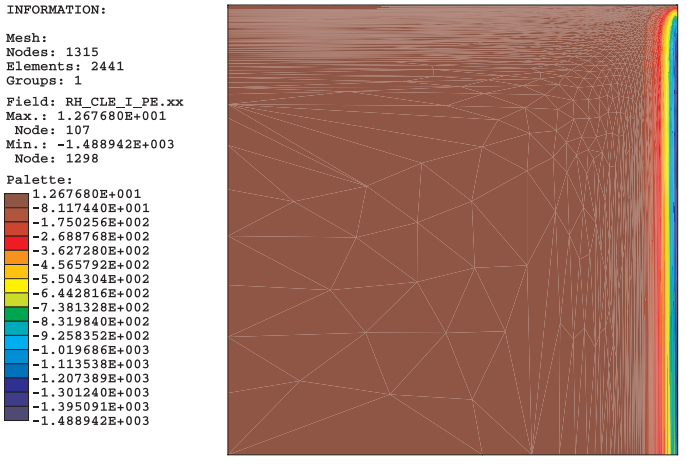

Fig. 11. Function $u(x,y) = (1-x^{40})(1-y^{40})$. xx-component of the recovered Hessian using the integral polynomial expansion technique.

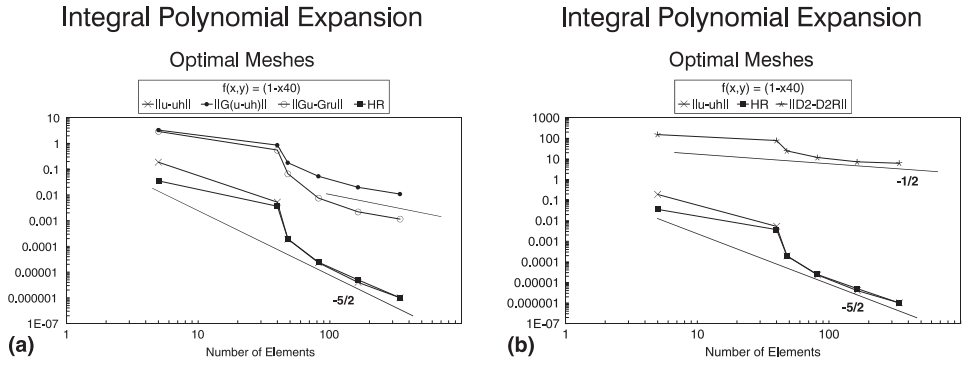

Fig. 12. Optimal mesh adaptive process for function $u(x,y) = (1 - x^{40})$.

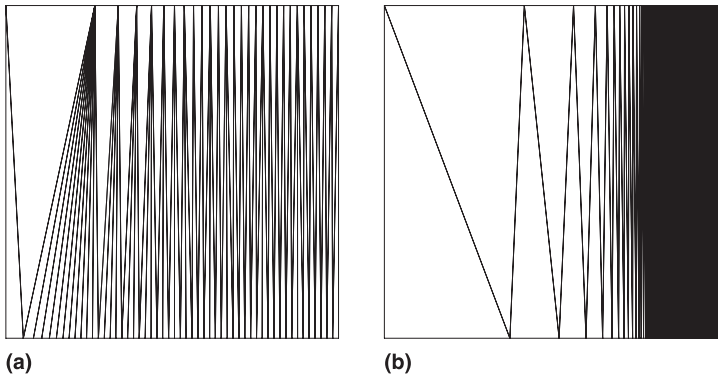

Fig. 13. First ($\theta = 0.6930$) and last ($\theta = 1.0000$) optimal meshes for function $u(x,y) = (1 - x^{40})$.

For the function $u(x,y) = (1 - x^{40})$ we plot the results in Figs. 12 and 13. In the former figure we take as number of elements the finite element partition along the x-axis. The first mesh corresponds to a uniform mesh with five elements over the x-axis and the adapted meshes have 40, 48, 82, 164 and 342 elements. In the last adapted mesh we have $s_T \in [1;10{,}000]$ and the effectivity indices are shown in Fig. 13 for both meshes.

Fig. 14 shows the comparison between uniform and adapted meshes using the integral polynomial expansion.

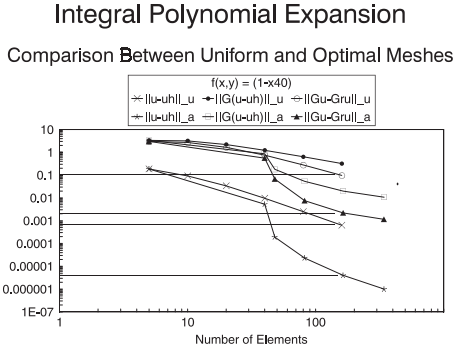

Fig. 14. Comparison between uniform and adapted meshes for function $u(x,y) = (1 - x^{40})$.

