Journal of Computational Physics 501 (2024) 112774

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Anisotropic mesh adaptation for high-order finite elements spaces with the log-simplex method. Application to discontinuous Galerkin methods

Olivier Coulaud *, Adrien Loseille, Pierre Schrooyen

ARTICLE INFO

Keywords:
Anisotropic metric based mesh adaptation
Error estimates
High-order mesh adaptation
High-order numerical schemes
Discontinuous Galerkin methods

ABSTRACT

In this article, a high-order solution-based mesh adaptation method is investigated. This later, which is called the log-simplex method, relies on the approximation of high-order differential form of the solution by a dedicated metric-field. The method is a natural extension of Hessian-based methods to the high-order case.

1. Introduction

When numerically simulating physical problem with finite elements method, having a mesh which well suits the considered problem is one of the most significant prerequisites. Indeed, in such computations, two kinds of errors occur, which are the implicit error due to the numerical scheme itself and the interpolation error due to projection of the solution onto a finite elements space. This later strongly depends on mesh quality with respect to the solution field, and its minimization is a very active field of research. In this article, we investigate one of the most common ways to obtain a suitable mesh, namely the anisotropic mesh adaptation. Through this approach, the problem is addressed by modifying an initial simplicial mesh into a new one, which minimizes the interpolation error of a given solution field. Notice that, on the contrary to the isotropic mesh adaptation which only takes into account the elements size (usually the edges' lengths), the anisotropic ones also consider the orientations of the elements as part of the optimization process. When considering classical $\mathbb{P}_1$ finite elements spaces, which are composed of piecewise linear functions over the mesh elements, the problem has been widely studied and has found a large amount of industrial applications (see for instance [2,4,17]). In this case, the interpolation error is actually governed by the Hessian matrix of the considered solution, which leads the mesh adaptation process. These methods are called Hessian-based methods, and have been tremendously fruitful for decades (see for instance [7,13,18,21,24]). However, when the physical problem is solved by high-order numerical schemes, which may be built through discontinuous Galerkin [34] or spectral differences [22] methods, Hessian-based methods are no more applicable nor easily extendable to these kinds of schemes. Indeed, though there exist publications addressing this purpose [14,15,19,30,38], they rarely give truly applicable algorithms, especially in 3 dimensional space.

The purpose of the present article lies in the extension of Hessian-based anisotropic mesh adaptation methods to the case of high-order finite elements spaces $\mathbb{P}_k$, where $k > 1$ is arbitrary. In this case, usual $\mathbb{P}_1$ error estimate can be replaced by its high-order version, lead by the $k+1$ differential of the solution. The main difficulty is then to translate this high-order differential into meshing features. Actually, one of the key points of Hessian-based methods is the fact that a metric-field can easily be deduced

* Corresponding author.
E-mail address: olivier.coulaud@cenaero.be (O. Coulaud).

https://doi.org/10.1016/j.jcp.2024.112774
Received 24 February 2023; Received in revised form 30 October 2023; Accepted 10 January 2024

Available online 15 January 2024

from the Hessian-matrix of a solution. This metric-field is then used to lead mesh adaptation, through the notion of unit mesh with respect to a metric-field (see Section 2.2). When high-order finite elements spaces are involved, the link between the high-order derivatives of the solution and a metric-field is not that straightforward to deduce. An interesting way to address this issue can be found in [8,31,32], where calculus of variations techniques allow to derive a metric-field from an alternative representation, based on aspect ratio, angle and density of the metric. Their method both applies to 2D and 3D cases, and has been mostly used for analytical high-order solutions. In the present article, the mesh adaptation method is tested for steady and unsteady numerical solutions which are computed from the Discontinuous Galerkin solver Argo. The theoretical background of it relies on Cao's work [5,6] where he established a relationship between the high-order differential of a solution and a metric-field. More precisely, this author has exhibited what kind of metric-fields would be suitable to minimize a given high-order solution, but the way to build such a metric-field remained open in general. In practical, the problem reduces to find the largest ellipse in 2D or ellipsoid in 3D included into an area delimited by the level-set of level 1 of the homogeneous polynomial of degree $k+1$ given by the $k+1$ differential of the solution (see Section 4). This derived problem has been notably addressed by Mirebeau in [29] up to $k=3$ and in the general case by Hecht and Kuate in [19]. Their method gives a first numerical implementation of the problem, but only applies in 2D and is quite costly to operate. This paper is devoted to the theoretical and numerical study of the log-simplex method, which enables to automatically derive a metric-field from a high-order solution field, both in 2D and 3D. It has been introduced in [10] and is based on two main ideas. First, instead of considering the optimization problem on a metric $\mathcal{M}$, the framework is translated in terms of the log-matrix $\mathcal{L} = \log(\mathcal{M})$. The second idea is the simplification of the non-linear optimization problem to a sequence of linear ones, which can be solved by a usual simplex algorithm. Notice that this method applies to any order and any kind of polynomial interpolations and that it is a true extension to the Hessian-based methods. Indeed, the application of the log-simplex method with $k=1$ reduces to the classical Hessian-based method described in [25,26].

This paper is organized as follows. The section 2 introduces the mesh optimization problem and gives some recalls about the metric-based adaptation framework. Then, in Section 3, a high-order continuous error estimate is exhibited, which is a natural extension to the $\mathbb{P}_1$ error model leading to the Hessian-based mesh adaptation methods. The section 4 is devoted to the description of the log-simplex algorithm, which is a way to translate the high-order error model in terms of metric field. Finally, in Section 5, the method is applied to several 2D and 3D numerical cases, and the improvement of the representation of the solution by high-order finite elements spaces induced by the mesh adaptation procedure is precisely measured.

2. Framework and notations

In this section, the mesh optimization problem is stated rigorously. It is also shown how a mesh adaptation problem can be expressed as a Riemannian metric space optimization problem, following the framework of [25,26]. Even though the considered solutions are represented with high-order polynomial functions, the mesh itself is composed of linear $\mathbb{P}_1$ simplex elements (straight triangles in 2D, straight tetrahedra in 3D).

2.1. High-order mesh adaptation problem

Throughout this paper, a fixed domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ is considered, as well as a linear simplicial mesh $\mathcal{H}$ of Ω. Let $\mathbb{P}_k(\mathcal{H})$ be the classical finite elements space given by

$$\mathbb{P}_k(\mathcal{H}) = \left\{ v : \Omega \to \mathbb{R} \mid v_{|\mathcal{T}} \text{ is a polynomial of degree } k \right\},$$

where $\mathcal{T}$ denotes a simplex element of $\mathcal{H}$.

The dimension of such a functions space is finite, and each function of $\mathbb{P}_k(\mathcal{H})$ can be determined by its values at interpolation nodes inside each element $\mathcal{T}$. Indeed, given a set of nodes $\{x_1, x_2, ..., x_n\} \in \mathcal{T}$, with $n = \frac{(k+1)(k+2)}{2}$ if $d=2$ and $n = \frac{(k+3)(k+2)(k+1)}{6}$ if $d=3$, a function $u : \Omega \to \mathbb{R}$ can be approximated by its projection $\Pi_k u$ onto $\mathbb{P}_k(\mathcal{T})$ as the unique polynomial of degree k such that

$$\Pi_k u(x_i) = u(x_i), \text{ for all } i \in \{1, ..., n\}.$$

The projection of a solution u onto $\mathbb{P}_k$ depends on the choice of nodes locations, and is exact if u is a polynomial whose degree is smaller than k. Throughout this article, the locations of nodes $x_i, i = 1, ..., n$ are assumed to be the same for each element of the mesh, and are not adaptation parameters. Notice that theoretical developments described in the present article remain valid for every kind of nodes locations.

Let $C(\mathcal{H})$ denote the mesh complexity $\mathcal{H}$, that is to say the number of tetrahedra of $\mathcal{H}$. In this paper is addressed the following problem.

Problem 1. Let $k \geq 1$ be a fixed order of interpolation. Given a smooth function $u : \Omega \to \mathbb{R}$, a fixed complexity $N > 0$ and $p \in [1, ..., +\infty]$, find the optimal simplicial mesh $\mathcal{H}$ of Ω with complexity N such that

$$\mathcal{H}_{opt} = \underset{\mathcal{H}:\, C(\mathcal{H})=N}{\operatorname{argmin}} \left\| u - \Pi_k u \right\|_{L^p}, \tag{2.1}$$

where $\|.\|_{L^p}$ is the classical Lebesgue L^p-norm on Ω.

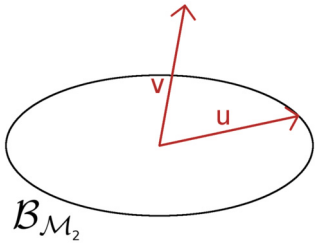


Fig. 1. The 2D unit ball $\mathcal{B}_{\mathcal{M}_2}$ of a 2D metric $\mathcal{M}_2$. The two vectors u and v have the same length for the usual Euclidean norm but different orientations. Hence, their lengths in the metric space associated to $\mathcal{M}_2$ differ: $|u|_{\mathcal{M}_2} = 1$ whereas $|v|_{\mathcal{M}_2} = 2$.

2.2. Continuous mesh framework

The mesh adaptation is performed using a metric-based adaptation process. This technique enables to produce anisotropic meshes, and is based on the use of a particular distance function, defined through a metric tensor. In this section, the theoretical background related to the metric-based mesh adaptation is recalled. It includes some basic notions about Riemannian geometry, and introduces the notions of continuous mesh and unit mesh with respect to a metric-field. In what follows, $\mathcal{M} \in \mathcal{M}_{d,d}(\mathbb{R})$ denotes a metric, that is to say a symmetric positive definite matrix of $\mathbb{R}^d$. With use of such a metric, an Euclidean metric space can be defined. It is stated in Definition 2.1 below.

Definition 2.1. Let $\mathcal{M}$ be a metric of $\mathbb{R}^d$. The Euclidean metric space $(\mathbb{R}^d, \mathcal{M})$ is the vector space associated to the dot product given by

$$\langle x, y \rangle_{\mathcal{M}} = x^t \mathcal{M} y, \text{ for } x, y \in \mathbb{R}^d. \tag{2.2}$$

Likewise, the norm $|.|_{\mathcal{M}}$ results from this scalar product and is given by

$$|x|_{\mathcal{M}} = \sqrt{x^t \mathcal{M} x}, \text{ for all } x \in \mathbb{R}^d. \tag{2.3}$$

The main principle from which anisotropic mesh adaptation is addressed comes from the use of scalar product and distance given by Definition 2.1, instead of usual Euclidean ones. In fact, all geometric quantities, such as length, area or volume can be computed through $|.|_{\mathcal{M}}$. Considering a metric-space instead of the usual Euclidean space involves anisotropy in the lengths computations. Indeed, in this framework, the unit ball $\mathcal{B}_{\mathcal{M}} = \left\{ x \in \mathbb{R}^d : |x|_{\mathcal{M}} = 1 \right\}$ is no longer a circle in 2D nor a sphere in 3D but an ellipse or an ellipsoid, and consequently the length of a segment may depend on its orientation (see Fig. 1). Definition 2.1 extends to the notion of Riemmanian metric-field, where a smooth metric field $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x} \in \Omega}$ is considered, whose values depend on $x \in \Omega$. Following this idea, the Riemannian length of a curve $\gamma : [t_0, t_1] \to \mathbb{R}^d$ is given by the integral formula

$$\ell_{\mathbf{M}}(\gamma) = \int_{t_0}^{t_1} |\gamma'(t)|_{\mathcal{M}(\gamma(t))} \, dt. \tag{2.4}$$

In the particular case where γ is a straight segment linking two points $a, b \in \mathbb{R}^d$, $\gamma(t) = (1-t)a + tb$, $t \in [0,1]$, the length of $[a, b]$ writes

$$\ell_{\mathbf{M}}(ab) = \int_0^1 \sqrt{ab^t \mathcal{M}(\gamma(t)) \, ab} \, dt. \tag{2.5}$$

In particular, if $\mathbf{M}$ is constant over Ω, it reduces to the Euclidean distance given by Definition 2.1. The link between a discrete mesh and a continuous metric field on Ω is established by the notion of unit element with respect to a metric, stated by Definition 2.2 below.

