European Journal of Computational Mechanics
Vol. 21, Nos. 1–2, February–April 2012, 72–91

Taylor & Francis
Taylor & Francis Group

High order optimal anisotropic mesh adaptation using hierarchical elements

R. Bois, M. Fortin, A. Fortin* and A. Couët

GIREF, Département de mathématiques et de statistique, Université Laval, Pavillon Vachon, 1045 avenue de la médecine, Québec, Canada, G1V 0A6

Anisotropic mesh adaptation has made spectacular progress in the past few years. The introduction of the notion of a metric, directly linked to the interpolation error, has allowed to control the elongation of elements as well as the discretisation error. This approach is however essentially restricted to linear ($P^{(1)}$) finite element solutions, though there exists some generalisations. A completely general approach leading to optimal meshes and this, for finite element solution of any degree, is still missing. This is precisely the goal of this work where we show how to estimate the error on a finite element solution of degree k using hierarchical basis for Lagrange finite element polynomials. We then show how to use this information to produce optimal anisotropic meshes in a sense that will be precised.

L'adaptation de maillages anisotropes a fait des progrès spectaculaires, notamment grâce à l'introduction de la notion de métrique liée à l'erreur d'interpolation. Bien qu'il existe des généralisations, cette approche est toutefois essentiellement réservée à des solutions de type Lagrange linéaire ($P^{(1)}$). Une approche générale menant à des maillages optimaux et ce, quel que soit le degré d'interpolation utilisé, est toujours manquante. C'est précisément le but de ce travail où nous montrons comment estimer l'erreur sur une solution numérique de degré quelconque à l'aide de bases hiérarchiques d'éléments finis de type Lagrange. Nous montrons ensuite comment utiliser cette information pour produire des maillages anisotropes optimaux, dans un sens qui sera précisé.

Keywords: hierarchical elements; optimal mesh; high order solutions; anisotropy; hierarchical error estimator; gradient recovery

Mots-clés: éléments hiérarchiques; maillages optimaux; maillages anisotropes; estimation; d'erreurs; estimation des dérivées

1. Introduction

One of the most important decisions that needs to be made when using the finite element method is the choice of a polynomial space for the discretisation of the different variables of the problem. Lagrange polynomials of degree less than 3 or 4 are the most commonly used, though Hermite polynomials are sometimes useful. Most of the time, these polynomials are written using the usual Lagrange basis functions $\phi_i(\boldsymbol{x})$ satisfying $\phi_i(\boldsymbol{x}_j) = \delta_{ij}$ where x_j are the element nodes and δ is the Kronecker symbol. There exists, however, other possible basis for Lagrange polynomials that do not satisfy this condition. Hierarchical basis functions, as

*Corresponding author. Email: afortin@giref.ulaval.ca

ISSN 1779-7179 print/ISSN 1958-5829 online

http://dx.doi.org/10.1080/17797179.2012.702431
http://www.tandfonline.com

described in Zaki (1993) and Ndikumagenge (2001), are one example. The idea is simple. Starting from a finite element of degree k, the discretisation space is enriched by adding new basis functions of degree $k+1$ while keeping basis functions of degree k at existing nodes. This allows to write any polynomial of degree $k+1$ as a sum

$$u_h^{(k+1)} = u_h^{(k)} + c_h^{(k+1)} \tag{1}$$

where $u_h^{(k)}$ is a polynomial of degree k and $c_h^{(k+1)}$ is a correction of degree $k+1$. This corresponds to a decomposition of the space of piecewise continuous polynomials of degree $k+1$ of the form $\mathcal{M}_h^{k+1} = \mathcal{M}_h^k \oplus \hat{\mathcal{M}}_h^{k+1}$. The hierarchical basis functions up to degree 3 in the two-dimensional case are the following:

Degree 1	Degree 2	Degree 3
λ_1	$\lambda_1\lambda_2$	$\lambda_1\lambda_2(\lambda_1-\lambda_2)$
λ_2	$\lambda_2\lambda_3$	$\lambda_2\lambda_3(\lambda_2-\lambda_3)$
λ_3	$\lambda_1\lambda_3$	$\lambda_1\lambda_3(\lambda_1-\lambda_3)$
		$\lambda_1\lambda_2\lambda_3$

where the λ_i's are the barycentric coordinates. The first three basis functions are associated to triangle vertices and the corresponding degrees of freedom are nodal values of the solution. The three quadratic and the first three cubic basis functions are attached to mid-side nodes but their degrees of freedom are now related to tangential derivatives of order 2 and 3 along the edges. The last cubic basis function is associated to the barycentre of the element.

Using a hierarchical basis presents a number of numerical advantages. For instance, very efficient iterative methods based on the decomposition (1) can be developed as described in El maliki (2007). This is specially true in the quadratic case. Indeed, using a hierarchical basis, any quadratic finite element field $u_h^{(2)}$ can be decomposed into a linear part $u_h^{(1)}$ and a quadratic correction $c_h^{(2)}$. Using this decomposition, any linear system $Au_h^{(2)} = b$ can be written in the form:

$$\begin{bmatrix} A_{ll} & A_{lq} \\ A_{ql} & A_{qq} \end{bmatrix} \begin{bmatrix} u_h^{(1)} \\ c_h^{(2)} \end{bmatrix} = \begin{bmatrix} b_l \\ b_q \end{bmatrix}$$

The matrix A_{ll} is exactly the same as one would obtain using a linear discretisation and its size is much smaller than the global matrix (about four times smaller in 2D and seven times in 3D). Moreover, it is shown in Verfürth (1996) that for elliptic problems, the condition number of the matrix A_{qq} is $O(1)$. This means that its condition number does not vary with the size of the matrix.

This structure is exploited in El maliki, Guénette, and Fortin (2011) to define a preconditioner for the Generalised Conjugate Residual (GCR) algorithm. It is shown that for large systems, the method is extremely time efficient for many different PDE's. As we will see, a second-order accurate solution can be obtained at a very small extra cost with respect to a first-order linear approximation.

Hierarchical basis functions have other interesting properties. In this work, we show how these can be exploited to build a general anisotropic mesh adaptation procedure that can be used for Lagrange polynomials of any degree in two or three dimensions. It is not our objective to make a complete review of the literature on anisotropic mesh adaptation and we

simply refer the interested reader to Hecht and Mohammadi (1997), Alauzet (2008), Lagüe (2006), Habashi et al. (2000) for metric-based methods and Babuska and Rheinboldt (1978), Formaggia and Perotto (2003), Huang, Kamenski, and Lang (2010), Kunert (2000), Micheletti and Perotto (2006), Picasso (2002), Verfürth (1996)) for residual-based or other methods.

We propose in this work a completely general approach based on a hierarchical estimation of the error. A similar idea was first introduced in Bank and Smith (1993) and more recently in Huang et al. (2010). Let $u_h^{(k)} \in \mathcal{M}_h^k$ be the solution of the following variational problem,

$$a(u_h^{(k)}, v_h) = f(v_h), \forall v_h \in \mathcal{M}_h^k, \tag{2}$$

where $a(\cdot,\cdot)$ is a bilinear form and f is a linear functional. To construct an error estimator to this problem, one can consider the following auxiliary problem

$$a(c_h^{(k+1)}, v_h) = f(v_h) - a(u_h^{(k)}, v_h), \quad \forall v_h \in \hat{\mathcal{M}}_h^{k+1}, \tag{3}$$

This is a global problem on the space $\hat{\mathcal{M}}_h^{k+1}$ involving matrix A_{qq} and is therefore well conditioned. Its solution, $c_h^{(k+1)}$, is simply an approximation to the residual of (2) and can be used in a mesh adaptation routine. Note that $u_h^{(k)} + c_h^{(k+1)}$ is an approximation of degree $k+1$ of the solution u. It is however different from $u_h^{(k+1)}$ which would be obtained by solving the complete variational problem in $\mathcal{M}_h^{k+1}$ and we therefore write $\hat{u}_h^{(k+1)} = u_h^{(k)} + c_h^{(k+1)}$. This of course leads to the construction of error estimators for any order finite elements but, once again, requires the knowledge of the problem at hand. Computations of the error by local problems as in Verfürth (1994) can be seen as local variants of the idea.