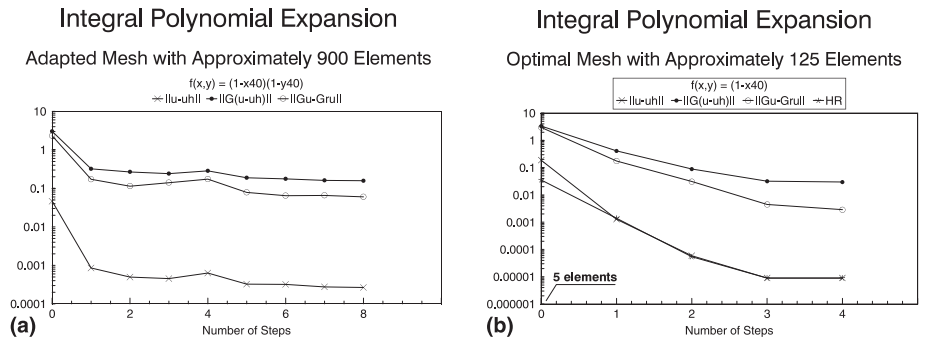

Fig. 15. Performance of the optimal-mesh-adaptive algorithm with approximately constant number of elements.

4.1.3. Adapted meshes with (approximately) constant number of elements

Another interesting result is obtained when trying to maintain a constant number of elements during the optimal-mesh-adaptive process. In this case, Fig. 15 shows the performance of the optimal-mesh-adaptive algorithm for the two analyzed functions when the integral polynomial expansion recovery algorithm is adopted.

4.2. Convection–diffusion problems

In this section some numerical results obtained by applying the proposed methodology to the solution of scalar transport problems are presented. For all examples, the velocity field $\mathbf{v}$ is constant ($\|\mathbf{v}\| = \mathbf{1}$), the medium is assumed to be homogeneous with a constant physical diffusivity tensor, and no reaction term is considered. Moreover, all solutions are obtained with the well-known SUPG [9] method.

4.2.1. One-dimensional example

In this one-dimensional example, though numerically solved in the square domain $\Omega = (0,1) \times (0,1)$, the source term is null ($f = 0$), the physical diffusivity is $k = 10^{-3}$, the principal boundary conditions are

$$u(0,y) = 0, \quad u(1,y) = 1,$$

and the natural boundary conditions are

$$\left.\frac{\partial u}{\partial y}\right|_{(x,0)} = 0, \quad \left.\frac{\partial u}{\partial y}\right|_{(x,1)} = 0.$$

The exact solution, $u(x,y) = (1 - \mathrm{e}^{x/k})/(1 - \mathrm{e}^{1/k})$, has a boundary layer of width k at $x \simeq 1$. For this problem two approaches have been used. The first, keeping approximately constant the number of elements around 400 elements, and the second, doubling the number for each new mesh. Fig. 16 shows the first *uniform* mesh with 394 elements, the *optimal* adapted mesh with approximately the same number of elements (366 elements) obtained at the end of step 5 and the final mesh (1515 elements, step 3) obtained by doubling the number of elements. The stretching is $s_T \in [1; 20{,}000]$ for the first case and $s_T \in [1;10{,}000]$ for the second case.

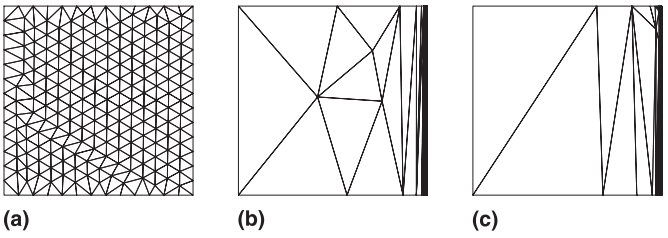

Fig. 16. One-dimensional problem: (a) uniform, (b) constant adapted (step 5), and (c) doubling adapted meshes (step 3).

Fig. 17 plots the solution and the x-component of the recovered gradient obtained with the optimal adapted mesh with 1515 elements. Fig. 18 shows the behavior of the anisotropic error estimator correspondent to Eqs. (13) and (14).The effectivity indices for the first mesh (394 elements) and the final adapted mesh (1515 elements) are 0.1356 and 1.2907, respectively.

4.2.2. Internal layer example

For this example we have $\Omega = (0,1) \times (0,1)$, $f = 0$, $k = 10^{-6}$ and the velocity field direction is 45°. The principal boundary conditions are

$$u(x,0) = 1,$$

$$u(0,y) = \begin{cases} 1 & \text{if } 0 \leqslant y \leqslant 0.2, \\ -5y + 2 & \text{if } 0.2 \leqslant y \leqslant 0.4, \\ 0 & \text{if } y \geqslant 0.4. \end{cases}$$

In this example we use the result obtained with a uniform mesh (224 elements) and apply the adaptive procedure to build a sequence of optimal meshes with 10, 45, 67 and 88 elements. The former mesh has $s_T \in [0; 300]$. Fig. 19 shows the uniform mesh, the mesh with 10 elements and the mesh with 88 elements. The solutions for the three meshes are shown in Fig. 20 where a zoom (square of edge length 0.137132 in the middle of the layer at $x = 0.198$) in the last mesh is also plotted to depict the quality of the solution. Fig. 21 plots the behavior of the anisotropic error estimator vs. the number of elements (both in log scale). Finally,

