# Anisotropic, Adaptive Finite Elements for a Thin 3D Plate

Marco Picasso, Adrien Loseille

# Anisotropic, Adaptive Finite Elements for a Thin 3D Plate

**Marco Picasso and Adrien Loseille**

**Abstract** An adaptive, anisotropic finite element algorithm is proposed to solve the 3D linear elasticity equations in a thin 3D plate. Numerical experiments show that adaptive computations can be performed in thin 3D domains having geometrical aspect ratio 1:1000.

## 1 The Linear Elasticity Model and the Numerical Method

Anisotropic adaptive algorithms are now widely used to solve complex systems based on partial differential equations, see for instance [5, 6, 11, 16]. Our goal is to experiment such techniques for the 3D linear elasticity system, the computational domain being a thin 3D plate.

Let $\Omega$ be the reference configuration of a bounded, polyhedral, elastic body of $\mathbb{R}^3$, $\partial\Omega = \Gamma_D \cup \Gamma_N$, $\Gamma_D$ not empty, $\Gamma_D \cap \Gamma_N = \emptyset$, $n$ the unit outer normal of $\partial\Omega$. Given $f \in L^2(\Omega)^3$, given the positive Lamé coefficients $\lambda, \mu$, we are looking for a displacement vector $u = (u_1, u_2, u_3)^T$ and a symmetric stress tensor $\sigma$ such that

$$
\begin{aligned}
&-\operatorname{div}\sigma = f && \text{in } \Omega, && (1)\\
&\sigma = 2\mu D(u) + \lambda \operatorname{div} u\, I && \text{in } \Omega, && (2)\\
&u = 0 && \text{on } \Gamma_D, &&\\
&\sigma n = 0 && \text{on } \Gamma_N. &&
\end{aligned}
$$

Hereabove, we have used the notation $D_{ij}(u) = \frac{1}{2}(\partial u_i/\partial x_j + \partial u_j/\partial x_i)$, and $I$ is the unit $3 \times 3$ tensor. Introducing $V = \{v \in H^1(\Omega)^3, v = 0 \text{ on } \Gamma_D\}$, the displacement

M. Picasso (✉)
EPFL, MATHICSE, Station 8, 1015 Lausanne, Switzerland
e-mail: marco.picasso@epfl.ch

A. Loseille
INRIA Paris-Rocquencourt, Gamma3, 78153 Le Chesnay, France
e-mail: adrien.loseille@inria.fr

S. Perotto, L. Formaggia (eds.), *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, SEMA SIMAI Springer Series 5,
DOI 10.1007/978-3-319-06053-8_11

weak form corresponding to this problem is to find $u \in V$ such that

$$\int_{\Omega} \Big( 2\mu D(u) : D(v) + \lambda \mathrm{div}\, u \; \mathrm{div}\, v \Big) dx = \int_{\Omega} f \cdot v dx \quad \forall v \in V, \tag{3}$$

where we have set $D(u) : D(v) = \sum_{i,j=1}^{3} D_{ij}(u) D_{ij}(v)$. Thanks to Korn's inequality

$$v \to \left( \sum_{i,j=1}^{3} \| D_{ij}(v) \|^2_{L^2(\Omega)} \right)^{1/2}$$

is a norm on $V$ and the above problem has a unique solution. For any $h > 0$, let $\mathscr{T}_h$ be a conforming mesh of $\overline{\Omega}$ into tetrahedrons $K$ with diameter $h_K$ less than $h$. Assume that the mesh is such that $\Gamma_D$ is the union of triangles lying on $\partial\Omega$. Let $V_h$ be the usual finite element space of continuous displacements having components that are linear on the tetrahedrons of $\mathscr{T}_h$, zero valued on $\Gamma_D$. Then, the Galerkin formulation corresponding to (3) is to find $u_h = (u_{1,h}, u_{2,h}, u_{3,h})^T \in V_h$ such that

$$\int_{\Omega} \Big( 2\mu D(u_h) : D(v_h) + \lambda \mathrm{div}\, u_h \; \mathrm{div}\, v_h \Big) dx = \int_{\Omega} f \cdot v_h \, dx \quad \forall v_h \in V_h. \tag{4}$$

The matrix of the linear system corresponding to (4) is symmetric positive definite so that the Conjugate Gradient (CG) method can be used. From Korn, Poincaré and the inverse inequalities, it can be shown that the number of iterations required to solve the linear system with a Jacobi preconditioner is $O(1/h)$, thus doubles when the mesh size is divided by two. Thus, the complexity is $O(1/h^4)$; it can be reduced to $O(1/h^3)$—which is optimal—when using multigrid as a preconditioner, this will not be the case in this paper.

Our goal is to consider the case when the computational domain $\Omega$ is a thin 3D volume—for instance $\Omega = (0,1) \times (0,1) \times (0,\varepsilon)$ with $\varepsilon$ small—and when anisotropic finite elements are used—that is tetrahedrons with large aspect ratio.