We propose in this paper a different construction for $c_h^{(k+1)}$ that will lead to a new approximation of u (still denoted $\hat{u}_h^{(k+1)}$) that does not require the solution of a global problem as in (3). We do not even need to know the variational formulation. Starting from a finite element solution $u_h^{(k)}$, the idea is to construct a more accurate solution of degree $k+1$ of the form (1). If a hierarchical basis is used, only $c_h^{(k+1)}$ needs to be computed. It will be shown that $c_h^{(k+1)}$ can be easily obtained if an appropriate approximation of degree k of the gradient of $u_h^{(k)}$ is available. We then show that its norm $\|c_h^{(k+1)}\|$ is an error estimator capable of driving a very efficient anisotropic adaptive remeshing method. We also show that the resulting meshes are optimal in the sense introduced in D'Azevedo and Simpson (1991). The case $k=1$ was described in Bois, Fortin, and Fortin (2012) and we propose here a generalisation of the approach.

The outline of the paper is as follows. In Section 2, we recall some general results on optimal meshes. In Section 3, we outline the construction of a general error estimator using hierarchical elements. Section 4 is devoted to the description of the local operations on the mesh in order to obtain a prescribed level of error. Then, in Section 5, numerical results are shown for 2D and 3D test cases. Finally, we conclude and give further comments in Section 6.

2. Optimal triangles

The shape of optimal triangles is now well known in the linear case ($k=1$) since the theoretical work of D'Azevedo and Simpson (1991). To our knowledge, a similar analysis does not exist for $k>1$ and we therefore adopt a numerical point of view. We thus consider a function

$u \in P^{(k+1)}(\Omega)$, the space of polynomials of degree $k+1$ on Ω. For a given triangle K with fixed area A_K, we would like to find the positions of its three vertices minimising the energy norm of the interpolation error $|\Pi^k(u) - u|_{1,K}$. Here, $\Pi^k(u)$ is the reinterpolation of u in the space $P^{(k)}(K)$ of polynomials of degree k over the triangle K. The resulting triangle will be called as optimal. To state this problem more clearly, we can assume, without loss of generality, that the barycentre of the triangle is at the origin. The problem is now written as

$$\min_{\{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3\}} \int_{K(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3)} |\nabla(\Pi^k(u) - u)|^2 dx \tag{4}$$

$$\text{subject to : } \text{Area}(K(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3)) = A_K \tag{5}$$

$$\text{and} \qquad \boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{x}_3 = 0 \tag{6}$$

where $\boldsymbol{x}_i = (x_i, y_i)$, $i = 1, 2, 3$, are the three vertices of the triangle K and the second constraint fixes the barycentre of K at the origin. For higher order elements, it is plausible that the optimal triangle has curved sides. It is clearly the case close to curved boundaries. We did not explore this possibility in this work and curved elements are employed uniquely on the boundary when necessary.

The constrained minimisation problem (4)–(6) can be solved using symbolic mathematical sofware like Maple™ or Mathematica® by first writing the objective function in terms of (x_i, y_i) and then substituting the constraints in it. We end up with an unconstrained minimisation problem for a function of three variables instead of six. Suppose that $v_1 = x_1$, $v_2 = y_1$ and $v_3 = y_2$ are the three remaining unknowns of this problem, then the solution satisfies the following system of non-linear equations

$$\frac{\partial}{\partial v_i}\left(\int_{K(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3)} |\nabla(\Pi^k(u) - u)|^2 dx\right) = 0, \quad i = 1, 2, 3$$

The solution of this system is in general not unique and the functional has saddle points which were obtained using Maple™. A similar analysis can be done in the three-dimensional case where we end up with a minimisation problem with eight unknowns.

The case $k = 1$ was already discussed in Bois et al. (2012) where, starting from an analytical solution $u \in P^{(2)}(\Omega)$ and a piecewise linear reinterpolation, we found the optimal triangles corresponding to the different quadratic forms (positive definite, semidefinite and indefinite) in accordance with the theoretical conclusions of D'Azevedo and Simpson (1991).

We now take a look at the case $k = 2$ where there is no theoretical results to our knowledge. Starting from an analytical solution $u \in P^{(3)}(\Omega)$ and its quadratic reinterpolation, we search for the optimal form of the triangles. The complete classification of all cubic polynomials was undertaken by Newton and it is now known that there are 78 different cases compared to three in the quadratic case (see Weisstein (2010) for instance). It is therefore not possible to present all cases and we restrict ourselves to the functions $z = (x - y)xy$ and $z = (10x - y)(10x^2 + y^2)$, the other cases following the same lines.

Depending on the choice of the cubic function, the optimal triangles take different orientations and shapes, going from equilateral, isosceles to rectangle and some with large aspect

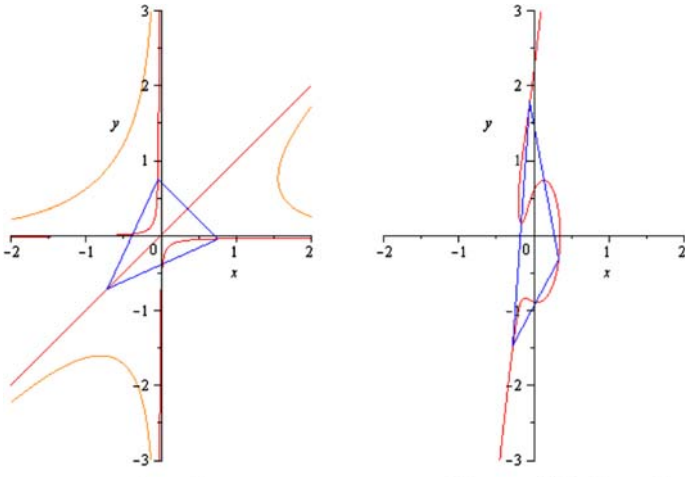


Figure 1. Optimal triangles for different cubic functions.

ratios. Two examples are presented in Figure 1. Most importantly, minimising the H^1-semi-norm of the error is clearly enough to produce anisotropic triangles as can be seen in the case $z = (10x - y)(10x^2 + y^2)$.

In practise, the interpolation error is not known and cannot be minimised. It is however possible to build an error estimator as will be described in the next section. This error estimator is then used to get optimal triangles and to reach a prescribed level of error or a given number of triangles.

3. Error estimator

Let u be an analytical solution, usually unknown, of a given partial differential equation. While the following procedure does not particularly depend on the PDE, it will, however, depend on the quality of the reconstructed gradients. Even though superconvergence of this reconstruction has been proven only for elliptic problems and special meshes, our experience shows that it works quite well in practice for a large variety of PDE's.

We denote $\mathcal{T}_h$ a triangulation of the domain Ω and K its elements and let V_h the space of continuous functions whose restriction to any element K of $\mathcal{T}_h$ belongs to the space $P^{(k)}(K)$ of polynomials of degree k over the triangle K. We therefore suppose that $u_h^{(k)} \in V_h$ is a finite element approximation of degree k to the true solution u. Our objective now is to approximate the error $\|u - u_h^{(k)}\|$ where $\|\cdot\|$ is an appropriate norm.

Hierarchical error estimators are based on the hypothesis that a more accurate approximation $\hat{u}_h^{(k+1)}$ of degree $k+1$ of the solution can be built from $u_h^{(k)}$. The precise construction of $\hat{u}_h^{(k+1)}$ will be soon described but we now suppose its existence. If $\hat{u}_h^{(k+1)}$ can be constructed such that it is an approximation of order $O(h^{k+r})$ with $r \geq 2$, then the error can simply be approximated by the relation:

$$\|u - u_h^{(k)}\| \leq \|u - \hat{u}_h^{(k+1)}\| + \|\hat{u}_h^{(k+1)} - u_h^{(k)}\| \simeq \|\hat{u}_h^{(k+1)} - u_h^{(k)}\| \tag{7}$$

As we shall see in the next section, the construction of $\hat{u}_h^{(k+1)}$ requires not only a finite element solution $u_h^{(k)}$ but it also necessitates a continuous approximation of degree k of its gradient that will be denoted $\boldsymbol{g}_h^{(k)} \in (V_h)^d$ (d is the space dimension). This is not at all an obvious task. Since $u_h^{(k)}$ is of Lagrange type, its derivatives are discontinuous at element interfaces and some form of projection is needed. The procedure is described in detail in Appendix A. In the following, we therefore suppose that we have $u_h^{(k)}$ and $\boldsymbol{g}_h^{(k)}$.