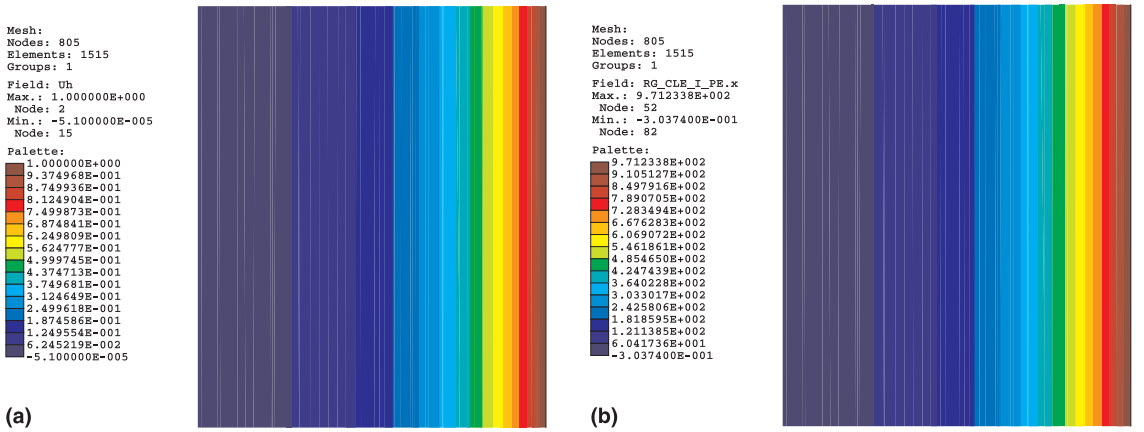

Fig. 17. One-dimensional problem: (a) solution, and (b) x-component of the recovered gradient.

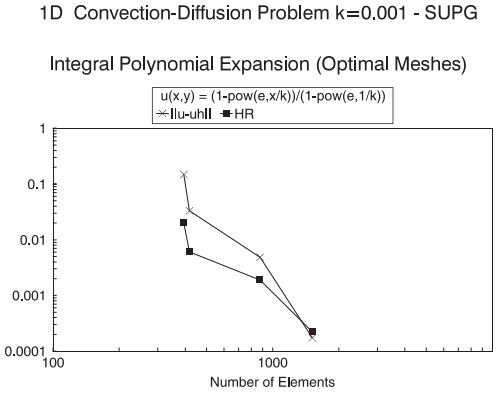

Fig. 18. One-dimensional problem. Anisotropic error estimator behavior.

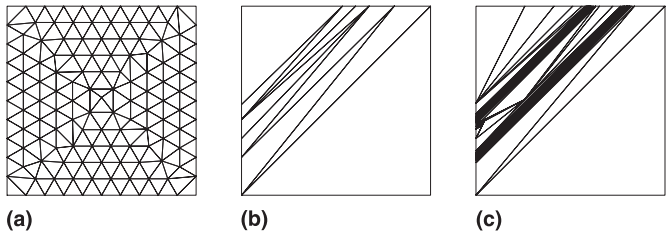

Fig. 19. Internal layer example: (a) initial uniform mesh, (b) first, and (c) last optimal meshes.

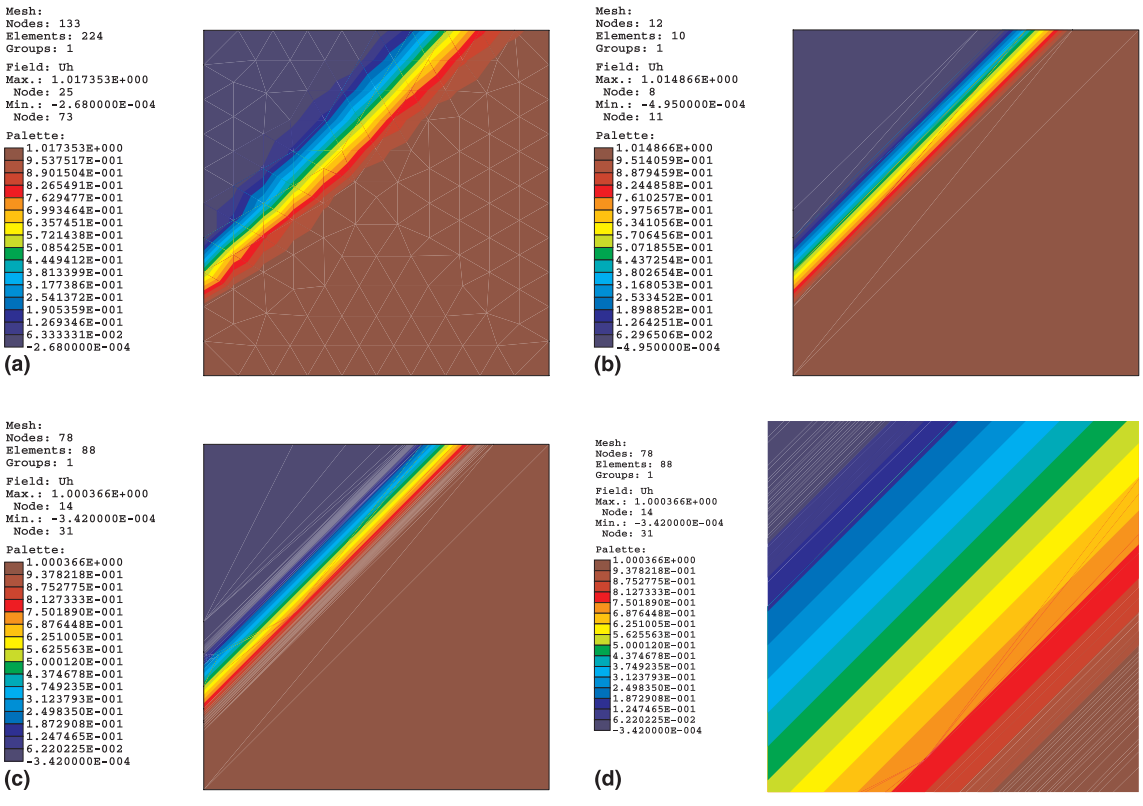

Fig. 20. Internal layer example. Solutions for the: (a) uniform, (b) first, (c) last optimal meshes, and (d) zoom for the last optimal mesh.