Definition 2.2. Let $\mathcal{M}$ be a metric of $\mathbb{R}^d$. An element is said to be unit with respect to $\mathcal{M}$ if its edges $(e_i)_{i \in \{1,\dots,d+1\}}$ satisfy

$$|e_i|_{\mathcal{M}} = 1, \text{ for all } i \in \{1, \dots, d+1\}.$$

Actually, a unit element with respect to a metric $\mathcal{M}$ is nothing more than an equilateral element with respect to the norm $|.|_{\mathcal{M}}$. This definition extends directly to the case of non-constant metric field $\mathbf{M}$, using $\ell_{\mathbf{M}}$ instead of $|.|_{\mathcal{M}}$. A mesh $\mathcal{H}$ is then said to be unit with respect to a metric field $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x} \in \Omega}$ if every element $\mathcal{T} \in \mathcal{H}$ is unit with respect to $\mathbf{M}$. In practical, the notion of unit element is replaced by the notion of quasi-unit element, that is to say that the edges of the concerned elements are close to 1, up to a given tolerance. Hence, in this paper, an element is said to be quasi-unit if all its edges $e_i, i = 1, \dots, d+1$, satisfy

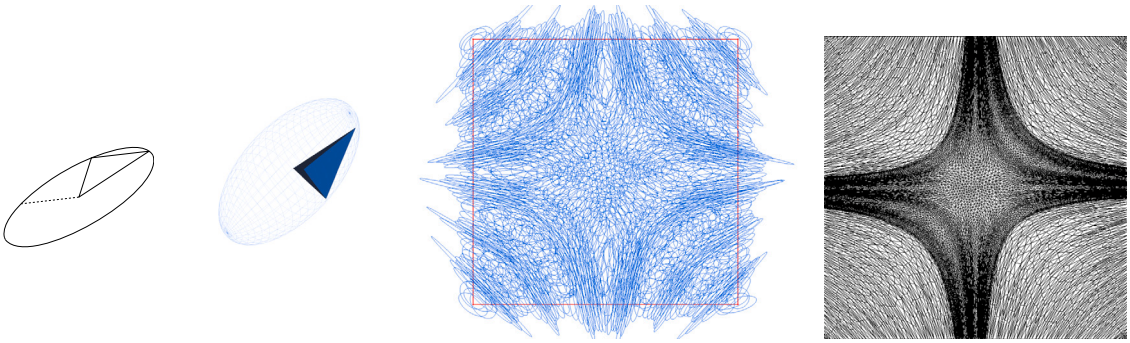


Fig. 2. Left: unit balls of 2D and 3D metrics and unit elements with respect to them. The dashed line on the ellipse represents a segment of length 1 with respect to the metric. Right: unit balls of a 2D metric-field on a square domain and unit mesh with respect to it.

$$\frac{\sqrt{2}}{2} \leq |e_i|_{\mathcal{M}} < \sqrt{2}. \tag{2.6}$$

In particular, if an edge e is too long, that is to say $|e|_{\mathcal{M}} \geq \sqrt{2}$, then splitting e at its middle would not create edges with too short lengths, smaller than $\frac{\sqrt{2}}{2}$. According to these definitions, there is a strong duality between a metric-field and a mesh, as shows Fig. 2. Most of the classical notions usually concerning meshes have their equivalence in terms of metric-field. For instance, if $\mathcal{T}$ is a unit tetrahedron with respect to a metric $\mathcal{M}$, then its volume is given by

$$|\mathcal{T}| = \frac{\sqrt{2}}{12} \det\left(\mathcal{M}^{-\frac{1}{2}}\right).$$

Likewise, the complexity of a mesh $\mathcal{H}$ which is unit with respect to $\mathbf{M} = (\mathcal{M}(x))_{x\in\Omega}$ has its metric equivalent which is

$$C(\mathbf{M}) = \int_{\Omega} (\det \mathcal{M}(x))^{\frac{1}{2}}\, dx.$$

By this way, the initial mesh adaptation Problem 1 is reduced to a metric optimization problem, stated by Problem 2 below.

Problem 2. Given a smooth function $u : \Omega \to \mathbb{R}$, a fixed complexity $N > 0$ and $p \in [1, ..., +\infty]$, find the optimal metric space $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ such that a unit mesh $\mathcal{H}$ with respect to $\mathbf{M}$ minimizes the L^p interpolation error of u.

3. Error model

According to previous sections, for a given solution u on $\Omega \in \mathbb{R}^d$, $d = 2, 3$, and a fixed complexity $N > 0$, it remains to define what a suitable metric-field is, with complexity N and such that a unit mesh with respect to it would minimize the high-order interpolation error of u given by (2.1). It is well known that this $\mathbb{P}_k$ interpolation error is governed by the differential of order $k+1$ of u. Indeed, for a smooth function u defined on an element $\mathcal{T}$ and $p \in [2, +\infty)$, the following interpolation inequality holds:

$$\left\|u - \Pi_k u\right\|_{L^p(\mathcal{T})} \leq C\, |\mathcal{T}|^{\frac{k+1}{d}} \left\|d^{(k+1)}u\right\|_{L^p(\mathcal{T})}, \tag{3.1}$$

where $|\mathcal{T}|$ denotes the volume of $\mathcal{T}$ and $d^{(k+1)}u$ the $k+1$ differential form of u. It is a homogeneous polynomial of degree $k+1$. In dimension 2, for $X_0 = (x_0, y_0)$, it is given by

$$d^{(k+1)}u(X_0)(X - X_0) = \sum_{i=0}^{k+1} \binom{k+1}{i} \frac{\partial^{(k+1)}u(X_0)}{\partial x^i \partial y^{k+1-i}} (x - x_0)^i (y - y_0)^{k+1-i}, \tag{3.2}$$

for all $X = (x, y) \in \mathbb{R}^2$. Likewise, in dimension 3, for $X_0 = (x_0, y_0, z_0) \in \mathcal{T}$, $d^{(k+1)}u(X_0)$, we get

$$d^{(k+1)}u(X_0)(X - X_0) = \sum_{i=0}^{k+1} \sum_{j=0}^{k+1-i} \binom{k+1}{i} \binom{k+1-i}{j} \frac{\partial^{(k+1)}u(X_0)}{\partial x^i \partial y^j \partial z^{k+1-i-j}} (x - x_0)^i (y - y_0)^j (z - z_0)^{k+1-i-j}, \tag{3.3}$$

for all $X = (x, y, z) \in \mathbb{R}^3$.

The inequality (3.1) can be found in a more general case in [9, Theorem 3.1.6]. This section shows how (3.1) can be translated into a metric-field, both in the cases of $\mathbb{P}_1$-based mesh adaptation and high-order based mesh adaptation.

3.1. Anisotropic $\mathbb{P}_1$ mesh adaptation

The metric-based mesh adaptation for $\mathbb{P}_1$ finite elements spaces has been introduced in the early 2000s and used in various cases, combined with the numerical resolution of partial differential equations (see for instance [3,11,17] and the references therein). The main idea relies on the interpolation error estimate (3.1) when $k = 1$. In this particular case, the interpolation error can be written

$$\left\|u - \Pi_1 u\right\|_{L^p(\mathcal{T})} \leq C\, |\mathcal{T}|^{\frac{2}{d}} \left\|H_u\right\|_{L^p(\mathcal{T})}, \tag{3.4}$$

where H_u denotes the Hessian matrix of u.

From (3.4), an optimal metric field can be derived, that is aligned with $\left|H_u\right|$, the absolute value of H_u, in terms of eigenvalues. More precisely, since H_u is a symmetric matrix of $\mathbb{R}^d$, it writes $H_u = R\,D\,R^t$, where $D \in \mathcal{M}_{d,d}(\mathbb{R})$ is the diagonal matrix whose diagonal is composed of the eigenvalues $\{\lambda_1, ..., \lambda_d\}$ and $R \in \mathcal{M}_{d,d}(\mathbb{R})$ is the rotation matrix whose columns are the corresponding eigenvectors of H_u. The absolute value of H_u is then defined by $|H_u| = R|D|R^t$, where $|D|$ is obtained by replacing $\{\lambda_1, ..., \lambda_d\}$ by their absolute values. The key point to the Hessian-based method described in [25,26] lies in the introduction of the metric-based error functional

$$E_p(\mathbf{M}, u) = \left(\int_\Omega \left| \text{trace}\left(\mathcal{M}^{-\frac{1}{2}}(x) \left|H_u(x)\right| \mathcal{M}^{-\frac{1}{2}}(x) \right) \right|^p dx \right)^{\frac{1}{p}}, \tag{3.5}$$

which is equivalent to $\|u - \Pi_1 u\|_{L^p}$ when the mesh $\mathcal{H}$ is unit with respect to $\mathbf{M}$. As shown in [24], a calculus of variations under the constraint $C(\mathbf{M}) = N$ then allows to compute the optimal metric-field which solves Problem 2, in the sense that it minimizes the right hand side of (3.5). It is given by

$$\mathcal{M}^p_{opt}(u)(x) = N^{\frac{2}{d}} \left(\int_\Omega \left(\det \left|H_u\right|\right)^{\frac{p}{2p+d}} \right)^{-\frac{2}{d}} \left(\det \left|H_u(x)\right|\right)^{-\frac{1}{2p+d}} \left|H_u(x)\right|. \tag{3.6}$$

For further details, we refer to [25,26].

From a practical point of view, given an initial coarse mesh $\mathcal{H}_0$ of the domain Ω and a fixed complexity $N > 0$, the $\mathbb{P}_1$ metric-based adaptation process follows Algorithm 1.

input : Initial mesh $\mathcal{H}_0$
Complexity N

output: Final mesh $\mathcal{H}_1$

repeat
- Compute $H_u(x)$, for each vertex x of $\mathcal{H}_0$
- Compute $\mathcal{M}(x) = \left(\det \left|H_u(x)\right|\right)^{-\frac{1}{2p+d}} \left|H_u(x)\right|$, for all x of $\mathcal{H}_0$
- Compute $\mathcal{M}^p_{opt} = \alpha\mathcal{M}$, with $\alpha > 0$ such that $C(\mathcal{M}^p_{opt}) = N$
- Remesh $\mathcal{H}_0$ and obtain $\mathcal{H}_1$ which is unit with respect to $\mathcal{M}^p_{opt}$
- Replace $\mathcal{H}_0$ by $\mathcal{H}_1$

until *convergence*;

Algorithm 1: $\mathbb{P}_1$ adaptation process.

3.2. From $\mathbb{P}_1$ to high-order error model

The most significant issue of the high-order problem comes from the fact that the right hand side of (3.1) cannot be expressed in a metric sense as easily as in the $\mathbb{P}_1$ interpolation case. Consequently, in order to extend the $\mathbb{P}_1$ mesh adaptation process to the $\mathbb{P}_k$ case, a further step in Algorithm 1 consists in approximating $d^{(k+1)}u(x)$ by a metric-dependent functional, for all $x \in \Omega$. As explained above, this issue does not occur when dealing with $\mathbb{P}_1$ finite elements spaces, for which the metric-dependent functional is the quadratic form defined from $\left|H_u(x)\right|$. On the contrary, if $k \geq 2$, the situation is more intricate. Let $u : \Omega \to \mathbb{R}$ be a smooth function and x_0 be a point of Ω. For all $x \in \Omega$ belonging to the vicinity of x_0, there exists a positive constant C such that

$$\left|u(x) - \Pi_k u(x)\right| \leq C \left| d^{(k+1)}u(x_0)(x - x_0) \right| + o\left(\left|x - x_0\right|_2^{k+1} \right), \tag{3.7}$$

where $|.|_2$ is the usual Euclidean norm.