3.1. *Construction of* $\hat{u}_h^{(k+1)}$

Our main focus will be on the linear and quadratic cases since these are of utmost importance in practice. We will also restrict the presentation to two dimensions, but the generalisations to higher order approximations and to higher dimensions are immediate.

Let us first introduce some notation. The vertices of the triangle K are denoted, $\boldsymbol{x}_i$ and λ_i are the associated barycentric coordinates. The edge vector between x_i and $\boldsymbol{x}_j$ has length h_{ij} while $\boldsymbol{e}_{ij}$ is a unit vector in the same direction (the vector between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is thus $h_{ij}\mathbf{e}_{ij}$). Its mid-side node is denoted $\boldsymbol{x}_{ij}$ The median between $\boldsymbol{x}_i$ and its opposite mid-side node has length l_i and m_i is a unit vector in the same direction. The three medians meet at the barycentre $\boldsymbol{x}_B$ of the triangle. Similarly, $g_i^{(k)}$ and $g_{ij}^{(k)}$ denotes, respectively, the recovered gradients at vertices and mid-side nodes obtained by the method described in Appendix A.

The idea for the construction of $\hat{u}_h^{(k+1)}$ is then very simple. Since $\hat{u}_h^{(k+1)}$ is an approximation of u and $\boldsymbol{g}_h^{(k)} = (\boldsymbol{g}_{hx}^{(k)}, \boldsymbol{g}_{hy}^{(k)})$ is an approximation of its gradient, then the different partial derivatives of order $k+1$ of $\hat{u}_h^{(k+1)}$ should coincide with the appropriate partial derivative of order k of one of the components of $g_h^{(k)}$. Moreover, note that from (1):

$$\frac{\partial^{k+1}\hat{u}_h^{(k+1)}}{\partial x^m \partial y^n} = \frac{\partial^{k+1} c_h^{(k+1)}}{\partial x^m \partial y^n} \quad \text{with} \quad m+n = k+1$$

since $u_h^{(k)}$ is a polynomial of order k and it completely disappears from the system.

We will illustrate the construction in the linear and quadratic cases. For example, starting from a linear finite element solution, mid-side nodes can be added and the corresponding quadratic basis functions are of the form $\lambda_i\lambda_j$. On each element K, we are searching for an approximation $\hat{u}_h^{(2)}$ of the form:

$$u_h^{(1)} + c_h^{(2)} = u_h^{(1)} + c_{12,K}^{(2)}\lambda_1\lambda_2 + c_{23,K}^{(2)}\lambda_2\lambda_3 + c_{31,K}^{(2)}\lambda_3\lambda_1$$

The unknown coefficients $c_{ij,K}^{(2)}$ are obtained by comparing the second-order derivatives of $c_h^{(2)}$ (or $\hat{u}_h^{(2)}$) with the first-order derivatives of $g_h^{(1)}$. We end up with a system of three equations for three unknowns of the form:

$$\begin{cases} \frac{\partial^2 c_h^{(2)}}{\partial x^2} = \frac{\partial g_{hx}^{(1)}}{\partial x}, & \frac{\partial^2 c_h^{(2)}}{\partial y^2} = \frac{\partial g_{hy}^{(1)}}{\partial y}, \\ \frac{\partial^2 c_h^{(2)}}{\partial x \partial y} = \frac{1}{2}\left(\frac{\partial g_{hx}^{(1)}}{\partial y} + \frac{\partial g_{hy}^{(1)}}{\partial x}\right), \end{cases} \tag{8}$$

which can be easily solved for the $c^{(2)}_{ij,K}$'s:

$$\begin{aligned} c^{(2)}_{12,K} &= \tfrac{1}{8}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\cdot(\boldsymbol{g}^{(1)}_2 - \boldsymbol{g}^{(1)}_1) = \tfrac{h_{12}}{8}\boldsymbol{e}_{12}\cdot(\boldsymbol{g}^{(1)}_2 - \boldsymbol{g}^{(1)}_1) \\ c^{(2)}_{23,K} &= \tfrac{1}{8}(\boldsymbol{x}_2 - \boldsymbol{x}_3)\cdot(\boldsymbol{g}^{(1)}_3 - \boldsymbol{g}^{(1)}_2) = \tfrac{h_{23}}{8}\boldsymbol{e}_{23}\cdot(\boldsymbol{g}^{(1)}_3 - \boldsymbol{g}^{(1)}_2) \\ c^{(2)}_{31,K} &= \tfrac{1}{8}(\boldsymbol{x}_3 - \boldsymbol{x}_1)\cdot(\boldsymbol{g}^{(1)}_1 - \boldsymbol{g}^{(1)}_3) = \tfrac{h_{31}}{8}\boldsymbol{e}_{31}\cdot(\boldsymbol{g}^{(1)}_1 - \boldsymbol{g}^{(1)}_3) \end{aligned} \tag{9}$$

Note that $c^{(2)}_{ij,K}$ depends solely on the coordinates x_i and x_j and the recovered gradients $g^{(1)}_i$ and $g^{(1)}_j$ at adjacent nodes. This guarantees the continuity of $\hat{u}^{(2)}_h$ at the boundary between two elements.

Remark: The coefficients $c^{(2)}_{ij,K}$ have a nice interpretation that makes a link with metric-based adaptation. For example, we have:

$$c^{(2)}_{12,K} = \frac{h_{12}}{8}\boldsymbol{e}_{12}\cdot(\boldsymbol{g}^{(1)}_2 - \boldsymbol{g}^{(1)}_1) = \frac{(h_{12})^2}{8}\boldsymbol{e}_{12}\cdot\frac{(\boldsymbol{g}^{(1)}_2 - \boldsymbol{g}^{(1)}_1)}{h_{12}} \simeq \frac{(h_{12})^2}{8}\boldsymbol{e}_{12}\cdot H(x_M)\cdot\boldsymbol{e}_{12}$$

where $H(\boldsymbol{x}_M)$ is the Hessian matrix of u evaluated at mid-side x_M. If the Hessian is positive definite, the coefficients can be interpreted as edge lengths using the metric induced by H. Note, however, that we do not need the positive definiteness of the Hessian matrix in our method.

Generalising the notion of a metric to higher order elements is not a trivial task (see Pagnutti and Ollivier-Gooch (2009) for instance). Another possible avenue consists of recovering third-order derivatives and to extract three different metrics corresponding to each component of the gradient of the FEM solution. These metrics can then be intersected into one usable metric to drive anisotropic mesh adaptation as proposed in Alauzet and Frey (2003).

We will show that, using our approach, the generalisation to higher order elements is almost trivial. In the quadratic case, we start with a quadratic FEM solution and one needs four supplementary degrees of freedom to build a cubic approximation. To achieve this, a second degree of freedom is attached to mid-side nodes with a corresponding cubic "wave" basis function. A fourth degree of freedom is added to the barycentre of the element with the basis function $\lambda_1\lambda_2\lambda_3$ ending up with

$$\begin{aligned} \hat{u}^{(3)}_h = u^{(2)}_h + c^{(3)}_h = u^{(2)}_h &+ c^{(3)}_{12,K}\lambda_1\lambda_2(\lambda_1 - \lambda_2) + c^{(3)}_{23,K}\lambda_2\lambda_3(\lambda_2 - \lambda_3) + c^{(3)}_{31,K}\lambda_3\lambda_1(\lambda_3 - \lambda_1) \\ &+ c^{(3)}_{4,K}\lambda_1\lambda_2\lambda_3 \end{aligned}$$