## 2 An Anisotropic Error Indicator

We now use the notations of [2, 3] in order to describe the mesh anisotropy, similar results can be found in [8]. For any tetrahedron $K$ of the mesh, let $T_K : \hat{K} \to K$ be the affine transformation which maps the reference tetrahedron $\hat{K}$ into $K$. Let $M_K$ be the Jacobian of $T_K$ that is

$$\mathbf{x} = T_K(\hat{\mathbf{x}}) = M_K \hat{\mathbf{x}} + \mathbf{t}_K.$$

Since $M_K$ is invertible, it admits a singular value decomposition $M_K = R_K^T \Lambda_K P_K$, where $R_K$ and $P_K$ are orthogonal and where $\Lambda_K$ is diagonal with positive entries. In the following we set

$$\Lambda_K = \begin{pmatrix} \lambda_{1,K} & 0 & 0 \\ 0 & \lambda_{2,K} & 0 \\ 0 & 0 & \lambda_{3,K} \end{pmatrix} \qquad \text{and} \qquad R_K = \begin{pmatrix} r_{1,K}^T \\ r_{2,K}^T \\ r_{3,K}^T \end{pmatrix}, \tag{5}$$

with the choice $\lambda_{1,K} \geq \lambda_{2,K} \geq \lambda_{3,K}$. In the frame of anisotropic meshes, the classical minimum angle condition is not required. However, for each vertex, the number of neighbouring vertices should be bounded from above, uniformly with respect to the mesh size $h$. Also, for each tetrahedron $K$ of the mesh, there is a restriction related to the patch $\Delta_K$, the set of tetrahedrons having a vertex common with $K$. More precisely, the diameter of the reference patch $\Delta_{\hat{K}}$, that is $\Delta_{\hat{K}} = T_K^{-1}(\Delta_K)$, must be uniformly bounded independently of the mesh geometry. This assumption excludes some too distorted reference patches, see [15]. This assumption is needed in order to prove the interpolation estimates—Clément's interpolant involves quantities on the reference patch—and implies that the local geometric quantities $\lambda_{i,K}, r_{i,K}, i = 1, 2, 3$, vary smoothly on neighbouring tetrahedrons. Two examples of admissible and non-admissible patches are presented in Fig. 1. The anisotropic mesh generator used in this paper has always produced admissible patches.

Let us now introduce our anisotropic error indicator. It is similar to the one presented in [14, 15] for the Laplace equation. For all $K \in \mathscr{T}_h$, let $\ell_{i,K}, i = 1, 2, 3, 4$ be the four faces of tetrahedron $K$, with unit normal $n_{i,K}$ (in arbitrary direction), let $[\cdot]$ denote the jump of the bracketed quantity across $\ell_{i,K}$, with the convention $[\cdot] = 0$ for a face $\ell_{i,K}$ on the boundary $\Gamma_D$. Then, our error indicator on tetrahedron $K$ is defined by

$$\eta_K^2 = \rho_K(u_h)\omega_K(e), \tag{6}$$

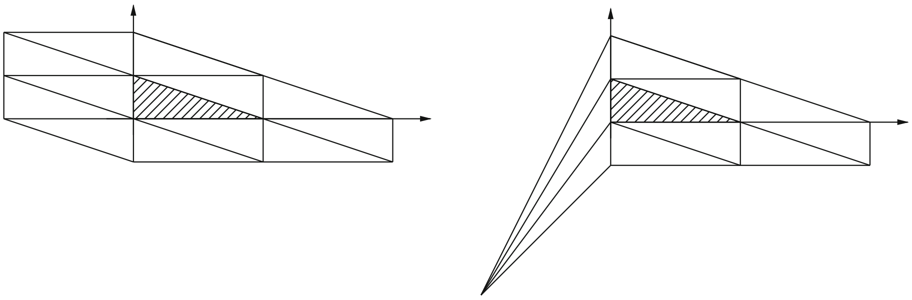

**Fig. 1** Admissible (*left*) and non-admissible (*right*) patches

where

$$\rho_K(u_h) = \frac{1}{2}\sum_{i=1}^{4}\left(\frac{|\ell_{i,K}|}{\lambda_{1,K}\lambda_{2,K}\lambda_{3,K}}\right)^{1/2}$$
$$\left(\int_{\ell_{i,K}}\left([\nabla u_{1,h}\cdot n_{i,K}]^2 + [\nabla u_{2,h}\cdot n_{i,K}]^2 + [\nabla u_{3,h}\cdot n_{i,K}]^2\right)dx\right)^{1/2},$$

where $e = (e_1, e_2, e_3)^T = u - u_h$ is the true error, and $\omega_K(e)$ is defined by

$$(\omega_K(e))^2 = \lambda_{1,K}^2\left(r_{1,K}^T G_K(e) r_{1,K}\right) + \lambda_{2,K}^2\left(r_{2,K}^T G_K(e) r_{2,K}\right) + \lambda_{3,K}^2\left(r_{3,K}^T G_K(e) r_{3,K}\right). \tag{7}$$

Here $G_K(e)$ denotes the $3\times 3$ matrix defined by

$$G_K(e) = \sum_{j=1}^{3}\begin{pmatrix} \int_{\Delta_K}\left(\frac{\partial e_j}{\partial x_1}\right)^2 dx & \int_{\Delta_K}\frac{\partial e_j}{\partial x_1}\frac{\partial e_j}{\partial x_2}dx & \int_{\Delta_K}\frac{\partial e_j}{\partial x_1}\frac{\partial e_j}{\partial x_3}dx \\ \int_{\Delta_K}\frac{\partial e_j}{\partial x_1}\frac{\partial e_j}{\partial x_2}dx & \int_{\Delta_K}\left(\frac{\partial e_j}{\partial x_2}\right)^2 dx & \int_{\Delta_K}\frac{\partial e_j}{\partial x_2}\frac{\partial e_j}{\partial x_3}dx \\ \int_{\Delta_K}\frac{\partial e_j}{\partial x_1}\frac{\partial e_j}{\partial x_3}dx & \int_{\Delta_K}\frac{\partial e_j}{\partial x_2}\frac{\partial e_j}{\partial x_3}dx & \int_{\Delta_K}\left(\frac{\partial e_j}{\partial x_3}\right)^2 dx \end{pmatrix}. \tag{8}$$