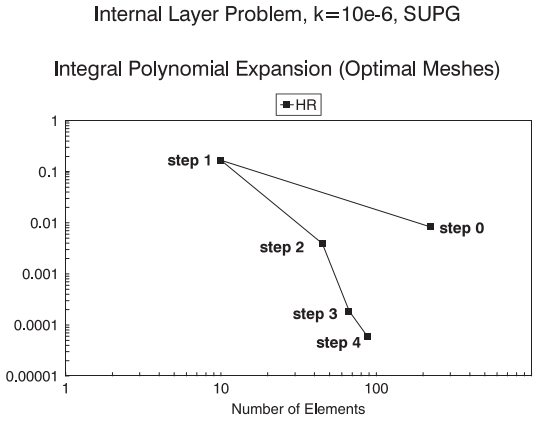

Fig. 21. Internal layer problem. Error estimator behavior.

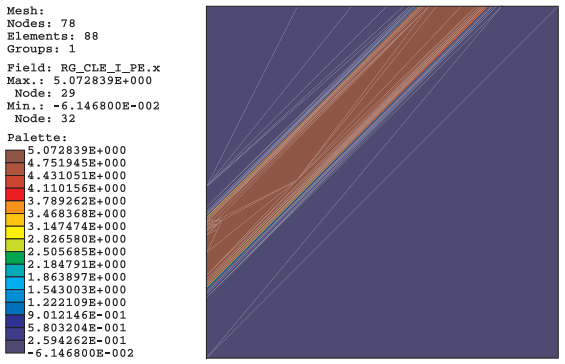

Fig. 22. Internal layer problem. x-Component of the recovered gradient using the integral polynomial expansion algorithm.

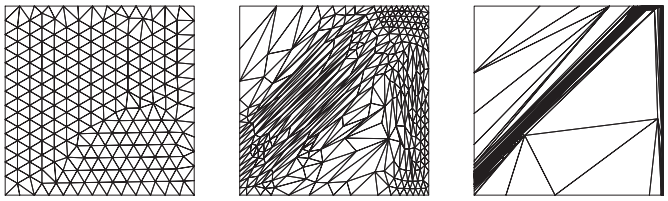

Fig. 23. External layer example: (a) initial uniform mesh, (b) first, and (c) last optimal meshes.

the x-component of the recovered gradient obtained using the integral polynomial expansion algorithm is shown in Fig. 22.

4.2.3. External layer example

For this example, we have $\Omega = (0,1) \times (0,1)$, $f = 0$, the velocity field direction is 45° and $k = 10^{-6}$. The principal boundary conditions are

$$u(x,0) = 1 \quad \forall x \in [0,1), \quad u(x,1) = 0, \quad u(1,y) = 0,$$
$$u(0,y) = \begin{cases} 1 & \text{if } 0 \leqslant y, \\ 0 & \text{if } y \geqslant 0.1. \end{cases}$$

The solution exhibits two strong external boundary layers along the lines $x = 1$ and $(0.9,1)$–$(1,1)$ and a strong internal boundary layer along the line $(0,0.1)$–$(0.9,1)$. The adaptivity process initiates with a uniform mesh with 396 elements. The first optimal mesh is built by setting approximately the same number of elements (544), and the others are generated by roughly doubling the number of elements for each new mesh. The uniform, first and last (step 5) optimal meshes are presented in Fig. 23. The last optimal mesh has 4813 elements and $s_T \in [1,4700]$. The approximate solution obtained with this mesh is shown in Fig. 24 (the associated anisotropic error estimator takes the value $\eta = 0.0019$). This figure also shows a zoom (an square of edge length 0.0004) in the middle point of the boundary $x = 1$ (the value of the solution u along the left edge of this square is $u = 0.9999925$).

4.2.4. The "double ramp" example

For this example we have $f = 1$, $k = 0.001$ and homogeneous boundary Dirichlet conditions are assumed. The domain depicted in Fig. 25 is contained in a square of edge length equal to 4. The solution exhibits a strong boundary layer along $x = 4$, two crosswind layers along $y = 0$ and $y = 4$ and a crosswind internal layer along $y = 2$. We initiate the adaptive process with a uniform mesh with 568 elements, and the first adapted mesh (618 elements) is obtained keeping approximately constant the number of elements. Roughly doubling the number of elements, the last adapted mesh (step 4) reaches 4887 elements. The