Now comparing the third-order derivatives of $c^{(3)}_h$ (or $\hat{u}^{(3)}_h$) with the second-order derivatives of $g^{(2)}_h$, we obtain a new system of four equations:

$$\begin{cases} \frac{\partial^3 c^{(3)}_h}{\partial x^3} = \frac{\partial^2 g^{(2)}_{hx}}{\partial x^2}, \quad \frac{\partial^3 c^{(3)}_h}{\partial y^3} = \frac{\partial^2 g^{(2)}_{hy}}{\partial y^2}, \\ \frac{\partial^3 c^{(3)}_h}{\partial x^2\partial y} = \frac{1}{3}\left(\frac{\partial^2 g^{(2)}_{hx}}{\partial x\partial y} + \frac{\partial^2 g^{(2)}_{hx}}{\partial y\partial x} + \frac{\partial^2 g^{(2)}_{hy}}{\partial x^2}\right), \\ \frac{\partial^3 c^{(3)}_h}{\partial x\partial y^2} = \frac{1}{3}\left(\frac{\partial^2 g^{(2)}_{hy}}{\partial x\partial y} + \frac{\partial^2 g^{(2)}_{hy}}{\partial y\partial x} + \frac{\partial^2 g^{(2)}_{hx}}{\partial y^2}\right). \end{cases} \tag{10}$$

Solving this system yields:

$$\begin{aligned} c^{(3)}_{12,K} &= \tfrac{h_{12}}{3}\boldsymbol{e}_{12}\cdot(\boldsymbol{g}^{(2)}_1 - 2\boldsymbol{g}^{(2)}_{12} + \boldsymbol{g}^{(2)}_2) \\ c^{(3)}_{23,K} &= \tfrac{h_{23}}{3}\boldsymbol{e}_{23}\cdot(\boldsymbol{g}^{(2)}_2 - 2\boldsymbol{g}^{(2)}_{23} + \boldsymbol{g}^{(2)}_3) \\ \boldsymbol{c}^{(3)}_{31,K} &= \tfrac{h_{31}}{3}\boldsymbol{e}_{31}\cdot(\boldsymbol{g}^{(2)}_3 - 2\boldsymbol{g}^{(2)}_{31} + \boldsymbol{g}^{(2)}_1) \end{aligned} \tag{11}$$

and

$$c^{(3)}_{4,K} = \frac{2}{3}\left(l_1\boldsymbol{m}_1\cdot(\boldsymbol{g}^{(2)}_1 - 3\boldsymbol{g}^{(2)}_B + 2\boldsymbol{g}^{(2)}_{23}) + l_2\boldsymbol{m}_2\cdot(\boldsymbol{g}^{(2)}_2 - 3\boldsymbol{g}^{(2)}_B + 2\boldsymbol{g}^{(2)}_{31}) + l_3\boldsymbol{m}_3\cdot(\boldsymbol{g}^{(2)}_3 - 3\boldsymbol{g}^{(2)}_B + 2\boldsymbol{g}^{(2)}_{12})\right) \tag{12}$$

where $\boldsymbol{g}^{(2)}_B$ is the value of $\boldsymbol{g}^{(2)}_h$ at the barycentre of the element.

Remark: Clearly, these coefficients are related to second-order directional derivatives of $g^{(2)}_h$ along the edges for the first three coefficients and along the vectors m_i for $c^{(3)}_{4,K}$. They are thus linked to third-order derivatives of $u^{(2)}_h$ as expected. Note that $g^{(2)}_B$ does not need to be computed since it vanishes in the expression for $c^{(3)}_{4,K}$. It was left there only to emphasise that $c^{(3)}_{4,K}$ is a weighted average of second-order derivatives of $g^{(2)}_h$.

In the general case, it is a linear system of dimension $\dim(P^{(k+1)}) - \dim(P^{(k)})$ which is $k+2$ equations in two dimensions and $(k+3)(k+2)/2$ in three dimensions that is needed to determine $c^{(k+1)}_h$ on each element.

The main steps of our adaptive strategy are the following:

- Starting from a finite element solution $u^{(k)}_h$ of a given PDE.
- The gradient $g^{(k)}_h$ is recovered at the same nodes as $u^{(k)}_h$ (see Appendix A).
- Coefficients are given by (9) (or (11)) to obtain $c^{(k+1)}_h$.

Once the reconstruction of $\hat{u}^{(k+1)}_h$ is complete, then relation (7) simply becomes

$$\|u - u^{(k)}_h\| \simeq \|c^{(k+1)}_h\|.$$

3.2. *The one-dimensional case*

The one-dimensional case is worth a few comments. The above development is much easier since the system reduces to one equation in all cases. Let

$$\phi^{(2)}(\xi) = 1 - \xi^2 \quad \text{and} \quad \phi^{(3)}(\xi) = \xi(1-\xi^2)$$

the quadratic and cubic hierarchical basis functions defined on the reference element. Transformed on an element $K = [x_i, x_{i+1}]$ of length h_K, we get

$$\phi^{(2)}_K(x) = \frac{4}{h^2_K}(x - x_i)(x_{i+1} - x) \quad \text{and} \quad \phi^{(3)}_K(x) = \frac{8}{h^3_K}(x - x_i)(x - x_B)(x_{i+1} - x)$$

where $x_B = (x_i + x_{i+1})/2$. We thus want to construct:

$$\hat{u}_h^{(2)}(x) = u_h^{(1)}(x) + c_{1,K}^{(2)}\phi_K^{(2)}(x) \quad \text{and} \quad \hat{u}_h^{(3)}(x) = u_h^{(2)}(x) + c_{1,K}^{(3)}\phi_K^{(3)}(x)$$

The coefficients $c_{1,K}^{(2)}$ and $c_{1,K}^{(3)}$ are then easily computed. The second (third) derivative of $\hat{u}_h^{(2)}(x)$ ($\hat{u}_h^{(3)}(x)$) is compared to the first (second) derivative of $g_h^{(1)}(x)$ ($g_h^{(2)}(x)$). Simple computations give:

$$c_{1,K}^{(2)} = -\frac{h_K}{8}(g_{i+1}^{(1)} - g_i^{(1)}) \quad \text{and} \quad c_{1,K}^{(3)} = -\frac{h_K}{12}(\boldsymbol{g}_{i+1}^{(2)} - 2\boldsymbol{g}_B^{(2)} + \boldsymbol{g}_i^{(2)})$$

The error is thus:

$$\begin{aligned} u(x) - u_h^{(1)}(x) \simeq \hat{u}_h^{(2)}(x) - u_h^{(1)}(x) = c_{1,K}^{(2)}\phi_K^{(2)}(x) &= -\frac{1}{2h_K}(g_{i+1}^{(1)} - g_i^{(1)})(x - x_i)(x_{i+1} - x) \\ &= \frac{1}{2}\frac{(g_{i+1}^{(1)} - g_i^{(1)})}{h_K}(x - x_i)(x - x_{i+1}) \approx \frac{1}{2}u''(x_B)(x - x_i)(x - x_{i+1}) \end{aligned}$$

in the linear case and:

$$u(x) - u_h^{(2)}(x) \simeq \hat{u}_h^{(3)}(x) - u_h^{(2)}(x) \simeq \frac{1}{3!}u'''(x_B)(x - x_i)(x - x_B)(x - x_{i+1})$$

in the quadratic case. The error estimator on the element is nothing but an approximation of the classical Lagrange interpolation error.

4. Adaptive remeshing

In Bois et al. (2012), the mesh adaptation strategy consisted of modifying the mesh locally in order to attain a prescribed level of error. The same idea will be presented here but as we will see, it is often more convenient to specify a number of triangles or a number of vertices. This will be described later on. For the moment, we start with u an analytical solution and $u_h^{(k)}$ a finite element approximation of degree k of u. The global strategy driving our mesh adaptation is to try to reach a target level e_Ω of error in L^2-norm. More specifically, we would like to have:

$$\|u - u_h^{(k)}\|_{0,\Omega}^2 = e_\Omega^2$$

We would also like to obtain some form of equirepartition of this error in the sense that the error should be constant $|u - u_h^{(k)}| = c$ everywhere in the domain. Using the L^2-norm, this means that:

$$e_\Omega^2 = \int_\Omega |u - u_h^{(k)}|^2 dx = c^2 \text{ meas } (\Omega) \text{ and thus } c^2 = \frac{e_\Omega^2}{\text{meas}(\Omega)}$$

This also implies that on each element K (or on a patch of elements), the local error e_K should satisfy:

$$e_K^2 = \frac{e_\Omega^2 \text{ meas}(K)}{\text{meas}(\Omega)}$$

If this target value can be reached on each element, then the global error e_Ω will be attained on the domain Ω. The mesh will therefore be modified in order to achieve this goal. To obtain a new mesh, only local operations on the mesh (node displacement, edge swapping, node elimination and edge refinement) are used. Edge refinement and node elimination are used to reach the target level of error e_Ω. From Section 2, we know that optimal anisotropic meshes can be obtained by minimising the energy norm of the error $|u - u_h^{(k)}|_{1,\Omega}$. This is done using edge swapping and node displacement. These two local operations on the mesh are performed only if they decrease the energy norm of the error. In practice, the true error is obviously replaced by the estimated error described in Section 3.1. Note that for time-dependent problems, this is repeated over each time step.