The purpose is now to approximate the right hand side of (3.7) with a "metric approximation" of $d^{(k+1)}u$. More precisely, for all $x_0 \in \Omega$, we look for a metric $\mathcal{Q}(x_0)$ of $\mathbb{R}^d$ such that

$$\left| d^{(k+1)}u(x_0)(x) \right| \leq \left| x^t \mathcal{Q}(x_0)\, x \right|^{\frac{k+1}{2}}, \text{ for all } x \in \mathbb{R}^d. \tag{3.8}$$

Notice that the order $\frac{k+1}{2}$ of the right hand side of (3.8) has been chosen so that both sides of the inequality (3.8) are homogeneous with respect to x with order $k+1$. Then, assuming the existence of such a metric field $\mathbf{Q} = (\mathcal{Q}(x))_{x\in\Omega}$, the inequality (3.7) becomes

$$\left|u(x) - \Pi_k u(x)\right| \leq C \left|\left(x - x_0\right)^t \mathcal{Q}(x_0) \left(x - x_0\right)\right|^{\frac{k+1}{2}} + o\left(\left|x - x_0\right|_2^{k+1}\right). \tag{3.9}$$

From (3.9), following the same theoretical developments as the ones described in [24,25], an optimal metric field $\mathbf{M}$ can be deduced, minimizing the L^p high-order interpolation error when considering unit meshes with respect to $\mathbf{M}$. It is defined by

$$\mathcal{M}_{opt}^{p,k}(u)(x) = N^{\frac{2}{d}} \left(\int_\Omega (\det \mathcal{Q})^{\frac{p(k+1)}{2p(k+1)+2d}}\right)^{-\frac{2}{d}} (\det \mathcal{Q}(x))^{-\frac{1}{p(k+1)+d}} \mathcal{Q}(x), \tag{3.10}$$

which is obtained, through a computation of variations, as the metric field with complexity N which minimizes the continuous error estimate

$$E_p^k(\mathbf{M}, u) = \left(\int_\Omega \left|\text{trace}\left(\mathcal{M}^{-\frac{1}{2}}(x)\mathcal{Q}(x)\mathcal{M}^{-\frac{1}{2}}(x)\right)\right|^p dx\right)^{\frac{1}{p}}. \tag{3.11}$$

Notice that this functional is equivalent to the usual interpolation error $\left\|u - \Pi_k u\right\|_{L^p}$, if the considered mesh is unit with respect to $\mathbf{M}$. In particular, if $k = 1$ and $\mathcal{Q} = \left|H_u\right|$, the identity (3.10) reduces to the $\mathbb{P}_1$ optimal metric field defined by (3.6). This framework has been notably studied from a theoretical point of view by [5,6] and from a numerical point of view by [19], but in dimension 2. On the contrary to these works, the method which is introduced in the present article can be numerically implemented in both 2D and 3D, and its cost remains affordable, especially when comparing to the cost of PDE resolutions through high-order solvers.

From an initial mesh $\mathcal{H}_0$ of Ω, the whole $\mathbb{P}_k$ adaptation scheme follows Algorithm 2 below.

input : Initial mesh $\mathcal{H}_0$
Complexity N

output: Final mesh $\mathcal{H}_1$

repeat
- Compute $d^{(k+1)}u(x)$, for every vertex x of $\mathcal{H}_0$
- Compute $\mathcal{Q}(x)$ satisfying (3.8), for all $x \in \mathcal{H}_0$
- Compute $\mathcal{M}(x) = (\det |\mathcal{Q}(x)|)^{-\frac{1}{p(k+1)+d}} \mathcal{Q}(x)$, for all x of $\mathcal{H}_0$
- Compute $\mathcal{M}_{opt}^{p,k} = \alpha\mathcal{M}$, with $\alpha > 0$ such that $\mathcal{C}(\mathcal{M}_{opt}^{p,k}) = N$
- Remesh $\mathcal{H}_0$ and obtain $\mathcal{H}_1$ which is unit with respect to $\mathcal{M}_{opt}^{p,k}$
- Replace $\mathcal{H}_0$ by $\mathcal{H}_1$

until *convergence*;

Algorithm 2: $\mathbb{P}_k$ adaptation process.

Compared to the $\mathbb{P}_1$ adaptation process, the main additional difficulty comes from the construction of the metric field $\mathbf{Q}$ which satisfies (3.8). Moreover, in order to be accurate, this inequality has to be satisfied as sharp as possible, which involves both theoretical and numerical challenges. The next section addresses this fundamental issue, and introduces the log-simplex method, which is a way to get such a metric-field. It consists in replacing the non-linear optimization problem of finding the best metric satisfying (3.8) by a sequence of linear ones.

4. Metric based high-order differential approximate

Throughout this section, let p denote a homogeneous polynomial of order $k+1$, which basically is $p = d^{(k+1)}u(x_0)$, for $x_0 \in \Omega$. The aim is the definition of a metric $\mathcal{Q}$ which satisfies (3.8), in an optimal way. First, by dividing (3.8) by $p(x)$, we notice that, for all $x \in \mathbb{R}^d$,

$$|p(x)| \leq \left(x^t\, \mathcal{Q}\, x\right)^{\frac{k+1}{2}} \quad \Longleftrightarrow \quad 1 \leq \left(\left(\frac{x}{|p(x)|^{\frac{1}{k+1}}}\right)^t \mathcal{Q} \left(\frac{x}{|p(x)|^{\frac{1}{k+1}}}\right)\right)^{\frac{k+1}{2}}.$$

Hence, since p is homogeneous of order $k+1$, then $\left|p\left(\frac{x}{|p(x)|^{\frac{1}{k+1}}}\right)\right| = 1$ and the inequality (3.8) can be replaced by

$$1 \leq x^t \mathcal{Q}\, x, \quad \text{for all } x \text{ such that } |p(x)| = 1. \tag{4.1}$$

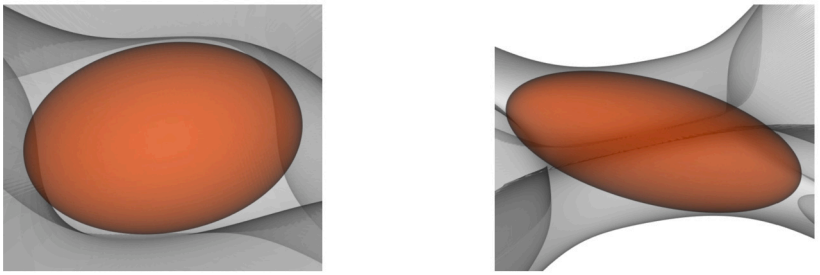


Fig. 3. Ellipsoids (in red) embedded into polynomial level sets (in grey), which satisfy the inequality (4.1). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

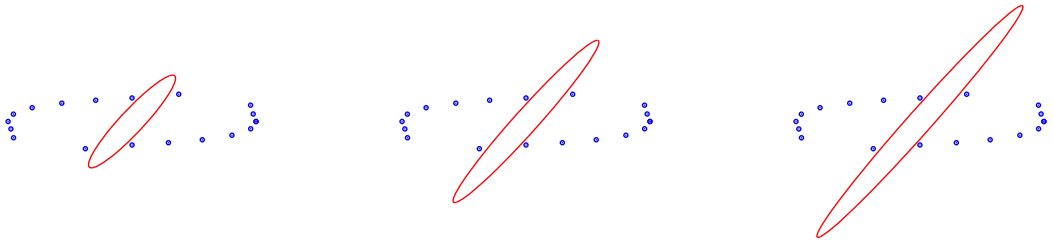


Fig. 4. Sequence of diverging metrics, which are embedded into a set of points belonging to the level set of a homogeneous polynomial.

From a geometrical point of view, the identity (4.1) translates the fact that the unit ball $\mathcal{B}_{\mathcal{Q}}$ of $\mathcal{Q}$ has to be embedded into the level-set of level 1 of p, as illustrated by Fig. 3. In order to be as accurate as possible, the considered metric should have the largest possible unit ball. Equivalently, the determinant of $\mathcal{Q}$ has to be minimized, leading to the following optimization problem.

Problem 3. Given a homogeneous polynomial p, find a metric $\mathcal{Q}$ of $\mathbb{R}^d$ such that

$$\begin{cases} \det \mathcal{Q} \quad \text{is minimal,} \\ 1 \leq x^t \mathcal{Q}\, x, \quad \text{for all} \quad x : |p(x)| = 1. \end{cases} \tag{4.2}$$

Unfortunately, Problem 3 brings two challenges. Firstly, the functional to minimize is the determinant of $\mathcal{Q}$, which is non-linear. Secondly, this problem has to be solved numerically. Consequently, the constraints can only be tested on a finite set of points $\{x_1, ..., x_n\}$ such that $|p(x_i)| = 1$, and not on the entire level set of $|p|$. In what follows a way to overcome those two issues is explained.

4.1. The approximated log-problem

As explained above, instead of dealing with the full level set of p, only a few points belonging to it are considered (around 30 points in the numerical simulations presented in Section 5). Consequently, given a set of points $\{x_1, ..., x_n\}$ such that $p(x_i) = 1$, for all $i \in \{1, ..., n\}$, Problem (4.2) reduces to discrete Problem 4 below.

Problem 4. Given a set of points $\{x_1, ..., x_n\}$ such that $|p(x_i)| = 1$ for all $i \in \{1, ..., n\}$, find a metric $\mathcal{Q}$ of $\mathbb{R}^d$ such that

$$\begin{cases} \det \mathcal{Q} \quad \text{is minimal,} \\ 1 \leq x_i^t \mathcal{Q}\, x_i, \quad \text{for all} \quad i \in \{1, ..., n\}. \end{cases} \tag{4.3}$$

As it is stated, Problem 4 is always ill posed. Indeed, it is always possible to find a straight path between two points, and build a sequence of metrics $(\mathcal{Q}_i)_{j \geq 1}$ satisfying the second line of (4.3) such that $\det \mathcal{Q}_j \to 0$ when j goes to infinity. Equivalently the volumes of the unit balls of such metrics go to infinity, as shown by Fig. 4.

In order to overcome this issue, we propose another approach based on the logarithmic interpretation of Problem (4.3). Let $\mathcal{L}$ be the logarithm matrix of $\mathcal{Q}$, which is obtained by replacing the positive eigenvalues of $\mathcal{Q}$ by their logarithm. More precisely, since $\mathcal{Q}$ is symmetric and positive definite, it can be written $\mathcal{Q} = R\, D\, R^t$, where

$$D = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \quad \text{and} \quad R = \left(v_1 | v_2 | v_3\right), \tag{4.4}$$

with $\{v_1, v_2, v_3\}$ the orthonormal eigenvectors of $\mathcal{Q}$ associated to its positive eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$. The logarithm matrix $\mathcal{L} = \log(\mathcal{Q})$ is then defined through the formula

$$\mathcal{L} = R \begin{pmatrix} \log(\lambda_1) & 0 & 0 \\ 0 & \log(\lambda_2) & 0 \\ 0 & 0 & \log(\lambda_3) \end{pmatrix} R^t. \tag{4.5}$$

Likewise, since both $\mathcal{Q}$ and $\mathcal{L}$ are diagonalizable matrices, it comes $\mathcal{Q} = \exp(\mathcal{L})$, in terms of matrix exponentials. Notice that $\mathcal{L}$ is not a metric but only a symmetric matrix. The main advantage which leads to consider $\mathcal{L}$ instead of $\mathcal{Q}$ comes from the fact that $\det(\mathcal{Q}) = \exp(\text{trace}(\mathcal{L}))$. Hence, the cost function $\mathcal{Q} \to \det(\mathcal{Q})$ which is not linear in $\mathcal{Q}$ is changed into $\mathcal{L} \to \text{trace}(\mathcal{L})$, which is linear in $\mathcal{L}$. The counterpart of this change of variable comes from the fact that the constraints of Problem 4 are not linear for $\mathcal{L}$, and become $x_i^t \exp(\mathcal{L})\, x_i \geq 1$, for all x_i. One of the main ideas of this article is stated in Proposition 4.1 below, and consists in approximating those non-linear constraints by linear ones, by taking advantage of usual convexity properties of the exponential function.