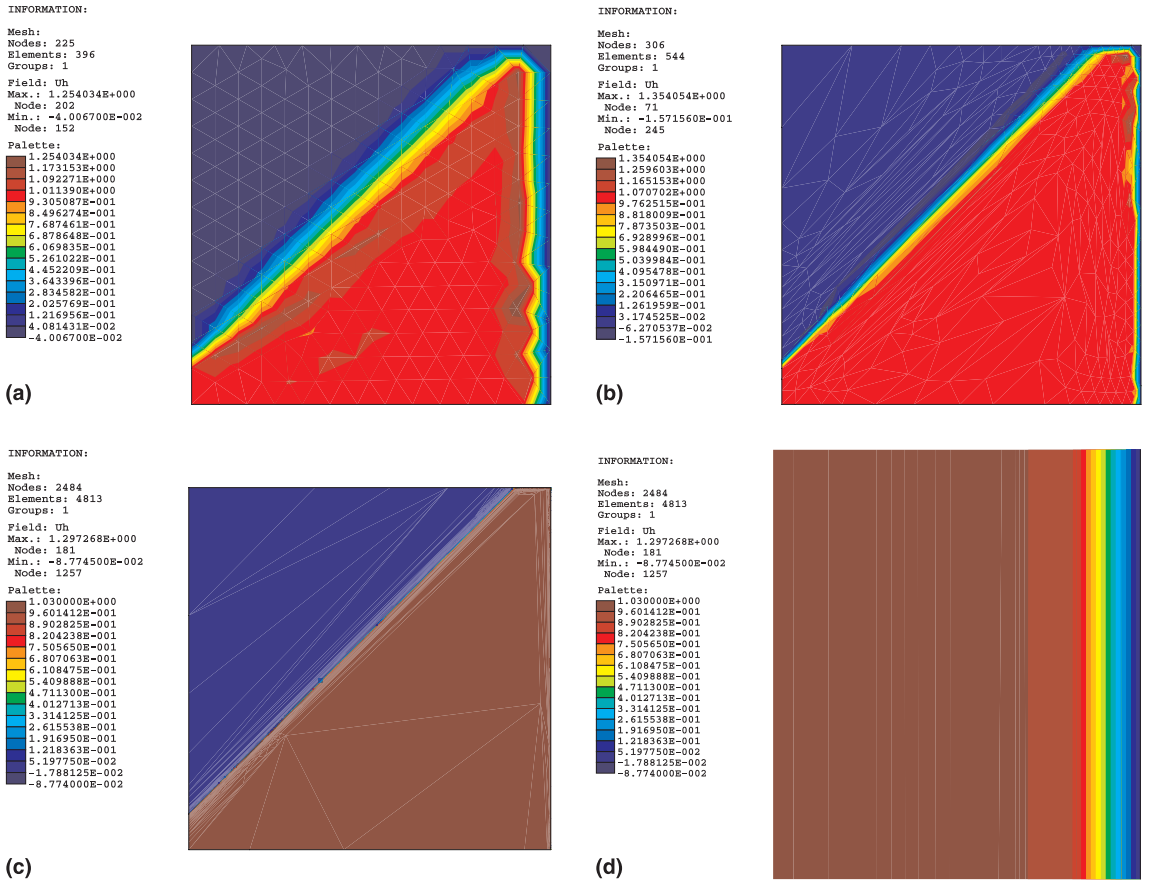

Fig. 24. External layer example. Solutions for the: (a) initial uniform mesh, (b) first, (c) last optimal meshes, and (d) zoom.

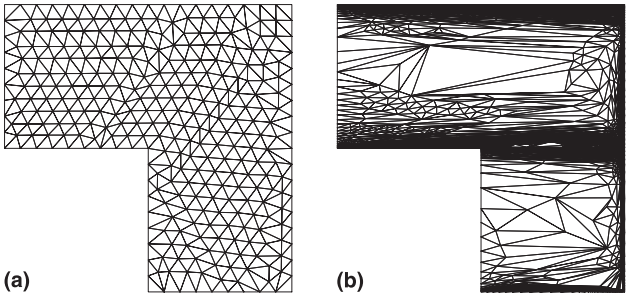

Fig. 25. "Double ramp" example: (a) initial uniform mesh, and (b) last adapted mesh.

uniform and the last adapted meshes are presented in Fig. 25, and the associated finite element solutions are plotted in Fig. 26.

4.3. Compressible flows problems

In this section some numerical results obtained by applying the proposed methodology for the solution of Euler and Navier–Stokes compressible flows are presented. For both problems the finite element approximations were obtained using the CAU method (see [1] and papers therein). Since for these problems, instead of a scalar field we have a vector field of unknowns, a question arises: *how can we extend the*

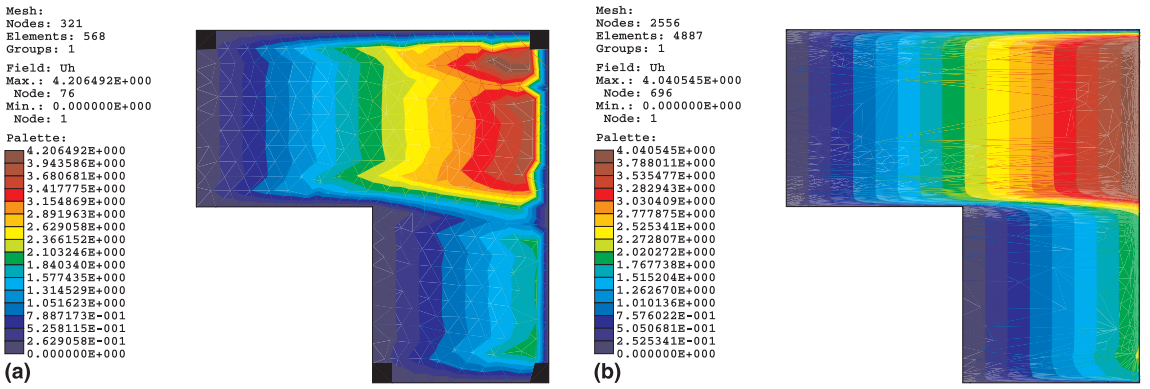

Fig. 26. "Double ramp" example. Solutions for the: (a) uniform, and (b) last adapted meshes.

optimal-mesh-adaptive algorithm to fit this problem? A very simple idea consists of taking one component of the vector field as the scalar field needed for our algorithm. The numerical results shown below, obtained with this simple approach, are quite acceptable (see also [6,16] for applications in Limit Analysis). Nevertheless, in a forthcoming paper, we will consider the idea of generalizing our error estimator (13) and (14) in order to take into account all vector field components simultaneously (see also [12]).