The indicator (6) is not a usual error estimator since $e = u - u_h$ (and therefore $u$) is still involved. However, the error $e$ can be estimated using post-processing techniques, so that (6) can be used to derive a computable quantity. An efficient anisotropic error indicator has been previously obtained replacing the derivatives

$$\frac{\partial e_j}{\partial x_i} \text{ in (8) by } \frac{\partial u_{j,h}}{\partial x_i} - \Pi_h\frac{\partial u_{j,h}}{\partial x_i}, \text{ i,j=1,2,3} \tag{9}$$

where $\Pi_h$ is an approximate $L^2(\Omega)$ projection onto $V_h$. More precisely, from constant values of $\partial u_{j,h}/\partial x_i$ on triangles, we build values at vertices $P$ using the formula

$$\Pi_h\left(\frac{\partial u_{j,h}}{\partial x_i}\right)(P) = \frac{1}{\sum\limits_{\substack{K\in\mathscr{T}_h\\ P\in K}}|K|}\sum_{\substack{K\in\mathscr{T}_h\\ P\in K}}|K|\left(\frac{\partial u_{j,h}}{\partial x_i}\right)_{|K} \qquad i,j=1,2,3.$$

Approximating $\partial e_j/\partial x_i$ by $(I - \Pi_h)\partial u_{j,h}/\partial x_i$ is at the base of the celebrated Zienkiewicz-Zhu error estimator and can be justified theoretically whenever superconvergence occurs, that is when $\nabla u_j - \Pi_h\nabla u_{j,h}$ convergences faster to zero than $\nabla u_j - \nabla u_{j,h}$. To our knowledge, the most recent and general result has been obtained for a second order elliptic problem and 2D mildly structured anisotropic meshes in [1]. In practice, on general 3D unstructured anisotropic meshes, superconvergence is not observed; however, $\nabla u_j - \Pi_h \nabla u_{j,h}$ is much smaller than $\nabla u_j - \nabla u_{j,h}$.

## 3 An Adaptive Algorithm

We have considered the anisotropic, adaptive algorithm presented in [14, 15], the goal being to build an anisotropic triangulation such that the estimated relative error is close to a preset tolerance *TOL*, namely

$$(1-\beta)TOL \leq \frac{\left(\sum_{K \in \mathcal{T}_h} \eta_K^2\right)^{1/2}}{\|\nabla u_h\|_{L^2(\Omega)}} \leq (1+\beta)TOL. \tag{10}$$

Here $\eta_K$ is defined by (6)–(8) and the post-processing (9) has been used in order to approximate the error gradient $G_K(e)$. Also, $0 < \beta < 1$ is the equidistribution parameter, $\beta = 0.25$ throughout the paper.

Our goal is to equidistribute locally the error in the three directions of stretching $r_{1,K}, r_{2,K}, r_{3,K}$, and to align the mesh along the eigenvectors of the matrix $G_K(e)$. In practice, all the meshes are generated using the `feflo` software [13] which requires a metric to be prescribed at the mesh vertices. The method used to build this metric is now described. For each vertex $P$ of the mesh, we compute

$$G_P(e) = \sum_{\substack{K \in \mathcal{T}_h \\ P \in K}} G_K(e),$$

where (9) has been used to estimate $G_K(e)$. We then compute an orthonormal basis $Q_P(e)$ of the eigenvectors of $G_P(e)$. Our goal is to align the tetrahedron around vertex $P$ with the eigenvectors of $G_P(e)$. The metric is then defined by

$$Q_P(e)^T \begin{pmatrix} \frac{1}{h_{1,P}^2} & 0 & 0 \\ 0 & \frac{1}{h_{2,P}^2} & 0 \\ 0 & 0 & \frac{1}{h_{3,P}^2} \end{pmatrix} Q_P(e), \tag{11}$$

where the desired mesh size at vertex $P$, $h_{1,P}$, $h_{2,P}$, $h_{3,P}$, is prescribed in order to satisfy (10). More precisely, we go back to (6) and split the estimator on triangle $K$ in the three directions of stretching corresponding to $r_{i,K}$ :

$$(\eta_{i,K})^4 = (\rho_K(u_h))^2 \lambda_{i,K}^2 \left( r_{i,K}^T G_K(e) r_{i,K} \right) \qquad i = 1,2,3,$$

and then compute the corresponding quantity at each vertex $P$ of the mesh

$$(\eta_{i,P})^4 = \sum_{\substack{K \in \mathscr{T}_h \\ P \in K}} (\eta_{i,K})^4 \qquad i = 1,2,3.$$