Proposition 4.1. *Let $\mathcal{Q}$ be a metric of $\mathbb{R}^d$ and $\mathcal{L}$ be the symmetric matrix $\mathcal{L} = \log(\mathcal{Q})$. If $x \in \mathbb{R}^d \setminus \{0\}$ satisfies*

$$x^t \mathcal{L}\, x \geq -|x|_2^2 \log\left(|x|_2^2\right), \tag{4.6}$$

then

$$x^t \mathcal{Q}\, x \geq 1. \tag{4.7}$$

Proof. Let $x \in \mathbb{R}^d \setminus \{0\}$ such that $x^t \mathcal{Q}\, x \geq 1$. Since $\mathcal{Q}$ is a symmetric positive definite matrix, it can be written $\mathcal{Q} = RDR^t$, with D and R defined by (4.4). Performing the change of variables $\tilde{x} = (\tilde{x}_1, ..., \tilde{x}_d) = R^t x$ and setting $\{\mu_1, ..., \mu_d\}$ the eigenvalues of $\mathcal{L}$, it comes

$$\begin{aligned} & x^t \mathcal{Q}\, x \geq 1 \\ \Leftrightarrow \quad & x^t \exp(\mathcal{L})\, x \geq 1 \\ \Leftrightarrow \quad & \sum_{j=1}^{d} \exp(\mu_j) \tilde{x}_j^2 \geq 1 \\ \Leftrightarrow \quad & \sum_{j=1}^{d} \exp(\mu_j) \frac{\tilde{x}_j^2}{|\tilde{x}|_2^2} \geq \frac{1}{|\tilde{x}|_2^2}, \end{aligned} \tag{4.8}$$

In particular, it is clear that $\displaystyle\sum_{j=1}^{d} \frac{\tilde{x}_j^2}{|\tilde{x}|_2^2} = 1$, and since R is a rotation matrix, then $|\tilde{x}|_2 = |x|_2$. Consequently, the convexity of the exponential function gives

$$\sum_{j=1}^{d} \exp(\mu_j) \frac{\tilde{x}_j^2}{|\tilde{x}|_2^2} \geq \exp\left(\sum_{j=1}^{d} \mu_j \frac{\tilde{x}_j^2}{|\tilde{x}|_2^2} \right) = \exp\left(\frac{x^t \mathcal{L}\, x}{|x|_2^2} \right). \tag{4.9}$$

Finally, the proof is achieved by noticing that

$$\exp\left(\frac{x^t \mathcal{L}\, x}{|x|_2^2} \right) \geq \frac{1}{|x|_2^2} \quad \Leftrightarrow \quad x^t \mathcal{L}\, x \geq -|x|_2^2 \log\left(|x|_2^2\right). \tag{4.10}$$

Indeed, assuming (4.10) and combining (4.9) and (4.8), it comes

$$x^t \mathcal{L}\, x \geq -|x|_2^2 \log\left(|x|_2^2\right) \quad \Longrightarrow \quad \sum_{j=1}^{d} \exp(\mu_j) \frac{\tilde{x}_j^2}{|\tilde{x}|_2^2} \geq \frac{1}{|\tilde{x}|_2^2} \quad \Longleftrightarrow \quad x^t \mathcal{Q}\, x \geq 1, \quad \square \tag{4.11}$$

Proposition 4.1 enables to consider the following alternative problem, instead of ill-posed optimization Problem (4.3).

Problem 5. Given a set of points $\{x_1, ..., x_n\}$ such that $|p(x_i)| = 1$ for all $i \in \{1, ..., n\}$, find a symmetric matrix $\mathcal{L}$ of $\mathbb{R}^d$ such that

$$\begin{cases} \text{trace}(\mathcal{L}) \text{ is minimal,} \\ x_i^t \mathcal{L}\, x_i \geq -|x_i|_2^2 \log\left(|x_i|_2^2\right), \quad \text{for all} \quad i \in \{1, ..., n\}. \end{cases} \tag{4.12}$$

Notice that the constraints on $\mathcal{L}$ in Problem 5 are an approximation of the ones on $\mathcal{Q}$ considered in Problem 4, in the sense that these are less restrictive. As a counterpart to this approximation, some further work, which is described in Section 4, needs to be done in order to recover a metric $\mathcal{Q}$ which accurately satisfies the constraints of Problem 4. The first advantage of formulation (4.12) comes from the fact that the problem is linear in $\mathcal{L}$, both for the cost function and the constraints. Furthermore, if the set $\{x_1, ..., x_n\}$ is wisely chosen, problem is now well-posed, since the trace of $\mathcal{L}$ cannot go to $-\infty$, or equivalently the determinant of $\mathcal{Q}$ cannot go to 0 nor the volume of the unit ball $\mathcal{B}_{\mathcal{Q}}$ goes to infinity.

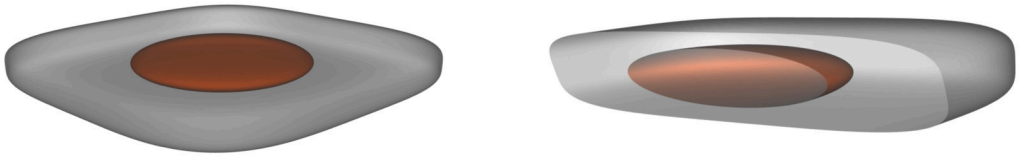


Fig. 5. Direct resolution of Problem 5 for a homogeneous polynomial of degree 4. The optimal metric (in red) is far from the level set of p (in grey).

Proposition 4.2. *Let $\mathcal{L}$ be a symmetric matrix of $\mathbb{R}^d$, with $d = 2, 3$. Assume that there exist d points $x_1, ..., x_d \in \mathbb{R}^d$ such that*

$$x_i^t \mathcal{L}\, x_i \geq -|x_i|_2^2 \log\left(|x_i|_2^2\right), \quad \text{for all} \quad i \in \{1, ..., d\}. \tag{4.13}$$

If $(x_1, ..., x_d)$ is an orthogonal basis of $\mathbb{R}^d$, then there exists a constant $C = C(x_1, ..., x_d) \in \mathbb{R}$ such that

$$\text{trace}(\mathcal{L}) \geq C. \tag{4.14}$$

Proof. The proof of this proposition is done in 3D, but the extension to the 2D case is straightforward. Let $\mathcal{L}$ be a symmetric matrix of $\mathbb{R}^3$, given by

$$\mathcal{L} = \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}, \quad \text{with} \quad a, b, c, d, e, f \in \mathbb{R}. \tag{4.15}$$

Let (x_1, x_2, x_3) be an orthogonal basis of $\mathbb{R}^3$ satisfying (4.13). Up to a rotation, x_1, x_2 and x_3 are given by $x_1 = (r_1, 0, 0)$, $x_2 = (0, r_2, 0)$ and $x_3 = (0, 0, r_3)$ with $r_1, r_2, r_3 > 0$. The assumption (4.13) reads

$$\begin{array}{ll} ar_1^2 \geq -r_1^2 \log(r_1^2), & a \geq -\log(r_1^2), \\ dr_2^2 \geq -r_2^2 \log(r_2^2). \iff & d \geq -\log(r_2^2), \\ fr_3^2 \geq -r_3^2 \log(r_3^2). & f \geq -\log(r_3^2). \end{array} \tag{4.16}$$

Consequently, $\mathcal{L}$ satisfies

$$\text{trace}(\mathcal{L}) = a + d + f \geq -\left(\log(r_1^2) + \log(r_2^2) + \log(r_3^2)\right). \quad \square \tag{4.17}$$

Since the log-problem (4.12) is linear and well posed, it can be solved by a classical simplex algorithm (see for instance [12]). Indeed, in 3D, the matrix $\mathcal{L}$ can be viewed as the vector $w_{\mathcal{L}} = (a, b, c, d, e, f)$ of $\mathbb{R}^6$, such that

$$\mathcal{L} = \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}. \tag{4.18}$$

For each $i \in \{1, ..., n\}$, x_i denotes both a point of $\mathbb{R}^3$ and its coordinates (x_i, y_i, z_i). The log-problem (4.12) reduces to find a vector $w_{\mathcal{L}}$ which minimizes $w \mapsto \langle v, w \rangle_{\mathbb{R}^6}$ and satisfies

$$x_i^2 a + 2x_i y_i b + 2x_i y_i c + y_i^2 d + 2y_i z_i e + f z_i^2 \geq C_i, \quad \text{for all} \quad i \in \{1, ..., n\}, \tag{4.19}$$

where $\langle ., . \rangle_{\mathbb{R}^6}$ is the usual scalar product of $\mathbb{R}^6$, $v = (1, 0, 0, 1, 0, 1)$ and $C_i = -|x_i|_2^2 \log\left(|x_i|_2^2\right)$.

With this formalism, each constraint corresponds to an hyperplane of $\mathbb{R}^6$, and the whole set of constraints forms a simplex in $\mathbb{R}^6$ in which the admissible solutions are taken. In particular the direction $-v$ prevails among the others to make the linear cost function $w \mapsto \langle v, w \rangle_{\mathbb{R}^6}$ decrease. In particular, if there is an optimal admissible solution, it must be a node at the intersection of six hyperplanes. In 2D, the problem may be stated the same way, but it is slightly simpler, since a 2D symmetric matrix is defined through three components only.