Algorithm 4.1. GLOBAL MESH ADAPTATION

INPUT: Initial mesh $\mathcal{M}_0$, maximum number of iterations *max_it*, e_Ω.
OUTPUT: Final mesh and computed solution.

$i = 0$
while Adaptation process modified the mesh and $i \neq max\_it$ **do**
  STEP 1: Solve PDE to obtain a solution $u_h^{(k)}$ on mesh $\mathcal{M}_i$;
  STEP 2: Use the gradient recovery method to compute $\mathbf{g}_h^{(k)}$ on mesh $\mathcal{M}_i$;
  STEP 3: Adapt mesh $\mathcal{M}_i$ using local operations to obtain $\mathcal{M}_{i+1}$.
  → Minimise the H^1 -seminorm of the error using edge swapping and node displacement (error equidistribution and element optimality)
  → Control the L^2-norm of the error (or the number of elements) using edge refinement and node elimination
  STEP 4: $i \leftarrow i + 1$.
end while

As already mentioned, prescribing a level of error is not that easy since the appropriate level may vary from one problem to the other. From a practical point of view, it is easier to prescribe a target number N_T of elements. We must therefore translate a number of elements into a target level of error. As we now describe, this is done iteratively.

Let us suppose that we have a finite element solution $u_h^{(k)}$ of degree k. Provided the analytical solution is smooth enough (say $u \in H^2(\Omega)$), we have:

$$\|u - u_h^{(k)}\|_{0,\Omega} \simeq Ch^{k+1}$$

On a regular mesh, the number of elements N behaves as $N \sim h^{-d}$ or $h \sim N^{-1/d}$ and consequently:

$$\|u - u_h^{(k)}\|_{0,\Omega} \simeq CN^{-(k+1)/d}$$

The target level of error e_Ω is therefore linked to the target number of elements N_T by the relation $e_\Omega \simeq CN_T^{-(k+1)/d}$. At some time in the adaptation process, the current number of elements N_C is known and our error estimator will provide an approximation of the current error e_C. These two quantities also verify $e_C \simeq CN_C^{-(k+1)/d}$ and consequently, to obtain a target number of elements N_T, the target error should be:

$$e_\Omega = e_C \left(\frac{N_C}{N_T}\right)^{\frac{k+1}{d}}$$

and this is the value that is provided as a target. Note that this value is updated at each step of the adaptation process as new values of e_C and N_C appear.

It is also important to note that the general algorithm does not change when passing to higher order mesh adaptation. The same topological operators are used to modify the mesh but since the error estimator varies with k, the local decisions whether to modify the mesh or not may be different. Hence, an optimal mesh for a linear solution is generally not optimal for a quadratic one. This will be illustrated in the next section.

Finally, it is often necessary to adapt the mesh on more than one solution. For multi-variable problems, the following strategy was adopted. First, all variables are normalised to 1. Then the error on all these variables is estimated and simply summed.

5. Numerical results

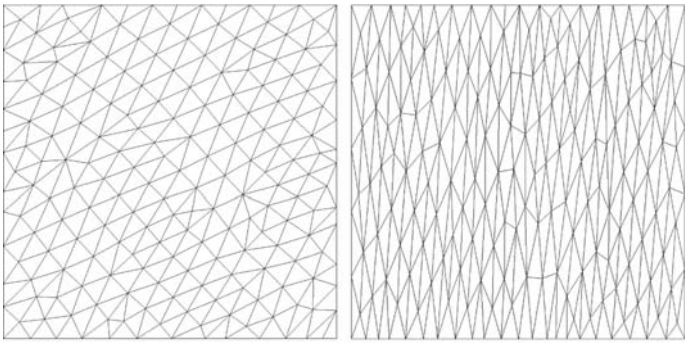
5.1. Reinterpolation problems

5.1.1. Cubic functions

We first illustrate that our adaptive strategy produces optimal triangles in the sense described in Section 2. Starting from a uniform mesh of the square $[0,1]^2$, the mesh is adapted using the exact interpolation error, that is no finite element problem is solved. Note that since the functions are cubic, the estimated error and the interpolation error are exactly the same. This is thus a pure optimal reinterpolation problem using a quadratic finite element ($k = 2$). For the two functions of Figure 1, we have obtained the meshes illustrated in Figure 2. In both cases, we observe the presence of many triangles with optimal shapes. Of course, it is not always possible to mesh the square $[0,1]^2$ with these optimal triangles but the adaptation tries to place as many optimal triangles as possible.

5.1.2. A three-dimensional example

This example is chosen to show that 3D anisotropic meshes can be obtained using our mesh adaptation algorithm. Here again, no finite element problem is solved but the interpolation error is estimated as described in Section 3.1, as opposed to the previous section where we had an exact representation of the error. The same remeshing strategy is used as in the two-



a) $xy(x-y)$ b) $z=(10x-y)(10x^2+y^2)$

Figure 2. Optimal meshes for the functions of Figure 1.

dimensional case by using edge refinement, node coarsening, edge flipping and node displacement for tetrahedral elements.

The function considered is a 3D version of the function proposed in George (2001)[1] and is defined as

$$u(x,y,z)=e^{-3\left((3x+0.3)^2+(3y+0.2)^2+(3z+0.1)^2\right)}-e^{-3\left((3x-0.3)^2+(3y-0.4)^2+(3z-0.1)^2\right)}+\frac{1}{2}\times\tanh\left(\sin(9(x^2-y^2))\cos(90(x^2+y^2))\right)$$

This function, in particular, has many steep variations and plateaus. It is very difficult to obtain a good representation of this function using a regular mesh (especially in 3D) as it needs a huge number of elements and mesh adaptation becomes essential.

As can be seen in Figure 3, the mesh is highly anisotropic where the solution allows it. In regions with plateaus, the mesh is very coarse as expected. The isovalues of the solution in the plane $z=0.5$ are also illustrated to show that the solution is well represented in the whole domain.

5.2. Two-dimensional unsteady non-linear diffusion equation

5.2.1. Position of the problem

Though our adaptive method can be applied to any partial differential equation or system of equations, we now consider an unsteady non-linear diffusion equation:

$$\begin{cases} \frac{\partial u}{\partial t}-\nabla\cdot(\boldsymbol{D}(u)\cdot\nabla u)=f & \text{in} \quad \Omega\times[0,T] \\ u=\bar{u} & \text{on} \quad \Gamma\times[0,T] \\ u(\cdot,0)=u_0 & \text{in} \quad \Omega \end{cases} \tag{13}$$

where Ω is a bounded domain of $\boldsymbol{R}^d$ $(d=2,3)$, $\boldsymbol{D}(u)=(D_{ij})$, $1\le i,j\le d$, is the diffusion tensor, $\bar{u}$ and u_0 are the boundary and initial conditions, respectively.