Let $N_P$ is the number of mesh vertices. Since

$$\sum_{i=1}^{3} \sum_{P \in \mathscr{T}_h} (\eta_{i,P})^4 = 4 \sum_{i=1}^{3} \sum_{K \in \mathscr{T}_h} (\eta_{i,K})^4 = 4 \sum_{K \in \mathscr{T}_h} (\eta_K)^4,$$

if

$$\left( \frac{4}{3N_P^2} \right)^{1/4} (1-\beta) TOL \|\nabla u_h\|_{L^2(\Omega)} \leq \eta_{i,P} \leq \left( \frac{4}{3N_P^2} \right)^{1/4} (1+\beta) TOL \|\nabla u_h\|_{L^2(\Omega)},$$

for $i = 1,2,3$ and for each vertex $P$ of the mesh, then (10) is satisfied. The desired mesh size at vertex $P$, $h_{1,P}$, $h_{2,P}$, $h_{3,P}$, is then computed as follows. If

$$\frac{4}{3N_P^2} (1-\beta)^4 TOL^4 \|\nabla u_h\|_{L^2(\Omega)}^4 > (\eta_{i,P})^4,$$

then the values of $h_{i,P}$ are set to $2\lambda_{i,P}$, $i = 1,2,3$, if

$$\frac{4}{3N_P^2} (1-\beta)^4 TOL^4 \|\nabla u_h\|_{L^2(\Omega)}^4 \leq (\eta_{i,p})^4 \leq \frac{4}{3N_P^2} (1+\beta)^4 TOL^4 \|\nabla u_h\|_{L^2(\Omega)}^4,$$

then the values of $h_{i,P}$ are set to $\lambda_{i,P}$, $i = 1,2,3$, if

$$(\eta_{i,P})^4 > \frac{4}{3N_P^2} (1+\beta)^4 TOL^4 \|\nabla u_h\|_{L^2(\Omega)}^4,$$

then the values of $h_{i,P}$ are set to $\lambda_{i,P}/2$, $i = 1,2,3$. Once new values of $h_{i,P}$ are obtained, the metric at each vertex is computed from (11) and the `feflo` software [13] is used to generate a new anisotropic mesh. The whole process is then repeated several times.

## 4 Anisotropic Mesh Generation

We give in the section some details on the algorithm and the mechanisms used in `feflo` [12] to refine the mesh according to the previous error estimate. The mesh generator fits the Riemannian metric framework of [4]. The goal is to generate a quasi-unit mesh with respect to the prescribed metric (11). The input of the mesh generator is the metric—a 3×3 symmetric positive definite matrix—thus we assume that, for each $x \in \Omega$, the metric $\mathcal{M}(x)$ is known.

The two fundamental operations in a mesh generator are the computation of length and volume. Let $E$ be the edge joining vertices $x_i$ and $x_j$. The length of $E$ and the volume of a tetrahedron $K$, with respect to the metric $\mathcal{M}$, are defined by:

$$\ell_{\mathcal{M}}(E) = \int_0^1 \sqrt{(x_j - x_i)^T \, \mathcal{M}(x_i + t(x_j - x_i)) \, (x_j - x_i)} \, \mathrm{d}t,$$

$$|K|_{\mathcal{M}} = \int_K \sqrt{\det(\mathcal{M}(x))} \, \mathrm{d}x.$$

From a discrete point view, the metric field needs to be interpolated [4] to compute an approximate edge length and tetrahedron volume. We consider a linear interpolation of the point-wise metric field $\mathcal{M}_i$ at the vertices $x_i$ of the mesh; the following approximations are then used:

$$\ell_{\mathcal{M}}(E) \approx \sqrt{(x_j - x_i)^T \, \mathcal{M}_i \, (x_j - x_i)} \, \frac{r-1}{r \ln(r)},$$

$$|K|_{\mathcal{M}} \approx \sqrt{\det\left(\frac{1}{4}\sum_{i=1}^{4} \mathcal{M}_i\right)} |K|,$$

where $|K|$ is the Euclidean volume of $K$ and $r$ stands for the ratio

$$\frac{\sqrt{(x_j - x_i)^T \, \mathcal{M}_i \, (x_j - x_i)}}{\sqrt{(x_j - x_i)^T \, \mathcal{M}_j \, (x_j - x_i)}},$$

see Example 1.3 in [9] for details. A mesh is said to be a unit-mesh with respect to $\mathcal{M}$ when the six edges of each tetrahedron $K$ satisfy

$$\ell_{\mathcal{M}}(E_i) \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right] \qquad i = 1, \ldots, 6, \tag{12}$$

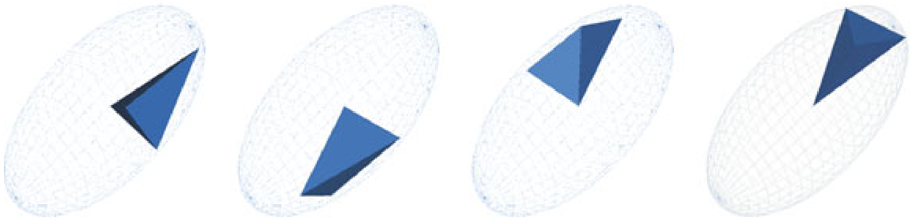

**Fig. 2** Some unit-elements with respect to a 3D metric represented by its unit-ball

and when the quality function $Q_{\mathcal{M}}$ satisfies

$$Q_{\mathcal{M}}(K) = \frac{36}{3^{\frac{1}{3}}} \frac{|K|_{\mathcal{M}}^{\frac{2}{3}}}{\displaystyle\sum_{i=1}^{6} \ell_{\mathcal{M}}^2(e_i)} \in [\alpha, 1], \tag{13}$$

where $\alpha > 0$ is a parameter. A classical and admissible value of $\alpha$ is 0.8. This value arises from some discussions on the possible tessellation of $\mathbb{R}^3$ with unit-elements [10]. Some unit-elements with respect to a 3D metric are depicted in Fig. 2.