4.2. The log-simplex algorithm

Of course, there is a counterpart to the approximated problem (4.12). Indeed, if we solve it directly with a simplex algorithm, the unit ball $\mathcal{B}_{\mathcal{Q}}$ of $\mathcal{Q} = \exp(\mathcal{L})$ may be far from the level set of value 1 of p (see Fig. 5). This is explained by the fact that the log-constraints of the problem (4.12) are often far from the original ones given by the problem (4.2), especially when the level set of p is highly anisotropic. In order to recover the initial constraints of (4.2), an iterative process is performed. The idea is to build a sequence $(\mathcal{Q}_j)_{j \in \mathbb{N}}$ of metrics satisfying the second line of (4.3), which converges to a metric $\mathcal{Q}$ solving (4.2). To go from the step j to the next step $j + 1$, the change of variable $y = \mathcal{Q}_j^{\frac{1}{2}} x$, which maps the unit ball $\mathcal{B}_{\mathcal{Q}_j}$ to the unit sphere (or circle in 2D), is done. More precisely, at the step j, Problem 5 is solved with the polynomial $p \circ \mathcal{Q}_j^{-\frac{1}{2}}$. This way, a symmetric matrix $\mathcal{L}_j$ is obtained in the mapped

space and the metric $\mathcal{Q}_{j+1}$ is computed through the inverse mapping $y \mapsto \mathcal{Q}_j^{-\frac{1}{2}} y$. In terms of metrics, if (x_1, x_2) are two points of $\mathbb{R}^d$ and (y_1, y_2) such that $y_1 = \mathcal{Q}_j^{\frac{1}{2}} x_1$ and $y_2 = \mathcal{Q}_j^{\frac{1}{2}} x_2$, then

$$y_1^t \exp(\mathcal{L}_j)\, y_2 = x_1^t \left(\mathcal{Q}_j^{\frac{1}{2}} \exp(\mathcal{L}_j) \mathcal{Q}_j^{\frac{1}{2}} \right) x_2. \tag{4.20}$$

Hence, the metric in the initial 3D space is given by $\mathcal{Q}_{j+1} = \mathcal{Q}_j^{\frac{1}{2}} \exp(\mathcal{L}_j) \mathcal{Q}_j^{\frac{1}{2}}$. Algorithm 3 summarizes this process. The choice of

input : Homogeneous polynomial p

output: Metric $\mathcal{Q}$ solving the Problem 3

$\mathcal{Q}_0 = I_3$
repeat
  Compute $\{y_1, ..., y_n\}$ such that $|p(\mathcal{Q}_j^{-\frac{1}{2}} y_i)| = 1$, for all $i \in \{1, ..., n\}$
  Get $\mathcal{L}_j$ through log-Problem 5 with input $\{y_1, ..., y_n\}$
  Compute $\mathcal{Q}_{j+1} = \mathcal{Q}_j^{\frac{1}{2}} \exp(\mathcal{L}_j) \mathcal{Q}_j^{\frac{1}{2}}$
  Replace $\mathcal{Q}_j$ by $\mathcal{Q}_{j+1}$
until *convergence*;

Algorithm 3: Log-simplex algorithm.

the points $\{y_1, ..., y_n\}$ on the level-set of level 1 of the polynomial p is done in order to be in the conditions of the Proposition 4.2. More precisely, let $\{z_1, ..., z_n\}$ be a set of points of S^2, the unit circle in 2D and unit sphere in 3D, such that d of them define an orthonormal basis of $\mathbb{R}^d$. Given a metric $\mathcal{Q}_j$, we obtain the set of points $\{y_1, ..., y_n\}$ which belong to the level set of $\left| p \circ \mathcal{Q}_j^{-\frac{1}{2}} \right|$ via the formula

$$y_i = \frac{z_i}{\left| p(\mathcal{Q}_j^{-\frac{1}{2}} z_i) \right|^{\frac{1}{k+1}}}. \tag{4.21}$$

Indeed, since p is homogeneous of degree $k+1$, we get

$$\begin{aligned} \left| p\left(\mathcal{Q}_j^{-\frac{1}{2}} y_i \right) \right| &= \left| p\left(\frac{\mathcal{Q}_j^{-\frac{1}{2}} z_i}{\left| p(\mathcal{Q}_j^{-\frac{1}{2}} z_i) \right|^{\frac{1}{k+1}}} \right) \right| \\ &= \left| \frac{1}{\left| p(\mathcal{Q}_j^{-\frac{1}{2}} z_i) \right|} p\left(\mathcal{Q}_j^{-\frac{1}{2}} z_i \right) \right| \\ &= 1. \end{aligned} \tag{4.22}$$

In the specific case where $\mathcal{Q}_j$ converges to a metric $\mathcal{Q}$, Theorem 4.1 ensures that $\mathcal{L}_j$ converges to 0 and that the linear constraints of Problems (4.3) and (4.12) become equivalent.

Theorem 4.1. *Let p be a homogeneous polynomial and $\{z_1, ..., z_n\}$ be a set of points of S^2 such that d of them define an orthonormal basis of $\mathbb{R}^d$. Let $(\mathcal{Q}_j)_{j \in \mathbb{N}}$ be the sequence of metrics of $\mathbb{R}^d$ defined by*

- $\mathcal{Q}_0 = I_d$
- $\mathcal{Q}_{j+1} = \mathcal{Q}_j^{\frac{1}{2}} \exp\left(\mathcal{L}_j\right) \mathcal{Q}_j^{\frac{1}{2}}$, *where $\mathcal{L}_j$ is an optimal solution of the log-problem (4.12) with the constraint points $\{y_1^j, ..., y_n^j\}$ defined by*

$$y_i^j = \frac{z_i}{\left| p(\mathcal{Q}_j^{-\frac{1}{2}} z_i) \right|^{\frac{1}{k+1}}}, \quad \text{for all} \quad i \in \{1, ..., n\}. \tag{4.23}$$

If the sequence $(\mathcal{Q}_j)_{j\in\mathbb{N}}$ converges to a metric $\mathcal{Q}$ then, for all i, the sequence $(y_i^j)_{j\in\mathbb{N}}$ converges to some $y_i \in \mathbb{R}^d$, $(\mathcal{L}_j)_{j\in\mathbb{N}}$ converges to $\mathcal{L} = 0$. Furthermore, we have:

$$y_i^t \mathcal{L}\, y_i \geq -\left|y_i\right|_2^2 \log\left(\left|y_i\right|_2^2\right) \quad \Longleftrightarrow \quad x_i^t \mathcal{Q}\, x_i \geq 1, \tag{4.24}$$

where $x_i = \mathcal{Q}^{-\frac{1}{2}} y_i$, for all $i \in \{1, \ldots, n\}$.

Proof. The proof of this theorem is almost straightforward. Assume that there exists a metric $\mathcal{Q}$ such that $\mathcal{Q}_j \to \mathcal{Q}$ when j goes to infinity. Since

$$\mathcal{L}_j = \log\left(\mathcal{Q}_j^{-\frac{1}{2}}\, \mathcal{Q}_{j+1}\, \mathcal{Q}_j^{-\frac{1}{2}}\right), \tag{4.25}$$

by passing to the limit when $j \to +\infty$, it is deduced that $\mathcal{L}_j$ converges to the symmetric matrix $\mathcal{L} = \log(I_d) = 0$. It is also clear that, since $(z_i)_{i=1,\ldots,n}$ is fixed, y_i^j converges for every i to $y_i = \dfrac{z_i}{p(\mathcal{Q}^{-\frac{1}{2}} z_i)^{\frac{1}{k+1}}}$ when $j \to +\infty$. In order to exhibit the constraints satisfied by z_i, letting j go to infinity in the constraint inequalities of (4.12) yields

$$0 \geq -\left|y_i\right|_2^2 \log\left(\left|y_i\right|_2^2\right) \Longleftrightarrow 0 \geq -\frac{1}{\left|p(\mathcal{Q}^{-\frac{1}{2}} z_i)\right|^{\frac{2}{k+1}}} \log\left(\frac{1}{\left|p(\mathcal{Q}^{-\frac{1}{2}} z_i)\right|^{\frac{2}{k+1}}}\right) \tag{4.26}$$

$$\Longleftrightarrow 0 \geq \log\left(\left|p(\mathcal{Q}^{-\frac{1}{2}} z_i)\right|^{\frac{2}{k+1}}\right) \tag{4.27}$$

$$\Longleftrightarrow 1 \geq \left|p(\mathcal{Q}^{-\frac{1}{2}} z_i)\right|^{\frac{2}{k+1}}. \tag{4.28}$$

Since, $\left|z_i\right|_2 = z_i^t \mathcal{Q}^{-\frac{1}{2}}\, \mathcal{Q}\, \mathcal{Q}^{-\frac{1}{2}}\, z_i = 1$, it comes

$$\frac{1}{\left|p(\mathcal{Q}^{-\frac{1}{2}} z_i)\right|^{\frac{2}{k+1}}} \geq 1 \Longleftrightarrow \frac{z_i^t \mathcal{Q}^{-\frac{1}{2}}\, \mathcal{Q}\, \mathcal{Q}^{-\frac{1}{2}}\, z_i}{\left|p(\mathcal{Q}^{-\frac{1}{2}} z_i)\right|^{\frac{2}{k+1}}} \geq 1 \tag{4.29}$$

$$\Longleftrightarrow \left(\frac{\mathcal{Q}^{-\frac{1}{2}} z_i}{\left|p(\mathcal{Q}^{-\frac{1}{2}} z_i)\right|^{\frac{1}{k+1}}}\right)^t \mathcal{Q} \left(\frac{\mathcal{Q}^{-\frac{1}{2}} z_i}{\left|p(\mathcal{Q}^{-\frac{1}{2}} z_i)\right|^{\frac{1}{k+1}}}\right) \geq 1 \tag{4.30}$$

$$\Longleftrightarrow x_i^t \mathcal{Q}\, x_i \geq 1, \tag{4.31}$$

which achieves the proof of Theorem 4.1. □

4.3. Infinite branches

By itself, the log-simplex algorithm is not sufficient to ensure that the sequence of metrics $(\mathcal{Q}_j)_{j\geq 1}$ defined by Theorem 4.1 converges. Actually, in most of the cases, the level set of the polynomial p, which represents the $k+1$ differential of a solution u, has infinite branches, that is to say a non vanishing point $x \in \mathbb{R}^d$ such that $p(x) = 0$. In those cases, the volumes of the unit balls of the metrics $\mathcal{Q}_j$ may grow up to infinity. There are actually two situations in which those cases may occur, which are the case of a degenerate polynomial whose kernel is a vector-space of dimension d and the case of a non-degenerate polynomial which vanishes on a vector-space of dimension $d^* < d$. Those two cases are treated separately.

4.3.1. Degenerate polynomials

In this case, which is the easiest to address, p is a polynomial who vanishes on a vector space of dimension d. For instance, in dimension 2 with $k = 2$, consider $p : (x, y) \to y^3$. The area included into the level-set of level 1 of $|p|$ is the set $\left\{ (x, y) \in \mathbb{R}^2 : -1 \leq y \leq 1 \right\}$ which is obviously 2-dimensional. One of the best "metric" satisfying Problem 3 would be

$$\mathcal{Q} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.32}$$

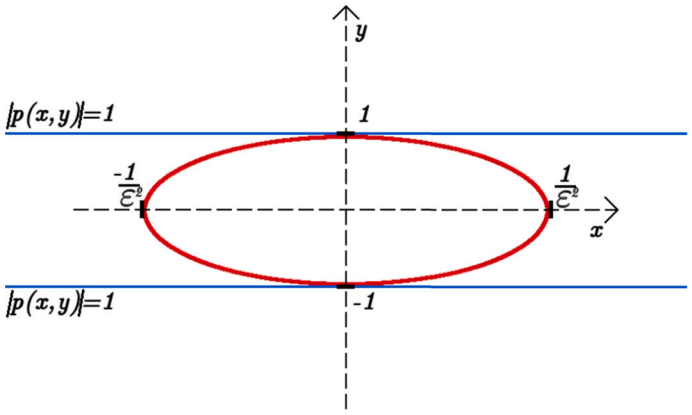


Fig. 6. A metric whose unit ball is embedded into the isoline of level 1 of the degenerate polynomial $p(x, y) = y^3$. In blue is printed the isoline of level 1 of $|p|$. In red is printed the unit ball of the metric $\mathcal{M}$ given by (4.33) with the vanishing eigenvalue replaced by ε.

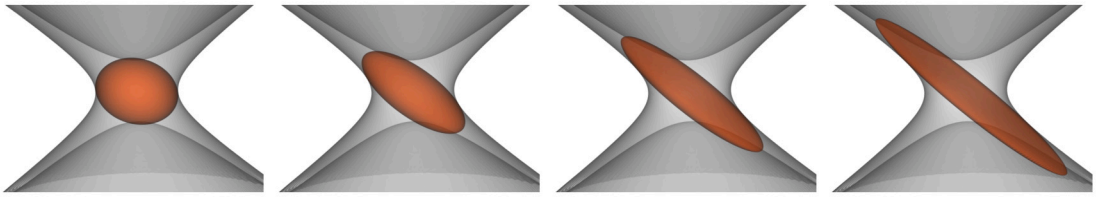


Fig. 7. A sequence of metrics whose volumes go to infinity (in red), while remaining included into the level-set of a polynomial of degree 2 (in grey).