4.3.1. Flow over a wedge

This problem deals with an inviscid supersonic flow (Mach number $M = 2$) over a wedge at an angle of 10°. The domain is $\Omega = (0,1) \times (0,1)$. At the inflow ($x = 0$) the prescribed variable values are: $\rho = 1.0$ (density), $\|v\| = 1.0$ (velocity), $p = 0.17857$ (pressure), and the slip condition was set along the wedge ($y = 0$). The outflow exact solution yields $\rho = 1.45843$, $v_x = 0.88731$, $v_y = 0.0$, $p = 0.30475$, $M = 1.64$. Taking the density as the key scalar field, the adapted meshes and associated solutions are depicted in Figs. 27 and 28. In particular, Fig. 27 shows the first uniform mesh (394 elements) and the last adapted mesh (step 3, 1466 elements and $s_T \in [1; 800]$). For these two meshes the density solutions are respectively plotted in Figs. 28a and b.

4.3.2. Supersonic flow over a flat plate

This classical example called Carter's problem, consists of a two-dimensional Mach three viscous flow over a semi-infinite flat plate at zero angle of attack. Due to the symmetry we consider only $\Omega = (0, 1.4) \times (0, 1.4)$. The undisturbed flow is set at the inflow and at the top of the domain and is given by: $\rho = 1.0$, $\|v\| = 1.0$, $T = 2.769 \times 10^{-4}$ (temperature), $R = 1000$ (Reynolds number). The stagnation temperature is prescribed on the plate, and Sutherland's law is employed to model the dependence of viscosity on temperature. Taking the density as the key scalar field, the adapted meshes and the corresponding solutions are depicted in Figs. 29 and 30. In particular Fig. 29 shows the first uniform mesh

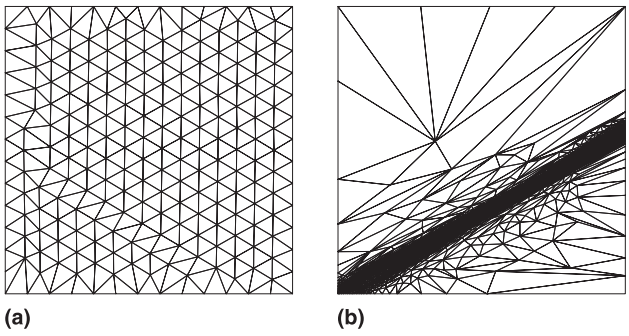

Fig. 27. Flow over a wedge: (a) initial uniform mesh (394 elements), and (b) last adapted mesh (1466 elements).

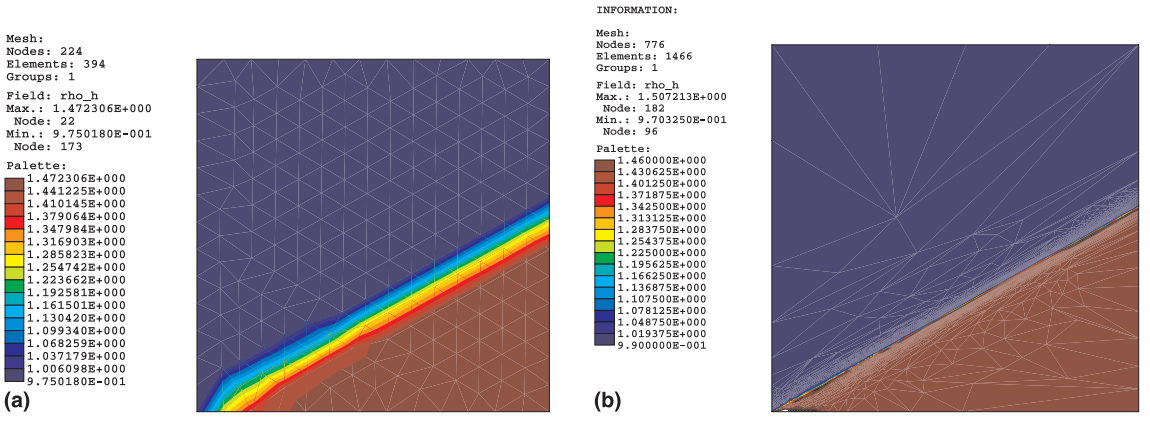

Fig. 28. Flow over a wedge. Density solution for the: (a) uniform, and (b) last adapted meshes.

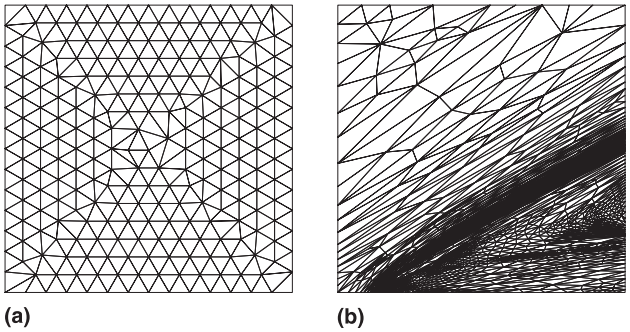

Fig. 29. Supersonic flow over a flat plate: (a) initial uniform mesh (442 elements), and (b) last adapted mesh (2820 elements).