To generate a unit-mesh in a given metric field $\mathcal{M}$, two operators are recursively used: edge collapse and point insertion on edge. The starting point for the insertion of a new point on an edge is to consider the shell of this edge composed of all elements sharing this edge. Each element of the shell is then divided into two new elements. The new point is accepted if each new tetrahedron has a positive volume. The edge collapse starts from the ball of the vertex to be deleted. Again, for the deletion of points inside the volume, the only possible rejection is the creation of a negative volume element.

We also combine the previous operators with a quality function $Q_{\mathcal{M}}$ together with the unit-length check. This supplementary check can be done at no cost since a lot of information can be re-used: the volume is already computed, as well as the length of the edges. By simply computing the quality function, we give to these operators the missing information on the orthogonal directions of the current scanned edge. For an optimal performance, two parameters are added in the rejection cases: a relative quality tolerance $q_r \geq 1$ and a global quality tolerance $q_a$. Rather than trying to increase $Q_{\mathcal{M}}$, a new configuration of elements is accepted if

$$q_r \, Q_{\mathcal{M}}^{ini} \leq Q_{\mathcal{M}}^{new} < q_a,$$

where $Q_{\mathcal{M}}^{ini}$ is the worse element quality of the initial configuration and $Q_{\mathcal{M}}^{new}$ is the worse quality of the new configuration. This approach is similar to the simulated annealing global optimization technique [7]. Note that the current version does not fully implement the classical metropolis algorithm where the rejection is based on a random probability. To ensure the convergence of the algorithm, the relative tolerance $q_r$ is decreased down to 1 after each pass of insertions and collapses. At the end of the process, the absolute tolerance $q_a$ is set up to the current worse quality among all elements.

This strategy yields a robust local remeshing procedure as a valid mesh is always provided on output. In particular, the volume and surface mesh generation are done simultaneously. Consequently, this procedure may be used advantageously for cases where global remeshing techniques become either unfeasible or unreliable. For the thin plate case, as a very high level of anisotropy is present on the surface, global remeshing approaches are likely to fail.

## 5 Numerical Experiments

Consider the case when $\Omega = (0,1) \times (0,1) \times (0,\varepsilon)$, with $\varepsilon = 0.01$ or $\varepsilon = 0.001$, $\Gamma_D$ are the lateral faces of the plate, $\Gamma_N$ the top and bottom faces corresponding to $z = 0$ and $z = \varepsilon$. Let $f = 1$ in (1), $\lambda = \mu = 1$ in (2). The exact solution is not known exactly but numerical experiments indicate that the maximum displacement up to three digits is between 0.0736 and 0.0737.

In Fig. 3, the mesh and vertical deformation are reported when $\varepsilon = 0.01$ or $\varepsilon = 0.001$ when using a $10 \times 10 \times 2$ mesh, each bloc being cup into six vertices. Starting from this $10 \times 10 \times 2$ mesh, convergence results with non-adapted meshes are presented in Table 1 when $\varepsilon = 0.01$. A locking effect can be observed when each bloc is cut into five tetrahedrons.

The adaptive procedure presented in the previous section is repeated 150 times, starting from the $10 \times 10 \times 2$ mesh considered above. At first, the *TOL* parameter which drives precision, see (10), is set to 1; it is halfed every 30 mesh generations, thus the adapted mesh number 29 corresponds to 30 mesh iterations with

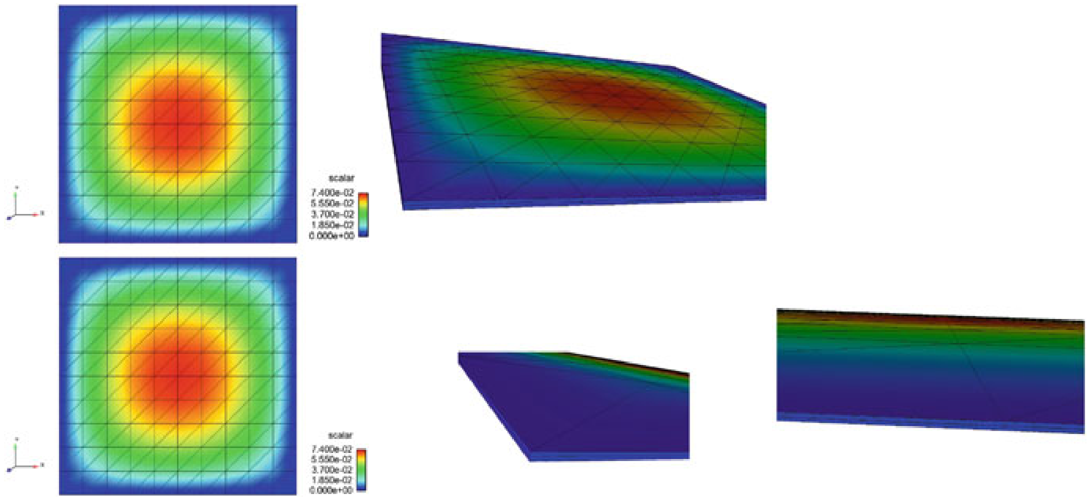

**Fig. 3** $10 \times 10 \times 2$ mesh, each bloc being cut in six tetrahedron, $\varepsilon = 0.01$ (*first row*) and $\varepsilon = 0.001$ (*second row*). *Left column*: xy plane, *Middle* and *Right column*: zoom