Unfortunately, this matrix is not a metric, since it is positive but not definite, resulting in an infinite area of its unit ball $\mathcal{B}_{\mathcal{Q}}$. Thus, $\mathcal{Q}$ as defined by (4.32) is not usable as it is, because it would break the good definition of the norm $|.|_{\mathcal{Q}}$ given by (2.3). This issue is numerically solved by replacing $\mathcal{Q}$ by $\mathcal{Q}_\varepsilon$, defined by

$$\mathcal{Q}_\varepsilon = \begin{pmatrix} \varepsilon & 0 \\ 0 & 1 \end{pmatrix}, \tag{4.33}$$

where $\varepsilon > 0$ is a small variable replacing too small eigenvalues. Fig. 6 illustrates this process.

4.3.2. Non-degenerate polynomials with infinite branches

In most of the cases, even when p is not degenerated, the level-set has infinite branches. For instance, in dimension 2 with $k = 2$, consider $p(x, y) = (x + y)(x^2 + y^2)$. In this case, p vanishes on the set $\{(\lambda, -\lambda) : \lambda \in \mathbb{R}\}$ and the isoline of level 1 has a "hole" along this axis. When directly applying Algorithm 3 to this polynomial, the sequence of metrics $\mathcal{Q}_i$ may diverge by producing metrics whose unit balls remain embedded into the level-set while areas increase to infinity. Fig. 7 is a 3D illustration of this fact for a homogeneous polynomial of degree 2. In the $\mathbb{P}_1$ case, this problem is solved by taking the absolute value of the hessian matrix of u instead of H_u itself. Here, the same idea is followed and $d^{(k+1)}u$ is replaced by a function which is homogeneous of degree $k + 1$, without infinite branches. It is done through Proposition 4.3. Notice that a similar 2D version of this proposition holds, as well.

Proposition 4.3. *Let p be a homogeneous polynomial of degree k given by*

$$p(x, y, z) = \sum_{i+j+l=k} a_{i,j,l} \binom{k}{i, j, l} x^i y^j z^l, \quad \text{where} \quad a_{i,j,l} \in \mathbb{R} \quad \text{and} \quad \binom{k}{i, j, l} = \frac{k!}{i!j!l!}.$$

The polynomial p can be written

$$p(x, y, z) = \sum_{i+j+l=k-2} x^i y^j z^l \left(X^t H_{ijl} X \right), \tag{4.34}$$

where $X = (x, y, z)$ and $(H_{i,j,l})_{i+j+l=k-2}$ are explicit symmetric matrices given by

$$H_{ijl} = \begin{pmatrix} a_{i+2,j,l} & 2a_{i+1,j+1,l} & 2a_{i+1,j,l+1} \\ 2a_{i+1,j+1,l} & a_{i,j+2,l} & 2a_{i,j+1,l+1} \\ 2a_{i+1,j,l+1} & 2a_{i,j+1,l+1} & a_{i,j,l+2} \end{pmatrix} \tag{4.35}$$

Proof. The proof of this proposition is straightforward. □

This proposition leads to replace the homogeneous polynomial p by the function

$$q(x,y,z) = \sum_{i+j+l=k-1} \left| x^i y^j z^l \right| \left(X^t \left| H_{ijl} \right| X \right), \tag{4.36}$$

where k is the interpolation order and $(H_{i,j,l})_{i+j+l=k-1}$ is given by Proposition 4.3. Through this method, we are able to avoid infinite branches in the iterative resolution of the log-problem (4.12).

Given a mesh $\mathcal{H}$ of Ω and $d^{(k+1)}u(x)$ for all $x \in \mathcal{H}$, we are now able to compute the metric field $\mathcal{Q}$ at each point x of $\mathcal{H}$. It is done via Algorithm 4 below. Finally, Algorithm 2 is complete, and can be applied to compute the optimal metric field $\mathbf{M}^{\mathbf{p,k}}_{\mathbf{opt}} = \left(\mathcal{M}^{p,k}_{opt}(x)\right)_{x\in\mathcal{H}}$. Notice also that, in order to obtain a relevant metric space for mesh adaptation, some gradation techniques may be applied to correct too strong singularities in the metric field. In the present article, we use the metric smoothing as described in [1]. The next section is devoted to numerical applications of this method.

input : A mesh $\mathcal{H}$ of Ω
$d^{(k+1)}u(x)$, for all $x \in \mathcal{H}$

output: $\mathbf{Q} = (\mathcal{Q}(x))_{x\in\mathcal{H}}$

foreach $x \in \mathcal{H}$ **do**
  Replace $d^{(k+1)}u(x)$ by its reduced form q, as given by Proposition 4.3
  Perform Algorithm 3 with input q.
end

Algorithm 4: Approximate the $k+1$ differential.

5. Results

In this section, several academic test cases are presented and analyzed. While the first case considers an analytical function, other test cases result from the coupling of the adaptation library MAdLib[1] in which the method was implemented with the Discontinuous Galerkin (DG) solver Argo (see refs [20,35]). The adaptation library MAdLib can be coupled to any solver, the purpose of this coupling is to illustrate the adaptation method and how it can be beneficial for typical flow applications. The coupling is verified in the second test case proposed. Argo is a multiphysics software which includes several conservation laws to target incompressible or compressible flow applications, acoustic propagation or thermo-mechanical problems. The main advantages of DG solvers such as Argo is their high-order accuracy on unstructured meshes. Argo handles a wide variety of element types and is well suited for mesh adaptation because of the data structure used [20]. The compactness of the scheme makes it very efficient for large parallel application. Several Riemann solvers are implemented within Argo to treat convective fluxes in between elements. To discretize the diffusive part, two methods are implemented within Argo, interior penalty method and Bassi-Rebay 2 schemes. The interface with MAdLib allows to control 4 parameters of the adaptation procedure (i) the maximum anisotropy of the elements, (ii) the maximum gradient of element size, (iii) the minimum length of a cell and (iv) the complexity factor which roughly indicates the ratio in between final and initial number of elements. Some of those will be discussed in the different test cases presented. The solver encompasses several time integration schemes both implicit and explicit. Within the Argo solver, the adaptation can be used in steady or unsteady cases. For steady computation, the mesh is adapted when the residuals have been decreased by a prescribed factor or after a given number of iterations. Several adaptation steps can be performed. For unsteady computation, the adaptation is performed on local solution at a given frequency. Therefore, when using unsteady adaptation procedure, the maximum gradient and anisotropy should be controlled to ensure that refined regions are still of interest for latter time. The high-order results presented in this section are visualized using Vizir [27]. The third case shows that the conclusion on analytical scalar test case holds for more complex runs for which analytical solutions are not available. Finally, the robustness of the method for unsteady simulation and the influence of adaptation parameters are investigated in the last test case.

5.1. Analytical smooth shock function

On the unit cube $\Omega = \left[-\frac{1}{2}, \frac{1}{2}\right]^3$, the function $u(x,y,z) = 10\,\mathrm{atan}\,(100x) + \cos(yz)$ is considered. This later represents a smooth "shock" when x is close to 0, combined with small variations in the yz-axis, as shows Fig. 8. The log-simplex adaptation strategy is performed, for interpolation orders from 1 to 5. The process is started with an initial coarse mesh, on which a k-order representation of u is considered. Then, this mesh is adapted according to the log-simplex method described above. The resulting adapted mesh then becomes the initial mesh of the next iteration, for which we update the values of u. During this iterative process, the complexity is doubled from one step to the next one. Notice that only the point-wise values of the discrete $\mathbb{P}_k$ solution are used to recover the differential form of order $k+1$., *i.e.*, the exact derivatives of the function are never used. To do so, we have extended the

[1] https://sites.uclouvain.be/madlib/.

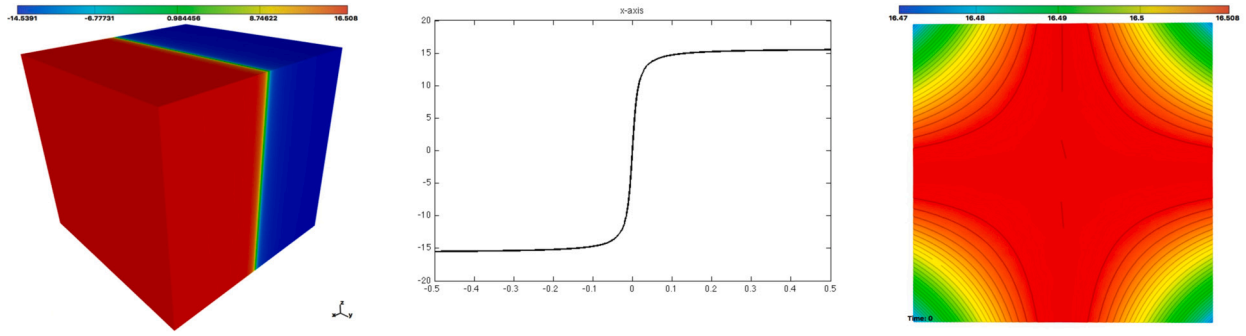


Fig. 8. Graph of the function $u(x, y, z) = 10\,\mathrm{atan}\,(100x) + \cos(yz)$ on $\Omega = \left[-\frac{1}{2}, \frac{1}{2}\right]^3$. Left: 3D mapping. Middle: 1D extraction along x-axis. Right: values of u in yz-axes on the cut-plane $x = 0.25$.

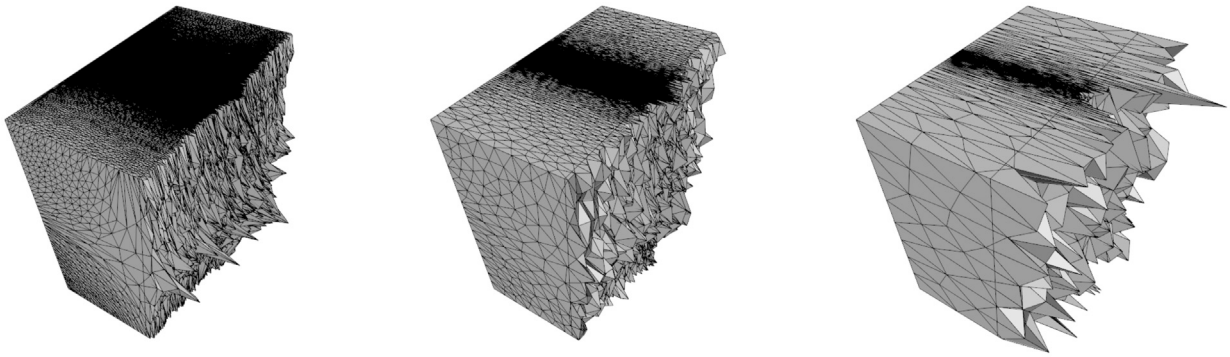


Fig. 9. $\mathbb{P}_1$, $\mathbb{P}_3$ and $\mathbb{P}_3$ adapted meshes for the function $u(x, y, z) = 10\,\mathrm{atan}\,(100x) + \cos(yz)$ on the unit cube $\Omega = \left[-\frac{1}{2}, \frac{1}{2}\right]^3$ with the same level of DoFs.