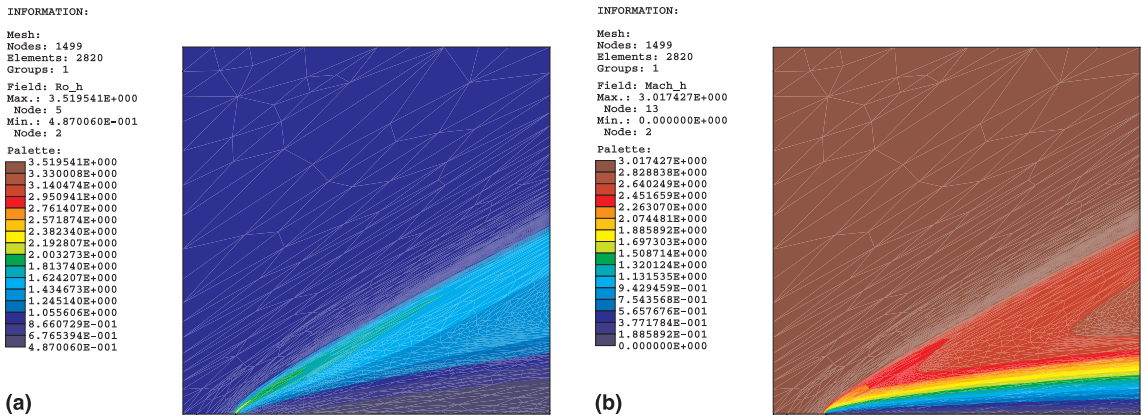

Fig. 30. Supersonic flow over a flat plate: (a) density, and (b) Mach fields for the last adapted mesh.

(442 elements) and the last adapted mesh (step 4, 2820 elements, $s_T \in [1;630]$). Fig. 30 shows the distribution of the density and Mach fields for the last adapted mesh.

5. Conclusions

In this work a new a posteriori anisotropic error estimator is derived using an approximated Hessian recovered operator. We also present an optimal-mesh-adaptive procedure. Construction of this procedure

relies on a constraint minimization problem for which the objective function is an upper bound of the error estimator.

Several numerical applications show that the proposed anisotropic error estimator provides an accurate representation of the exact error. Moreover, for several convection-dominated problems as well as supersonic compressible flows, it was shown that the optimal-mesh-adaptive procedure provides a mesh refinement and element stretching which appropriately capture boundary and internal layers. As a result, highly accurate solutions are obtained keeping the number of unknowns affordably low.

Another interesting aspect of the proposed optimal-mesh-adaptive procedure is that it is *problem independent*, i.e., the algorithm can be easily incorporated in any finite element solver to perform an adaptive analysis (see also [6]] for applications in Limit Analysis).

Otherwise, despite of the fact that we use an *advancing front* mesh generator, our approach can be used together with any unstructured mesh generation that takes into account a local metric $\mathbf{S}$ (see expression 8))) defined on the *old* mesh taken as a background mesh for the adaptive procedure.

Future works are under way for the mathematical justification of the assumption (4) and (5); extension to vector fields unknowns; time-dependent problems and three-dimensional applications.

Acknowledgements

This work was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and by PRONEX LNCC-FINEP Project no. 007/97. Claudio Padra was partially supported by PIA 7116 Conicet-Argentina. The authors are grateful for the software facilities provided by the TACSOM Group (http://www.lncc.br/~tacsom).

References

[1] R. Almeida, A.C. Galeão, An adaptive Petrov–Galerkin formulation for the compressible Euler and Navier–Stokes equations, Comput. Meth. Appl. Mech. Eng. 129 (1996) 157–176.
[2] I. Babuška, A. Aziz, On the angle condition in the finite element method, SIAM J. Numer. Anal. 13 (1976) 214–226.
[3] I. Babuška, W.C. Rheinboldt, A posteriori error estimators in the finite element method, Int. J. Numer. Meth. Eng. 12 (1978) 1597–1615.
[4] I. Babuška, W.C. Rheinboldt, Error for adaptive finite element method, SIAM J. Numer. Anal. 15 (1978) 736–754.
[5] I. Babuška, R. Rodríguez, The problem of the selection of an a-posteriori error indicator based on smoothening techniques, Int. J. Numer. Meth. Eng. 36 (1993) 536–567.
[6] L. Borges, R.A. Feijóo, C. Padra, N. Zouain, A directional error estimator for adaptive finite element analysis, in: E. Oñate, S. Idelsohn (Eds.), Computational Mechanics: New Trends and Applications, Proceedings of the fourth World Congress on Computational Mechanics, Buenos Aires, Argentina, CIMNE, Barcelona, Spain, June 1998.
[7] B. Boroomand, O.C. Zienkiewicz, Recovery by equilibrium in patches (REP), Int. J. Numer. Meth. Eng. 40 (1997) 137–164.
[8] B. Boroomand, O.C. Zienkiewicz, An improved (REP) recovery and the effectivity robustness test, Int. J. Numer. Meth. Eng. 40 (1997) 3247–3277.
[9] A.N. Brooks, T.J.R. Hughes, Streamline upwind Petrov–Galerkin formulations for convection-dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Meth. Appl. Mech. Eng. 32 (1982) 199–259.
[10] G.C. Buscaglia, E.A. Dari, Adaptive anisotropic mesh optimization: construction of solution adapted finite element meshes, Personal communication, 1996.
[11] G.C. Buscaglia, E.A. Dari, Anisotropic mesh optimization and its applications in adaptivity, Int. J. Numer. Meth. Eng. 40 (1997) 4119–4136.
[12] M.J. Castro-Díaz, F. Hecht, B. Mohammadi, O. Pironneau, Anisotropic unstructered mesh adaption for flow simulations, Int. J. Numer. Meth. Eng. 25 (1997) 475–491.
[13] P. Clément, Approximation by finite element functions using local regularization, RAIRO Anal. Numér. R-2 (1975) 77–84.
[14] J. Domppierre, M.G. Vallet, M. Fortin, W.G. Habashi, D. Aït-Ali-Yahia, S. Boivin, Y. Bourgault, A. Tam, Edge-based mesh adaptation for CFD, in: Conference on Numerical Methods for the Euler and Navier–Stokes Equations, Montreal, 1995, pp. 265–299.
[15] E.A. Fancello, A.C.S. Guimarães, R.A. Feijóo, M.J. Vénere, Automatic two-dimensional mesh generation using object-oriented programming, in: São Paulo (Ed.), Proceedings of 11th Brazilian Congress of Mechanical Engineering, Brazilian Association of Mechanical Sciences, December 1991, pp. 635–638.
[16] R.A. Feijóo, L. Borges, N. Zouain, Estimadores a posteriori y sus aplicaciones en el análisis adaptativo, Research Report, no. 01/97, Programa de Engenharia Mecânica, COPPE, Univ. Federal de Rio de Janeiro, Brasil, 1997.