**Table 1** Convergence results with non-adapted meshes and $\varepsilon = 0.01$. Here tet/bloc denotes the number of tetrahedron per bloc

| Mesh | tet/bloc | nb. vert. | $CG$ | $u_{max}$ |
|---|---|---|---|---|
| $10 \times 10 \times 2$ | 6 | 363 | 103 | 0.0731 |
| $20 \times 20 \times 4$ | 6 | 2, 205 | 401 | 0.0735 |
| $40 \times 40 \times 8$ | 6 | 15, 129 | 1229 | 0.0736 |
| $80 \times 80 \times 16$ | 6 | 111, 537 | 2294 | 0.0737 |
| $10 \times 10 \times 2$ | 5 | 363 | 45 | 0.0129 |
| $20 \times 20 \times 4$ | 5 | 2, 205 | 229 | 0.0340 |
| $40 \times 40 \times 8$ | 5 | 15, 129 | 879 | 0.0571 |
| $80 \times 80 \times 16$ | 5 | 111, 537 | 2514 | 0.0687 |

Locking occurs when each bloc is cut into five tetrahedrons. CG denotes the number of iterations of the CG algorithm with diagonal preconditioner. The CG algorithm is stopped when the relative residual is less than $10^{-6}$

$TOL = 1$, the adapted mesh number 149 corresponds to 30 mesh iterations with $TOL = 0.0625$. The obtained adapted meshes corresponding to a plate thickness $\varepsilon = 0.01$ are shown in Fig. 4, those corresponding to a plate thickness $\varepsilon = 0.001$ are shown in Fig. 5. A careful examination of the adapted meshes reveals that there is only one layer of elements across the plate thickness and that mesh refinement is more important at the corners of the plate. Tables 2 and 3 contain the important numbers associated to these numerical experiments.

## 6 Conclusions

An adaptive, anisotropic finite element algorithm has been proposed to solve the 3D linear elasticity equations on a thin plate. Since the tetrahedrons are allowed to have large aspect ratio, the local mesh size can be small in the direction of the plate's thickness and coarse in the other directions. Numerical experiments show that adaptive computations can be performed on thin plates having geometrical aspect ratio 1:1000.

We are looking forward to perform numerical experiments on curved plates and/or plates having non homogenous thickness. In the case of curved plates, an error estimator for the geometry error has to be investigated. Also, the surface mesh should be reprojected onto the true geometry using the CAD data.