L^2 projection to the case of high-order differential forms (see [36]). More precisely, for each element $\mathcal{T}$ of the mesh, the k-order derivatives are computed from the exact polynomial representation provided by the solution values at the interpolation nodes. Then, this element-based derivatives are projected to the mesh vertices. Finally, a numerical gradient with respect to these k-order derivatives is performed, allowing to build the $(k+1)$ differential form of the solution at each vertex of the mesh. Once the numerical $(k+1)$ differential form is recovered, we apply the log-simplex algorithm and derive the optimal L^p metric $\mathcal{M}^{p,k}_{opt}(u)$ for a given complexity N. Fig. 9 shows the obtained adapted meshes. One interesting feature is the fact that adaptation differs from one order to the other. Indeed, as can be observed in Fig. 9, the $\mathbb{P}_1$ procedure aligns the elements with respect to the small variations in y and z, whereas the higher order adaptations do not take care of those variations, since they are already well represented by the high-order finite elements, even on large tetrahedra. The interpolation error $\|u - \Pi_k u\|_{\mathbf{L}^2(\Omega)}$ is computed using a 10^{th} order Gauss quadrature integration. In order to compare simultaneously different interpolation orders, the degrees of freedom (DoF) are used instead of the number of nodes. The error is then used to compare the convergence rate to the optimal one. According to the equalities (3.11) and (3.10), the interpolation error induced by a unit mesh with respect to the optimal metric field $\mathcal{M}^{p,k}_{opt}(u)$ satisfies

$$\left\|u - \Pi_k u\right\|_{L^p(\Omega)} \leq \frac{C}{N^{\frac{k+1}{3}}}, \quad \text{with} \quad C > 0. \tag{5.1}$$

We also compare the interpolation error with respect to sequences of uniform meshes. In this case, the asymptotic rate of convergence is reached for larger DoF for $k = 3, 4, 5$ due to the Gibbs phenomena that occur on small complexity (coarse) meshes. The most interesting feature is to see that the asymptotic rate of convergence is not reached for practical sizes in the case of uniform meshes, see Fig. 10 whereas this order is captured far earlier with adaptivity. For $\mathbb{P}_5$, we have roughly 10 orders of magnitude gain for the adapted mesh with respect to the uniform one. For the first 3 meshes for $k = 3, 4, 5$, the meshes are extremely coarse, less than 3 000 nodes. As a consequence, the computed interpolation error is massive, especially when making the comparison with meshes adapted to lower order solutions. Indeed, for high-order interpolation, the sharp variation around $x = 0$ and the coarseness of the mesh cause a Runge's phenomenon, that is to say artificial oscillations in the high-order polynomial representation of the solution. This notably emphasizes the sensitivity of high-order methods to the mesh quality. Table 1 gives the CPU costs related to u, where the adapted mesh with respect to the $\mathbb{P}_1$ finite elements is the finest obtained for this case, and the other ones have comparable numbers of degrees of freedom. For comparable numbers of DoFs, the full adaptation process, which includes metric computation and mesh adaptation, becomes cheaper as the order of interpolation increases. In details, the derivatives and metrics computation times increase together with the order of interpolation, but there are fewer elements do deal with. Consequently, the significant

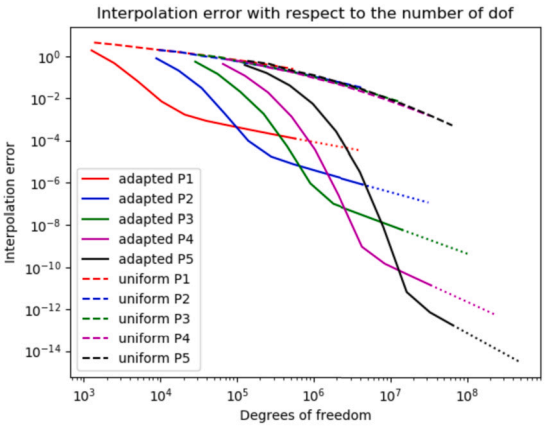


Fig. 10. Rate of convergence for the optimal meshes for order 1,2,3,4 and 5 and comparison with uniform meshes. Adapted meshes correspond to plain lines while uniform meshes to dashed lines. The same color corresponds to the same interpolation order. The doted lines represent the expected theoretical order of convergence.

Table 1
Comparison of computational cost adapting the mesh with different interpolation polynomial order.

	P1	P2	P3	P4	P5
degrees of freedom	2 463 091	2 299 983	1 926 453	2 299 810	2 219 674
interpolation error	6.8×10^{-6}	3.7×10^{-7}	2.3×10^{-8}	8.5×10^{-11}	1.2×10^{-11}
total CPU time (s)	547	517	382	138	144
derivatives (s)	3	96	99	61	70
metric field (s)	19	327	228	61	66
remeshing (s)	501	62	31	11	7

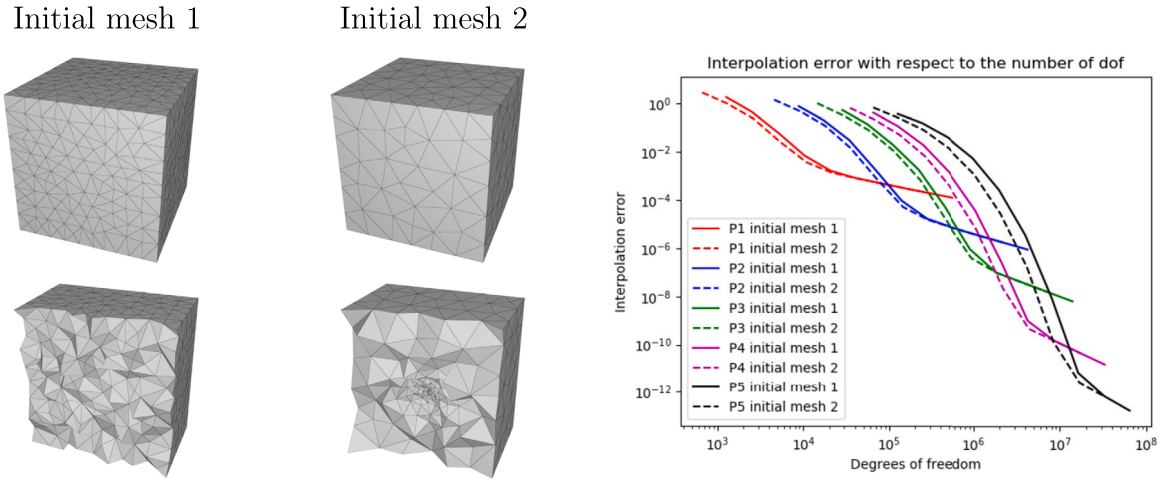


Fig. 11. Comparison of the convergence of the adaptation process with two different initial meshes. Left: uniform initial mesh 1 composed of 6387 tetrahedra. Middle: non-uniform initial mesh 2, composed of 3290 tetrahedra. Right: Interpolation error with respect to the number of degrees of freedom, starting from those two initial meshes.

gain in the remeshing part of the process makes the whole adaptation step faster when the interpolation order is higher. We also emphasize that, with the same number of DoFs, the interpolation error drastically decreases as the order of interpolation increases.

In order to emphasize the consistency of the method, another experience has been done for this test case, in which the same mesh adaptation process is performed with two different initial meshes. Fig. 11 shows the results for interpolation orders from 1 to 5. The first initial mesh is composed of isotropic tetrahedra with the same size on the whole domain. The second one is globally coarser than the first one, and is more refined at the center of the cube than in the remaining of the domain. As shows Fig. 11, for a given interpolation order $k \geq 1$, once the asymptotic convergence is reached, the two interpolation errors curves coincide. We can also see that the asymptotic convergence is reached faster when the coarser initial mesh is used, which is actually expected, and due to a further adaptation step in the early stages of the refining process.

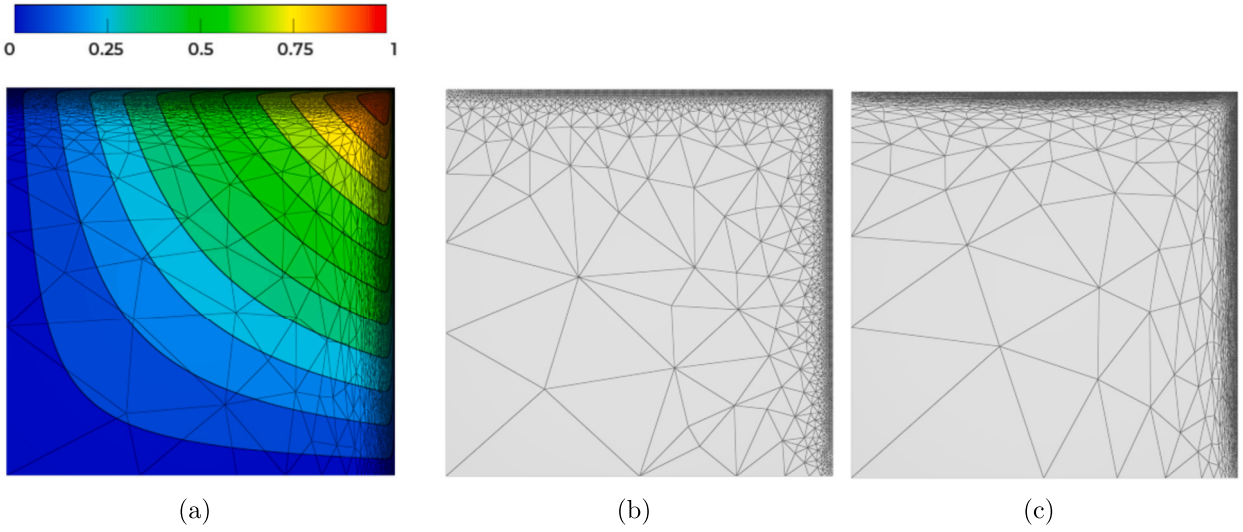


Fig. 12. (a) Scalar boundary layer solution interpolated with a fourth order polynomial on the respective adapted mesh with 33427 triangles. Subfigures (b) and (c) show the mesh adaptation on a second order polynomial solution for maximum anisotropy limited to 1 and 10 respectively with the same complexity.

5.2. Scalar boundary layer

The purpose of this test case is to verify the coupling of the adaptation library and the Argo solver. This test is proposed in Rangarajan et al. [33] and more information can be found therein. The steady computation of a convection-diffusion-reaction problem in 2D with stiff gradients close to the wall is investigated. The scalar equation solved is

$$\beta \cdot \nabla u - \epsilon \nabla^2 u = s, \tag{5.2}$$

where the reaction term "s" is computed using the manufactured solution

$$u(x,y) = \left(x + \frac{e^{\frac{x}{\epsilon}} - 1}{1 - e^{\frac{1}{\epsilon}}} \right) \left(y + \frac{e^{\frac{y}{\epsilon}} - 1}{1 - e^{\frac{1}{\epsilon}}} \right). \tag{5.3}$$

The velocity β is taken as $(1,1)$ on the domain $\Omega = (0,1)^2$, the solution is zero on the boundaries, and the diffusion coefficient is taken as $\epsilon = 0.005$. The solution is shown in Fig. 12 together with two examples of adapted meshes. The anisotropic adaptation in the thin boundary layer can be observed in the meshes.

First, the problem is solved for cartesian isotropic meshes using different resolutions and polynomial orders ($k = 1,2,3,4$). Then, for each initial isotropic mesh and each polynomial order, the adaptation is performed keeping the same complexity. The number of elements in the isotropic mesh and adapted mesh are roughly the same, in order to easily compare the results. The L^2 errors comparing the numerical and analytical solutions are plotted with respect to the number of DoFs in Fig. 13. The convergence order is retrieved for the solver when the mesh size becomes sufficiently small. Again, less degrees of freedom are required to reach the correct rate of convergence (see Equation (5.1)) when anisotropic adaptation is used.

Secondly, the effect of the maximum anisotropy authorized during the adaptation is studied and results can be observed in Fig. 13. Comparing the error convergence for cartesian mesh and isotropic adaptation shows already the advantages of adaptation procedure. Increasing the maximum anisotropy, not only the error decreases but the rate of convergence increases.