This problem will be solved by discretising in time using a fully implicit backward finite difference of order 2 (BFD2)

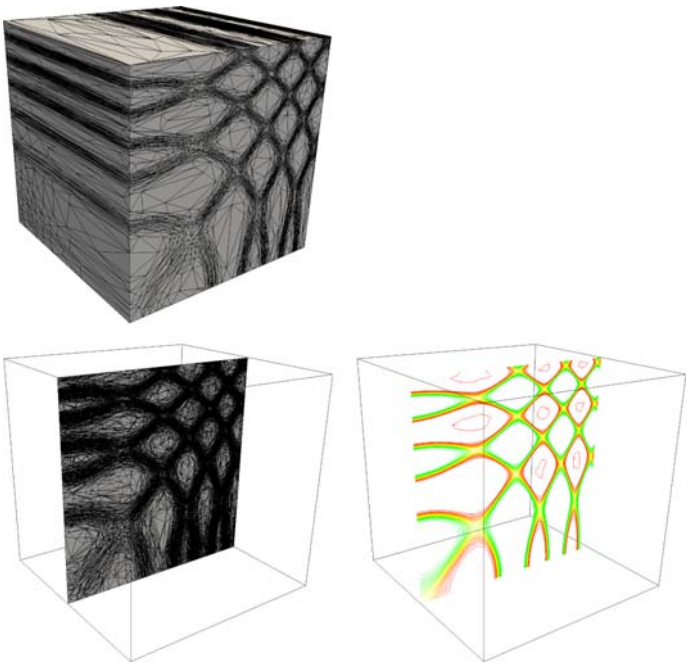


Figure 3. (Top left) adapted mesh; (top right) mesh in the plane $z = 0.5$; (bottom) isovalues in the plane $z = 0.5$.

$$\frac{\partial u}{\partial t}(t_n) = \frac{3u_n - 4u_{n-1} + u_{n-2}}{2\Delta t} + O(\Delta t)^2$$

Note that this time discretisation involves the solutions at times t_n, t_{n-1} and t_{n-2}. In an adaptive scheme, this means that solutions from previous time steps must be reinterpolated on the current mesh. The strategy is thus the following. In every time step n, we suppose that the mesh is adapted for the solutions u_{n-1} and u_{n-2}. A resolution of the problem is then performed to get a first approximation of the solution at time t_n denoted $\tilde{u}_n$. The mesh is then adapted for $\tilde{u}_n$, u_{n-1} and u_{n-2} and all these variables are reinterpolated on the new mesh. Finally, a second resolution is done to get the final u_n at time t_n before moving on to the next time step. Of course, this could be repeated many times but the computational time would evidently increase.

Newton's method is used to solve the non-linear system at each time step. The resulting linear systems is solved using a direct solver (MUMPS) described in Amestoy, Duff, and L'Excellent (2000) and is applied when using both linear and quadratic elements.

5.2.2. *Test using manufactured solutions*

For this test, we use the method of manufactured solutions (Roache, 2002). For this, let the circle of radius $\frac{3}{8}$ centred at $(\frac{1}{2}, \frac{1}{2})$ parametrised by $(x_0(t), y_0(t)) = (\frac{3}{8}\cos(t) + \frac{1}{2}, \frac{3}{8}\sin(t) + \frac{1}{2})$, $t \in [0, 2\pi]$, inside $\Omega = [0, 1]^2$ (the unit square).

Now, let u be the Gaussian function defined as

$$u(x, y, t) = e^{-1000((x-x_0(t))^2 + (y-y_0(t))^2)}$$

and the diffusion tensor as

$$D(u) = \begin{pmatrix} u+1 & 0 \\ 0 & u+1 \end{pmatrix}$$

Note that for each $t \in [0, 2\pi]$, u is maximal at (x_0, y_0) and is very sharp around this point. If we substitute this function in the left-hand side of [13], we can generate the functions f, $\bar{u}$ and u_0 such that a finite element code can be used to compute the approximation u_h. We will now use u_h to adapt the mesh using Algorithm 4 and evaluate the true error $u - u_h$ at each time step and compare against regularly refined meshes. For the adapted meshes, the target number of elements was set to approximately 9500. Of course, this actual number of elements slightly varies with time around that number.

Figure 4 shows meshes and corresponding solutions at time steps $t = \frac{\pi}{2}$ and $t = \frac{3\pi}{2}$. Again, since the mesh is adapted to the solutions at time steps t_n, t_{n-1} and t_{n-2} as mentioned before, the concentrated elements near the current solution forms more or less an ellipse instead of a disc. The actual form of the ellipse depends on the time step size.

The complete results for both linear and quadratic elements are summarised in Figure 5. From these results, we can see that mesh adaptation is very beneficial in terms of precision of the solution. When the mesh is suitably adapted (with around 9500 elements), we obtain essentially the same (and sometimes even better) level of precision using approximately 14 times less elements compared to the most refined regular mesh (131,072 elements). This is true for both linear and quadratic discretisations. Note that this factor could be even greater if we had used a sharper function u since more elements would be needed to have a good

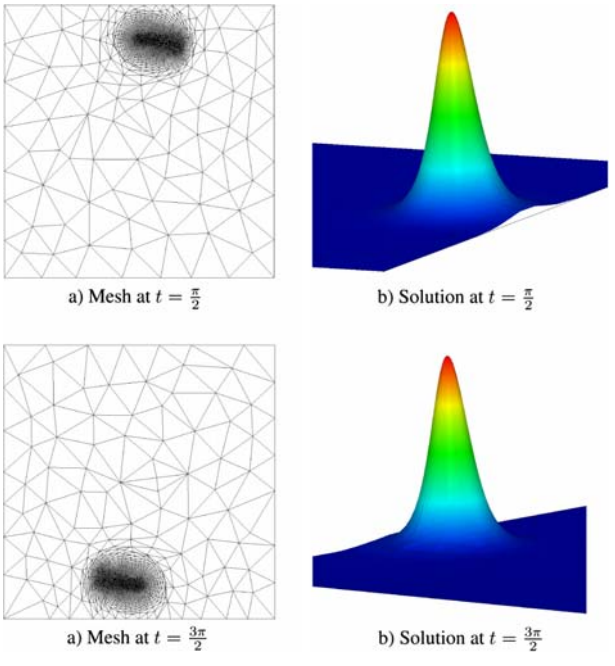


Figure 4. Meshes and solutions at two different time steps.