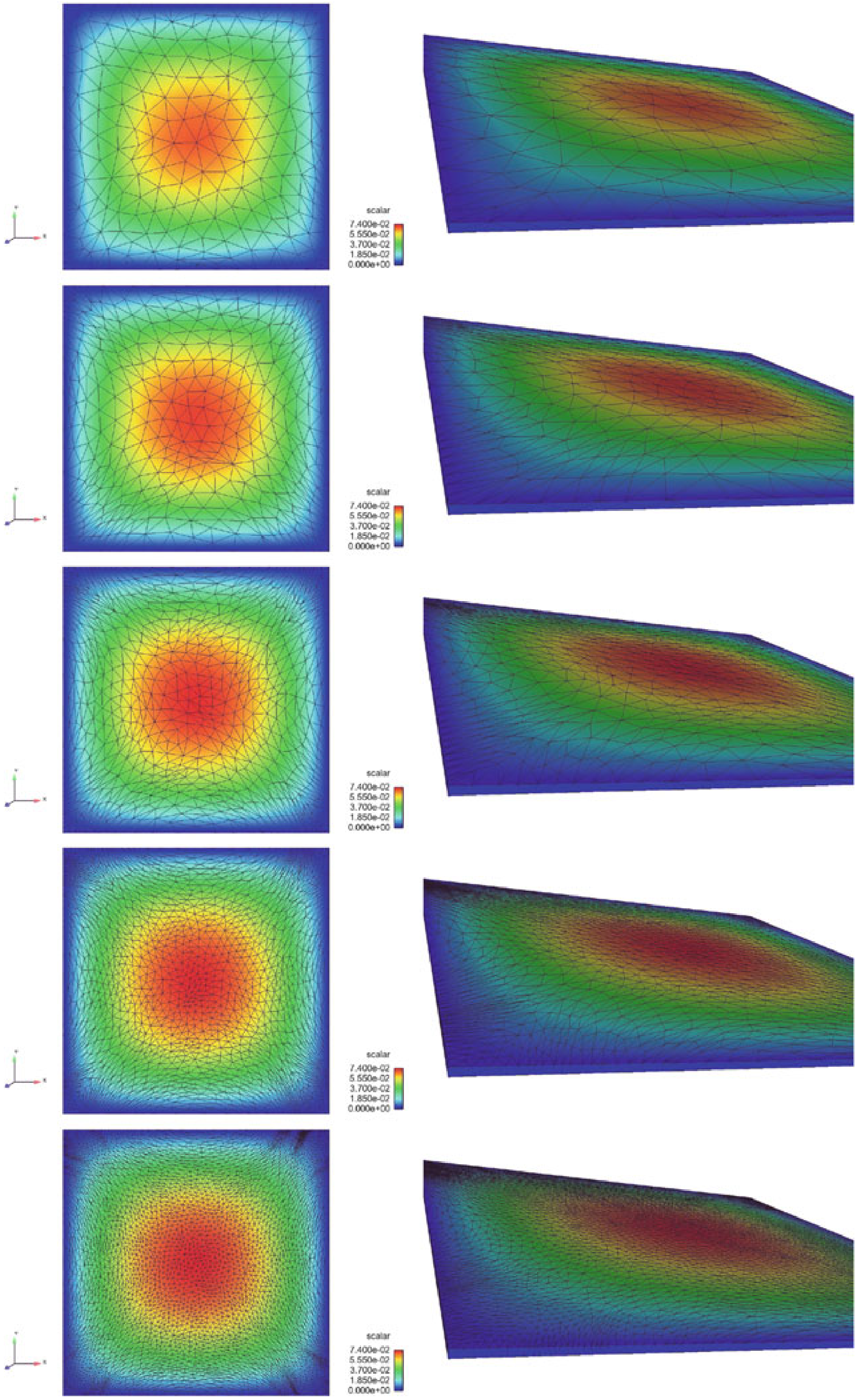

**Fig. 4** Adapted meshes with plate thickness $\varepsilon = 0.01$ and several values of *TOL* (*TOL* $= 1$ on row 1, *TOL* $= 0.5$ on row 2,..., *TOL* $= 0.0625$ on row 5). *Left column*: xy plane, *Right column*: zoom of the (0,0,0) corner

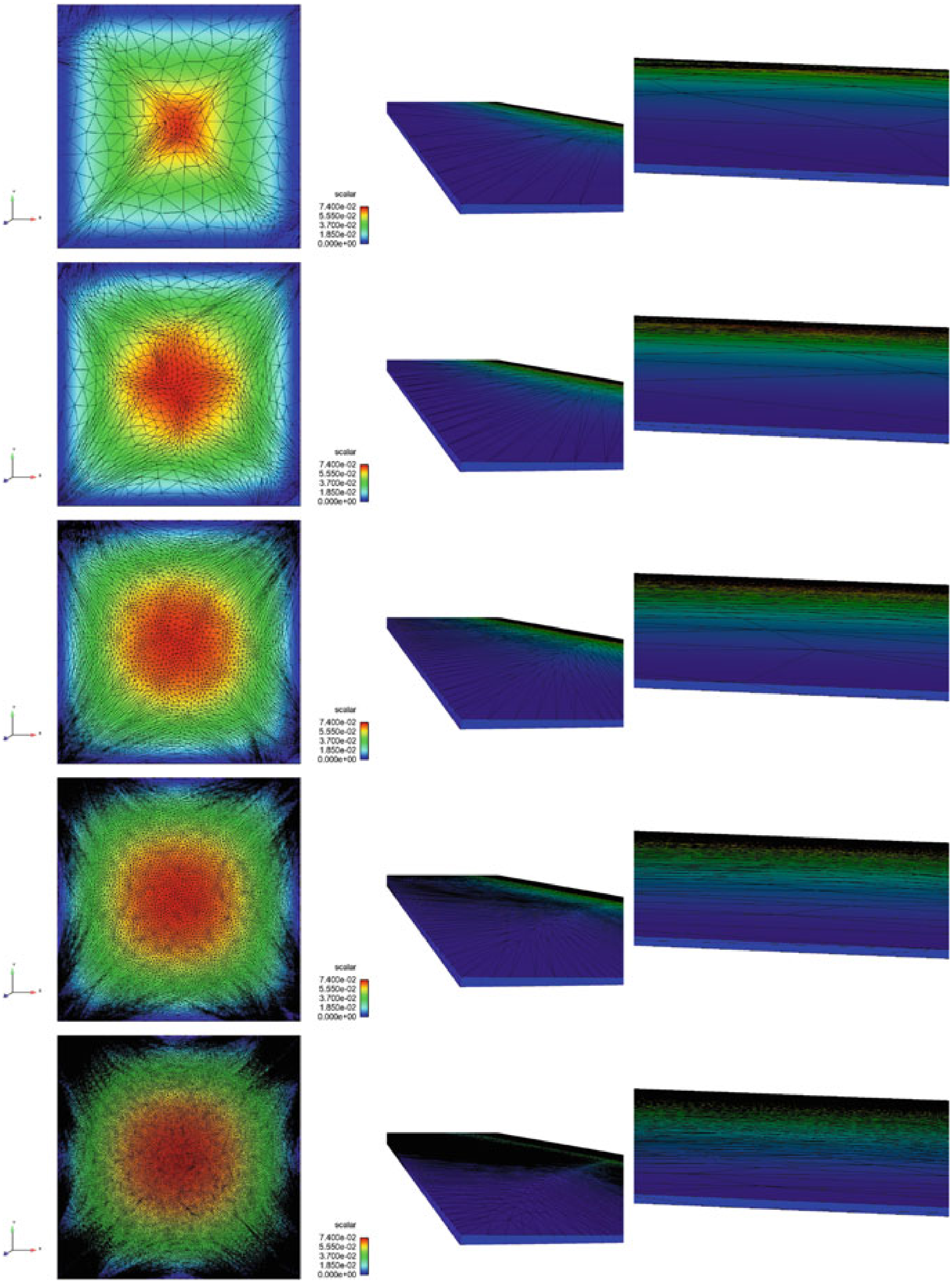

**Fig. 5** Adapted meshes with plate thickness $\varepsilon = 0.001$ and several values of *TOL* (*TOL* $= 1$ on row 1, *TOL* $= 0.5$ on row2,..., *TOL* $= 0.0625$ on row 5). *Left column*: xy plane. *Middle column*: zoom of the (0,0,0) corner. *Right column*: zoom at the center of an edge