[17] P. Grisvard, Elliptic Problems in Nonsmooth Domains, Pitman, Boston, 1985.
[18] D.L. Marcum, N.P. Weatherill, M.J. Charmant, F. Beaven, Adaptive unstructured grid generation for viscous flow applications, in: Proceedings of the 12th AIAA Comp. Fluid Dyn. Conf., AIAA Paper 95–1726CP, San Diego, USA, 1995.
[19] T.J.R. Hughes, L.P. Franca, M. Mallet, A new finite element formulation for computational fluid dynamics, Part VI: Convergence analysis of the generalized SUPG formulation for linear time-dependent multidimensional advective–diffusive systems, Comput. Meth. Appl. Mech. Eng. 63 (1997) 97–112.
[20] D.J. Mavriplis, Adaptive mesh generation for viscous flows using Delaunay triangulation, J. Comput. Phys. 90 (1990) 271–291.
[21] M. de Oliveira, A.C.S. Guimarães, R.A. Feijóo, M.J. Vénere, E.A. Dari, An object oriented tool for automatic surface mesh generation using the advancing front technique, Int. J. Latin America Appl. Res. 27 (1997) 39–49.
[22] J. Peiró, A finite element procedure for the solution of the Euler equations on unstructured meshes, Ph.D. Thesis, Department of Civil Engineering, University of Swansea, UK, 1989.
[23] J. Peraire, K. Morgan, J. Peiró, Unstructured mesh methods for CFD, I.C. Aero Report no. 90-04, Imperial College, UK, 1990.
[24] J. Peraire, J. Peiró, K. Morgan, Adaptive remeshing for three-dimensional compressible flow computations, J. Comput. Phys. 103 (1992) 269–285.
[25] S. Rippa, Long and thin triangles can be good for linear interpolation, SIAM J. Numer. Anal. 29 (1992) 257–270.
[26] N.-E. Wiberg, F. Abdulwahab, Patch recovery base on superconvergent derivatives and equilibrium, Int. J. Numer. Meth. Eng. 36 (1993) 2703–2724.
[27] N.-E. Wiberg, F. Abdulwahab, S. Ziukas, Enhanced superconvergent patch recovery incorporating equilibrium and boundary conditions, Int. J. Numer. Meth. Eng. 37 (1994) 3417–3440.
[28] O.C. Zienkiewicz, R.L. Taylor, The Finite Element Method, 4th ed., McGraw-Hill, New York, 1991.
[29] O.C. Zienkiewicz, J.Z. Zhu, A simple error estimator and adaptive procedure for practical engineering analysis, Int. J. Numer. Meth. Eng. 24 (1987) 337–357.
[30] O.C. Zienkiewicz, J.Z. Zhu, Superconvergent recovery technique and a posteriori error estimators, Int. J. Numer. Meth. Eng. 30 (1990) 1321–1339.
[31] O.C. Zienkiewicz, J.Z. Zhu, The superconvergent patch recovery and a posteriori error estimator. Part 1. The recovery technique, Int. J. Numer. Meth. Eng. 33 (1992) 1331–1364.
[32] O.C. Zienkiewicz, J.Z. Zhu, The superconvergent patch recovery and a posteriori error estimator. Part 2. Error estimates and adaptivity, Int. J. Numer. Meth. Eng. 33 (1992) 1365–1382.
[33] O.C. Zienkiewicz, J.Z. Zhu, The superconvergent patch recovery (SPR) and adaptive finite element refinement, Comput. Meth. Appl. Mech. Eng. 101 (1992) 207–224.
[34] O.C. Zienkiewicz, J.Z. Zhu, Superconvergent patch recovery techniques – some further tests, Commun. Appl. Numer. Meth. 9 (1993) 251–258.