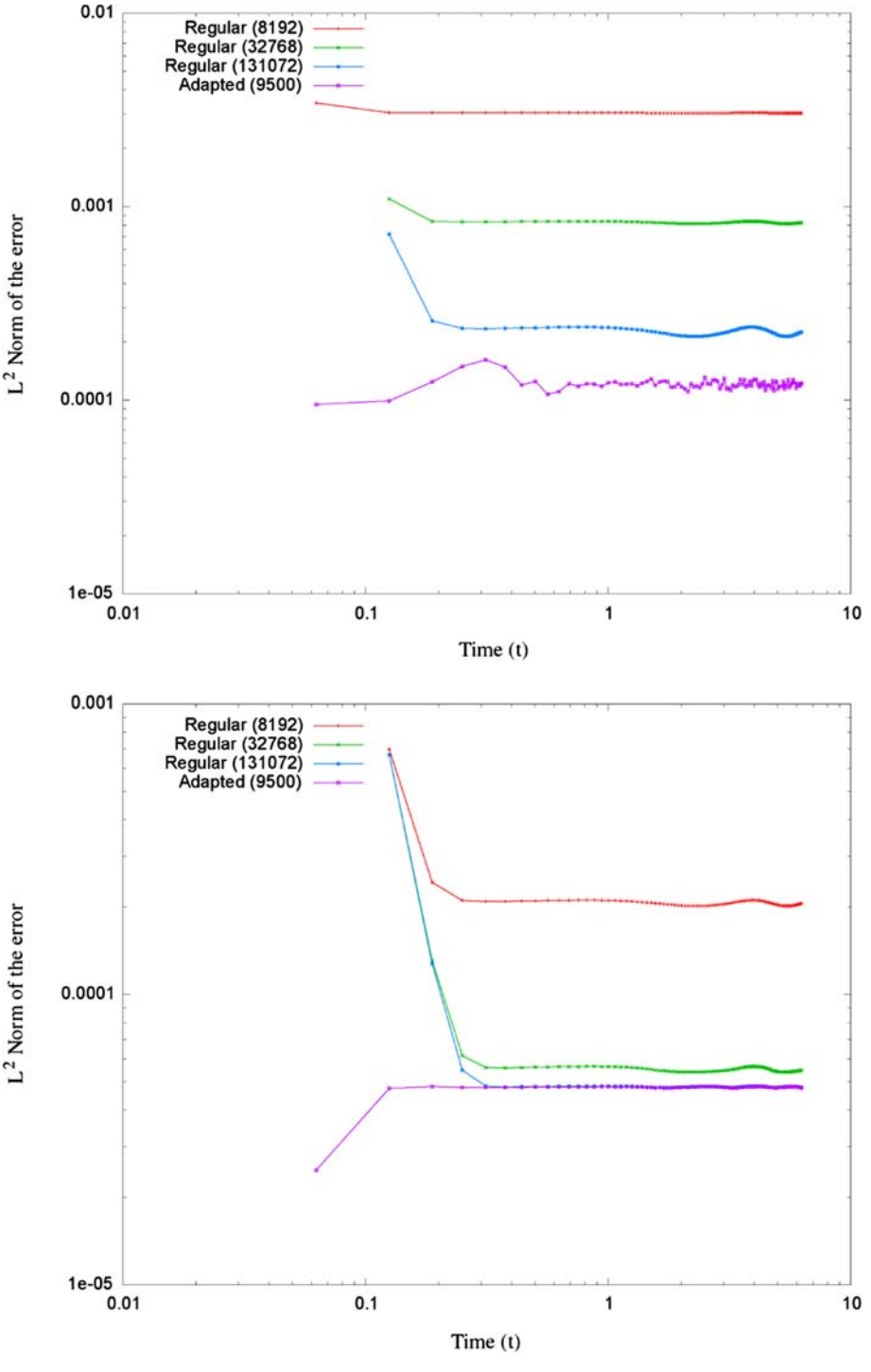


Figure 5. Log-Log plot of the error over time using $P^{(1)}$ elements (top) and $P^{(2)}$ elements (bottom).

Table 1. CPU times (in s) using 1 core of Intel(R) Core(TM)2 Quad 2.83 GHz processor.

Type	# Elements	Time P^1 (MUMPS)	Time P^2 (MUMPS)	Time P^2 (Iterative solver)
Regular	2048	119	131	137
	8192	445	533	512
	32,768	1872	2325	2049
	131,072	7470	11,811	9739
Adapted	9500	6480	6763	6763

approximation using a regular mesh (and the benefits of mesh adaptation would therefore be greater in this case).

Another important aspect is the computational cost. In order for mesh adaptation to be truly beneficial, it must also be more cost efficient compared with regularly refined meshes. Table 1 summarises CPU times for the different simulations. For the adapted cases, the simulation times include two resolutions of the non-linear system and one adaptation step in between, performed at each time step. The last column gives the CPU time using the preconditioned GCR method (El maliki et al., 2011) discussed in Section 1.

The table shows that obviously, quadratic solutions are more expensive. But when using the GCR iterative solver and the hierarchical basis, the extra cost with respect to a linear solution decreases significantly. This means that a much more accurate quadratic solution can be obtained at very little extra cost with respect to a linear solution. For the adapted meshes, no gain was observed between the direct and the iterative solver in the quadratic case. The iterative method is more sensible to anisotropic meshes and therefore convergence is slightly slower but its performance is still comparable to a direct solver. Direct solvers are no longer usable in the three-dimensional case while the iterative solver will still be very efficient and we expect that combined with mesh adaptation, the results would be highly satisfactory.

5.3. *The flow around a circular cylinder*

We now consider the flow of a Newtonian fluid around a circular cylinder at Reynolds' number 30. The flow is stationary as it becomes time dependent at a critical Reynolds' number around 45 as shown in Engelman and Jamnia (1990). A uniform velocity profile $u = (1, 0)$ is imposed at the entrance and on the top and bottom boundaries. The velocity vanishes on the cylinder and a free flow ($\sigma \cdot n = 0$) is imposed at the exit section (σ is the Cauchy stress tensor). The cylinder has diameter 1 centred at $(0, 0)$ and is placed in the rectangular box with corners $(-20, -20)$ and $(45, 20)$.

The incompressible Navier–Stokes equations are then solved in a velocity–pressure $(u - p)$ formulation using Newton's method for the linearisation. Two different discretisations were tested: the first-order mini element $(P_1 - P_1)$ (stabilised with a bubble function at the barycentre of the element for the velocity) and the second-order Taylor–Hood $(P_2 - P_1)$ element (see Brezzi & Fortin, 1991). This will allow a comparison of the resulting meshes when using linear and quadratic solutions.

The adaptation strategy is then used with the same target error for both discretisations. Figure 6 presents the resulting meshes and clearly, the number of elements for a quadratic discretisation is much lower than for the linear one for similar accuracy. The flow presents a recirculation zone behind the cylinder which is clearly captured by both meshes. The

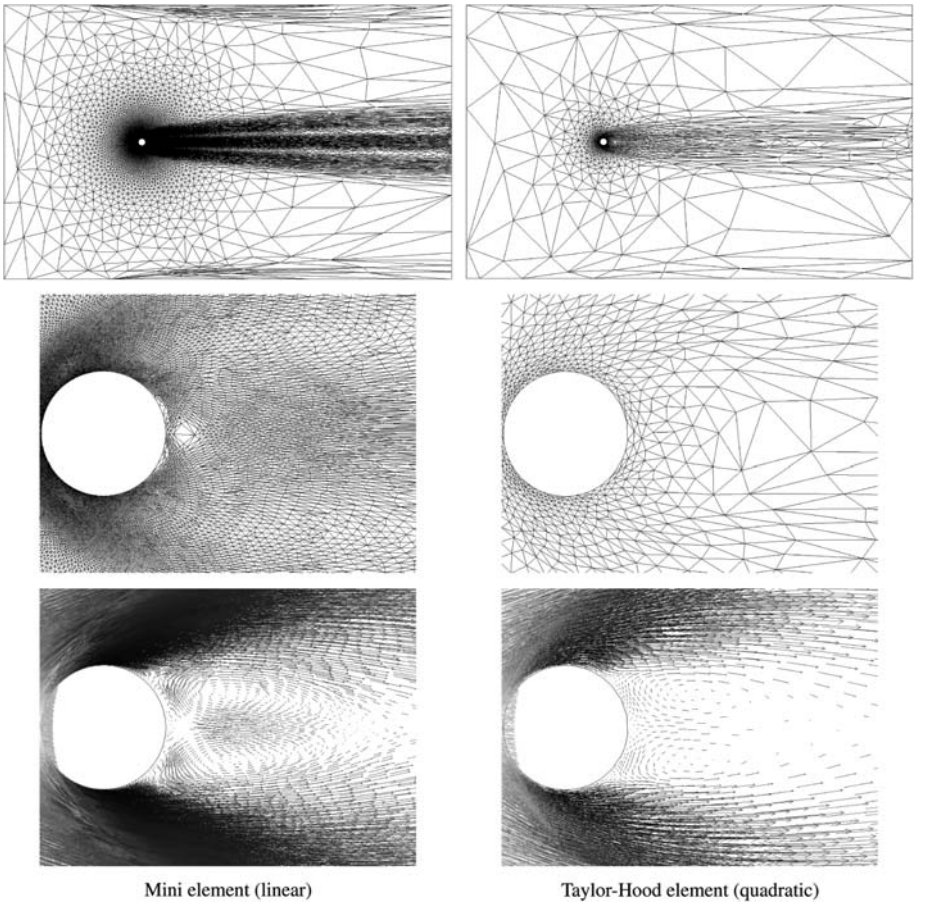


Figure 6. Meshes and velocity field: mini and Taylor–Hood elements.

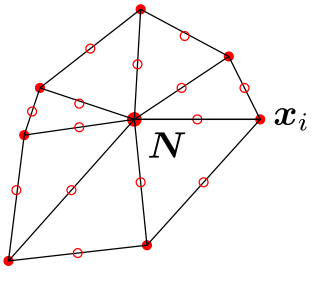


Figure 7. Neighbouring nodes to vertex N.

elements are anisotropic in the wake in both cases. The mesh is also slightly adapted on the top and bottom boundaries, showing that the boundary conditions (uniform flow) are not perfectly appropriate. These two boundaries should be placed still further from the cylinder.