**Table 2** Results with adapted meshes and plate thickness $\varepsilon = 0.01$ and several values of *TOL*

| Mesh nb. | *TOL* | nb. vert. | *CG* | $u_{max}$ | $\max \lambda_1$ | $\min \lambda_3$ |
|---|---|---|---|---|---|---|
| 0 | | 363 | 103 | 0.0731 | 0.16 | 0.0050 |
| 29 | 1 | 490 | 337 | 0.0697 | 0.17 | 0.0042 |
| 59 | 0.5 | 768 | 384 | 0.0713 | 0.19 | 0.0032 |
| 89 | 0.25 | 1419 | 401 | 0.0734 | 0.18 | 0.0030 |
| 119 | 0.125 | 3045 | 416 | 0.0736 | 0.15 | 0.0014 |
| 149 | 0.0625 | 7610 | 455 | 0.0736 | 0.11 | 0.00062 |

Here $\max \lambda_1 = \max_{K \in \mathscr{T}_h} \lambda_{1,K}$, $\min \lambda_3 = \min_{K \in \mathscr{T}_h} \lambda_{3,K}$

**Table 3** Results with adapted meshes and plate thickness $\varepsilon = 0.001$ and several values of *TOL*

| Mesh nb. | *TOL* | nb. vert. | *CG* | $u_{max}$ | $\max \lambda_1$ | $\min \lambda_3$ |
|---|---|---|---|---|---|---|
| 0 | | 363 | 117 | 0.0731 | 0.16 | 0.00050 |
| 29 | 1 | 1, 274 | 1, 656 | 0.0717 | 0.18 | 0.000049 |
| 59 | 0.5 | 2, 790 | 2, 680 | 0.0745 | 0.18 | 0.000061 |
| 89 | 0.25 | 7, 069 | 3, 375 | 0.0732 | 0.18 | 0.000072 |
| 119 | 0.125 | 17, 402 | 4, 090 | 0.0736 | 0.17 | 0.000054 |
| 149 | 0.0625 | 49, 718 | 4, 008 | 0.0737 | 0.16 | 0.000033 |

## References

1. Cao, W.: Superconvergence analysis of the linear finite element method and a gradient recovery postprocessing on anisotropic meshes. Math. Comp. **84**(291), 89–117 (2015)
2. Formaggia, L., Perotto, S.: New anisotropic a priori error estimates. Numer. Math. **89**(4), 641–667 (2001)
3. Formaggia, L., Perotto, S.: Anisotropic error estimates for elliptic problems. Numer. Math. **94**(1), 67–92 (2003)
4. Frey, P., George, P.L.: Mesh generation. Application to finite elements, 2nd edn. ISTE Ltd and John Wiley & Sons, New York (2008)
5. Hachem, E., Kloczko, T., Digonnet, H., Coupez, T.: Stabilized finite element solution to handle complex heat and fluid flows in industrial furnaces using the immersed volume method. Int. J. Numer. Methods Fluids **68**(1), 99–121 (2012)
6. Hassan, W.: Algorithmes d'adaptation de maillages anisotropes et application Ãă l'aérodynamique. Ph.D. thesis, EPFL, 1015 Lausanne, Switzerland (2012). Nb. 5304
7. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
8. Kunert, G.: A posteriori $L_2$ error estimation on anisotropic tetrahedral finite element meshes. IMA J. Numer. Anal. **21**(2), 503–523 (2001)
9. Loseille, A.: Adaptation de maillage anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluides. Ph.D. thesis, University Paris VI (2008)
10. Loseille, A., Alauzet, F.: Continuous mesh framework part i: Well-posed continuous interpolation error. SIAM J. Numer. Anal. **49**(1), 38–60 (2011)
11. Loseille, A., Dervieux, A., Alauzet, F.: Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations. J. Comput. Phys. **229**(8), 2866–2897 (2010)
12. Loseille, A., Löhner, R.: Adaptive anisotropic simulations in aerodynamics. AIAA-10-0169 (2010)

13. Loseille, A., Löhner, R.: Anisotropic mesh generation for high-fidelity simulations in cfd. INRIA Preprint (2014)
14. Picasso, M.: An anisotropic error indicator based on Zienkiewicz-Zhu error estimator : application to elliptic and parabolic problems. SIAM J. Sci. Comp. **24**, 1328–1355 (2003)
15. Picasso, M.: Adaptive finite elements with large aspect ratio based on an anisotropic error estimator involving first order derivatives. Comput. Methods Appl. Mech. Eng. **196**(1-3), 14–23 (2006)
16. Piggott, M.D., Farrell, P.E., Wilson, C.R., Gorman, G.J., Pain, C.C.: Anisotropic mesh adaptivity for multi-scale ocean modelling. Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. **367**(1907), 4591–4611 (2009)