6. Conclusions

A completely general error estimator valid for Lagrange polynomials of any order was presented. A very accurate error estimator was built using hierarchical basis of Lagrange finite element discretisations. It can be applied to a large variety of two- and three-dimensional, stationary and time-dependent problems. The estimator is rich enough to drive optimal anisotropic mesh adaptation.

Acknowledgements

The authors wish to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

Note

1. There is an error in the definition of this function in George (2001) here corrected.

References

Alauzet, F. (2008). High-order methods and mesh adaptation for Euler equations. *International Journal for Numerical Methods in Fluids, 56*(8), 1069–1076.

Alauzet, F., & Frey, P. (2003). *Estimateur d'erreur géométrique et métriques anisotropes pour l'adaptation de maillage. Partie I: aspects théoriques*. Technical Report n° 4759, INRIA.

Amestoy, P.R., Duff, I.S., & L'Excellent, J.-Y. (2000). Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering, 184*(2–4), 501–520.

Babuska, I., & Rheinboldt, W.C. (1978). A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering, 12*, 1597–1615.

Bank, R.E., & Smith, R.K. (1993). A posteriori error estimates based on hierarchical bases. *SIAM Journal on Numerical Analysis, 30*(4), 921–935.

Bois, R., Fortin, M., & Fortin, A. (2012). A fully optimal anisotropic mesh adaptation method based on a hierarchical error estimator. *Computer Methods in Applied Mechanics and Engineering, 209–212*, 12–27.

Brezzi, F., & Fortin, M. (1991). *Mixed and hybrid finite element methods*, vol. 15 of Springer series in computational mathematics. New York, NY: Springer-Verlag.

D'Azevedo, E.F., & Simpson, R.B. (1991). On optimal triangular meshes for minimizing the gradient error. *Numerische Mathematik, 59*(1), 321–348.

El maliki, A. (2007). *Résolution de problèmes aux limites à l'aide de méthodes itératives hiérarchiques à préconditionneur variable*. PhD thesis, Département de mathématiques et de statistique, Université Laval, Québec, Canada.

El maliki, A., Guénette, R., & Fortin, M. (2011). An efficient hierarchical preconditioner for quadratic discretisations of finite element problems. *Numerical Linear Algebra with Applications*. n/a-n/ a. Published online.

Engelman, M., & Jamnia, M. (1990). Transient flow past a circular cylinder: A benchmark solution. *International Journal for Numerical Methods in Fluids, 11*, 985–1000.

Formaggia, L., & Perotto, S. (2003). Anisotropic error estimates for elliptic problems. *Numerische Mathematik, 94*, 67–92. doi: 10.1007/s00211-002-0415-z.

George, P.-L. (Ed.). (2001). *Maillage et adaptation, Mécanique et ingénierie des matériaux*. Paris: Hermès Science Publication.

Habashi, W.G., Dompierre, J., Bourgault, Y., Ait Ali Yahia, D., Fortin, M., & Vallet, M.-G. (2000). Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solverindependent CFD. Part I: General principles. *International Journal for Numerical Methods in Fluids, 32*, 725–744.

Hecht, M., & Mohammadi, B. (1997). Mesh adaptation by metric control for multi-scale phenomena and turbulence. In *35th Aerospace Sciences Meeting & Exhibit*, n° 97-0859, Reno, USA.

Huang, W., Kamenski, L., & Lang, J. (2010). A new anisotropic mesh adaptation method based upon hierarchical a posteriori error estimates. *Journal of Computational Physics, 229*(6), 2179–2198.

Kunert, G. (2000). An a posteriori residual error estimator for the finite element method on anisotropic tetrahedral meshes. *Numerische Mathematik, 86*, 471–490.

Lagüe, J.-F. (2006). *Adaptation de maillage basée sur une erreur d'interpolation locale*. PhD thesis, Université Pierre et Marie Curie, Paris VI.

Manole, C., Vallet, M.-G., Dompierre, J., & Guibault, F. (2005). Benchmarking second-order derivatives recovery of a piecewise linear scalar field. In *Proceedings of the 17th IMACS World Congress Scientific Computation, Applied Mathematics and, Simulation*, 11–15 July 2005, Paris, France: Villeneuve d'Ascq, France, Ecole Centrale de Lille.

Micheletti, S., & Perotto, S. (2006). Reliability and efficiency of an anisotropic Zienkiewicz–Zhu error estimator. *Computer Methods in Applied Mechanics and Engineering, 195*(9–12), 799–835.

Ndikumagenge, F. (2001). *Estimateur d'erreur a posteriori basé sur une méthode hiérarchique et adaptation de maillage*. Master's thesis, Université Laval, Québec, Canada.

Pagnutti, D., & Ollivier-Gooch, C. (2009). A generalized framework for high order anisotropic mesh adaptation. *Computers & Structures, 87*, 670–679.

Picasso, M. (2002). An anisotropic error indicator based on Zienkiewicz–Zhu error estimator: Application to elliptic and parabolic problems. *SIAM Journal on Numerical Analysis, 24*, 1328–1355.

Roache, P.J. (2002). Code verification by the method of manufactured solutions. *ASME Journal of Fluids Engineering, 124*(1), 4–10.

Verfürth, R. (1994). A posteriori error estimation and adaptive mesh-refinement techniques. *Journal of Computational and Applied Mathematics, 50*(1–3), 67–83.

Verfürth, R. (1996). *A review of a posteriori error estimation and adaptive mesh-refinement techniques of Advances in numerical mathematics*. Chichester: Wiley-Teubner.

Weisstein, E.W. (2010). Cubic curve. Retrieved from MathWorld–A Wolfram Web, Resource.

Zaki, A. (1993). *Simulation numérique des problèmes de convection sur des maillages adaptatifs non structurés du type h–p*. PhD thesis, École Polytechnique de Montréal, Montréal, Canada, July.

Zhang, Z., & Naga, A. (2005). A new finite element gradient recovery method: Superconvergence property. *SIAM Journal on Numerical Analysis, 26*(4), 1192–1213.

Zienkiewicz, O.C., & Zhu, J.Z. (1992a). The superconvergent patch recovery and a posteriori error estimate, part I: The recovery technique. *International Journal for Numerical Methods in Fluids, 33*, 1331–1364.

Zienkiewicz, O.C., & Zhu, J.Z. (1992b). The superconvergent patch recovery and a posteriori error estimate, part II: Error estimates and adaptivity. *International Journal for Numerical Methods in Fluids, 33*, 1365–1382.

Appendix A. Reconstruction of derivatives at nodes

There exists many gradient recovery techniques and it is not our purpose to make a review of existing methods but simply refer readers to Zienkiewicz and Zhu (1992a, 1992b), Zhang and Naga (2005) and for a benchmarking study, see some methods (see Manole, Vallet, Dompierre, & Guibault, 2005). Our experience shows that the method proposed by Zhang and Naga (2005) is one of the most effective methods, and we now recall their method.

Starting from a finite element solution $u_h^{(k)}$ of degree k, they build, at each vertex N of the mesh, a polynomial $p^{(k+1)}$ of degree $k+1$ by solving a least squares problem of the form

$$\inf_{p \in P^{(k+1)}} \sum_i |p(x_i) - u_h^{(k)}(x_i)|^2$$

where x_i are the coordinates of all the nodes adjacent to N. These neighbouring nodes include vertices identified by large red dots in Figure 7, nodes located on edges identified by red circles and eventually nodes inside the triangles. Figure 7 illustrates the quadratic case $k = 2$.

For high values of k, the patch may have to be extended to another layer of elements in order to get a well-posed problem. The gradient is then recovered by evaluating the derivatives of this polynomial at N. This procedure is performed at triangle vertices only. At mid-side nodes, the computed poly-

nomials corresponding to the adjacent vertices are differentiated, evaluated at mid-side and averaged. Note that this is different from averaging the computed gradients at adjacent vertices. A similar procedure is done at the barycentre when necessary. Using their methodology, the gradient can be reconstructed at the same nodes as $u_h^{(k)}$ and will be denoted as $g_h^{(k)}$. It is also shown that this gradient recovery enjoys the polynomial preserving property (if $u \in P^{(k+1)}$, $g_h^{(k)} = \nabla u$).