Finally, as shown in Section 5.1 and Fig. 9, adapted meshes will be different, depending on the polynomial interpolation order of the field used to compute the metric. To illustrate this feature, the interpolation error can be computed on meshes which have been adapted using a lower interpolation order than the solution. Fig. 14 compares the L_2 error convergence for (i) a 1st order polynomial interpolation on adapted mesh for which metrics are computed using $\mathbb{P}_1$ fields, (ii) a 4th order polynomial interpolation using same adapted meshes as (i), and finally (iii) a 4th order polynomial interpolation on a mesh adapted using a metric based on a $\mathbb{P}_4$ solution. For (ii), the optimal convergence rate is barely achieved, and the comparison with the interpolation error on the $\mathbb{P}_4$ adapted mesh (iii) emphasizes the importance of performing order dependent mesh adaptations.

Note that when high value of the maximum anisotropy parameter is used together with high polynomial interpolation this can impact accuracy of the discretization scheme. According to Hillewaert [20], for DG solvers using a symmetric interior penalty method, the choice of the penalty coefficient for the diffusive part is crucial especially when elements become largely anisotropic. An elementwise penalty coefficient should be selected comparing to a facewise method to treat strong anisotropic elements and keep the convergence order of the scheme [20]. In this case, the solver was not able to achieve nominal convergence rate for a $p = 4$ interpolation order on P_1 mesh when a maximum anisotropy is fixed to 100. Fig. 14 presents results limiting the anisotropy to 10 and using the elementwise penalty.

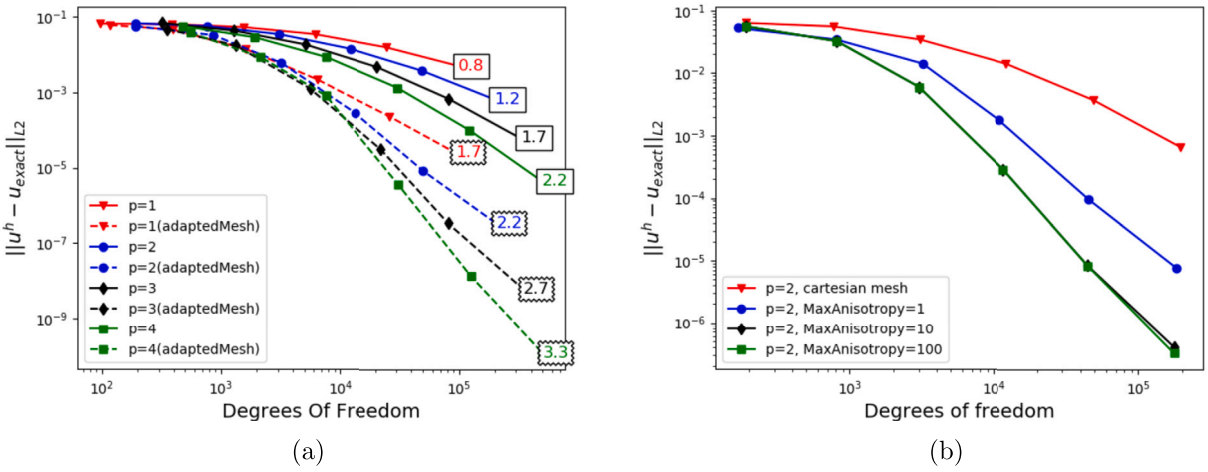


Fig. 13. Scalar boundary layer mesh convergence analysis for different polynomial orders (a) and influence of the anisotropy on the accuracy and convergence analysis (b).

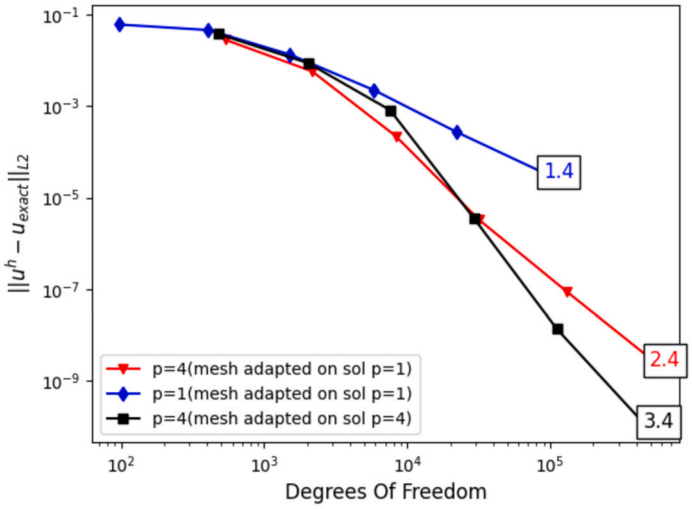


Fig. 14. Scalar boundary layer mesh convergence analysis using mesh adapted on $p = 1$ or $p = 4$ solution.

5.3. NACA0012

Anisotropic meshing has been proved to be beneficial to simulate compressible or incompressible flows (see for instance [28,37, 17,23]). The log-simplex adaptation method is tested on the solution of the flow past a NACA0012 airfoil. The steady incompressible flow pasts a NACA0012 profile at $Re = 5000$ and $M = 0.1$ oriented with an inclination of 2° is computed with the Argo solver. The mesh adaptation is led by the total pressure field and one of the adapted meshes is shown in Fig. 15.

First, to evaluate the gain of mesh adaptation, the errors are computed on the lift and drag coefficients C_L and C_D for adapted and non-adapted grids for different polynomial interpolation orders. Reference coefficients $C_{L,ref}$ and $C_{D,ref}$ are computed using a fine grid and an interpolation order $k = 4$. The errors are computed as

$$error_{CL} = 100 \cdot \frac{C_L - C_{L,ref}}{C_{L,ref}}, \tag{5.4}$$

$$error_{CD} = 100 \cdot \frac{C_D - C_{D,ref}}{C_{D,ref}}. \tag{5.5}$$

The lift and drag coefficients computed with the fine mesh are $C_{L,ref} = 0.0661$ and $C_{D,ref} = 0.0509$. Table 2 shows the error evolution with the polynomial interpolation. The accuracy gained by adaptation is clearly visible in the table for both lift and drag coefficients. Secondly, as observed on analytical test cases, the difference in anisotropy between the adaptation on $\mathbb{P}_1$ solution and $\mathbb{P}_4$ is clearly observed in Fig. 16. For the high order solution, the adapted mesh is less anisotropic thanks to the accuracy achieved by the high-

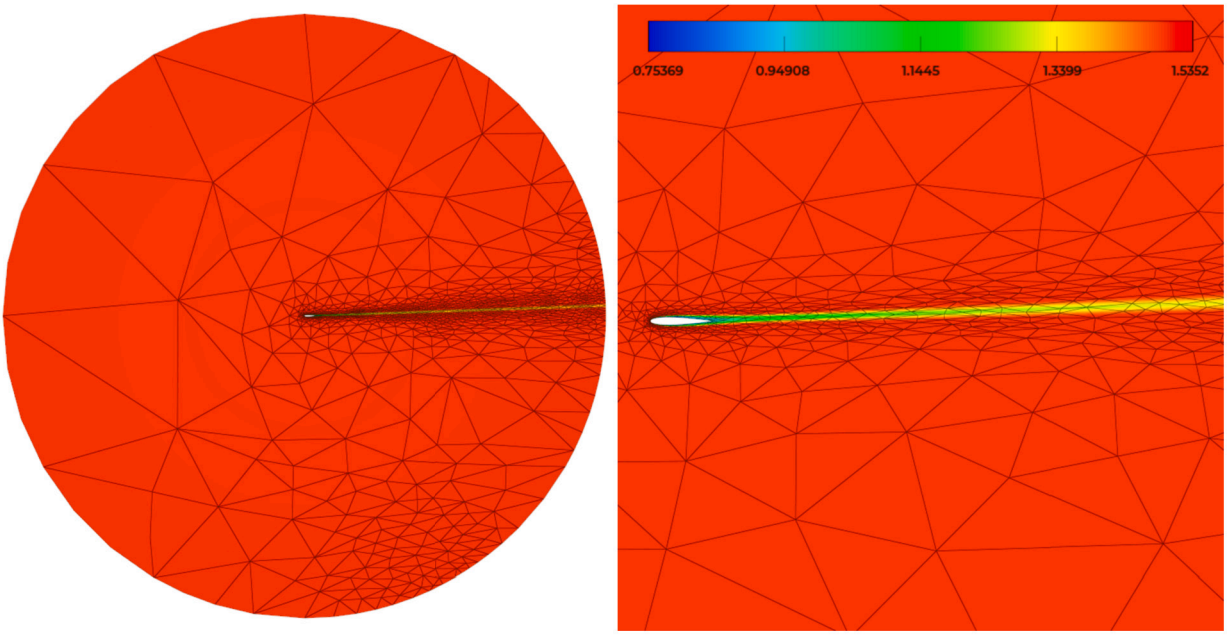


Fig. 15. Total pressure past a NACA012 airfoil at $Re = 5000$, $M = 0.1$ and with an angle of attack of 2°. Solution computed using the Argo solver at $k = 4$.

Table 2
Convergence of lift and drag coefficient for NACA0012.

	error_{CD}	error_{CD} (adapted)	error_{CL}	error_{CL} (adapted)
$k = 1$	3.27	2.88	110.34	72.16
$k = 2$	4.02	2.21	29.85	2.33
$k = 3$	4.18	1.11	9.33	2.18
$k = 4$	1.62	0.005	8.77	0.094

order polynomial. In this case, the authors have decided to keep the complexity constant hence both meshes contain roughly the same number of elements (≈ 2400 triangles).

Finally, the solver convergence to compute the steady solution is studied. For steady computation, Argo enables to perform several adaptation loops. For each adaptation iteration, the solver can either restart its convergence using the initial solution or use the solution obtained after convergence at previous iteration. In the latter, the solution is mapped from the old mesh to the new adapted mesh. To do so, the interpolation nodes of the new mesh are localized in the previous one, and the solution is then interpolated accordingly. For this test case, 4 iterations are performed, successively computing the solution and adapting the mesh with respect to it. Starting from the initial solution, the decrease of the residuals obtained with both restart methods can be compared in Fig. 17. Notice that the residual is non-dimensionalized with respect to the initial residual.

At the initial step, since both methods start from the same initial solution, the two residual curves exactly coincide, which is obviously expected. Surprisingly, at the second iteration, the computation for which the initial solution for the restart is taken from the previous computation converges almost twice slower than taking the arbitrary initial solution. This is due to the fact that, at the first iteration, the mesh is uniform and not sufficiently refined in some critical areas, such as the wake behind the airfoil. Consequently, the computed solution is quite far from the "true" one, and neither the mesh nor the solution are actually converged. However, from the third iteration, there is a massive gain in starting from the solution of the previous iteration, the solver converging more than three times faster than when keeping the initial solution for the restart. This highlights the impact of adapting the mesh, not only on the accuracy of the converged solution but also on the convergence of the solver itself.

5.4. Flow past a cylinder at Reynolds 185

In this 2D test case, the unsteady adaptation is tested on the compressible flow past a cylinder at Reynolds 185. At that Reynolds, a Von Karman vortex street can be observed. The domain of computation is $[-15D, 45D] \times [-15D, 15D]$, with $D = 1$ the diameter of the cylinder. The adaptation is performed every 5 time steps using the vorticity. An anisotropic mesh refinement following the vortex shedding can be observed in several snapshots shown in Fig. 18. A strong refinement in the boundary layer can be observed to capture the vorticity sheet in that region (see Fig. 19). As described before, the unsteady adaptation procedure adapts the mesh based on local solution at time t and this mesh is kept until $t + 5dt$. This can explain some shift comparing the vortices and the underlying mesh. Maximum gradient, anisotropy and remeshing frequency should be properly chosen to capture the wake with accuracy. Note

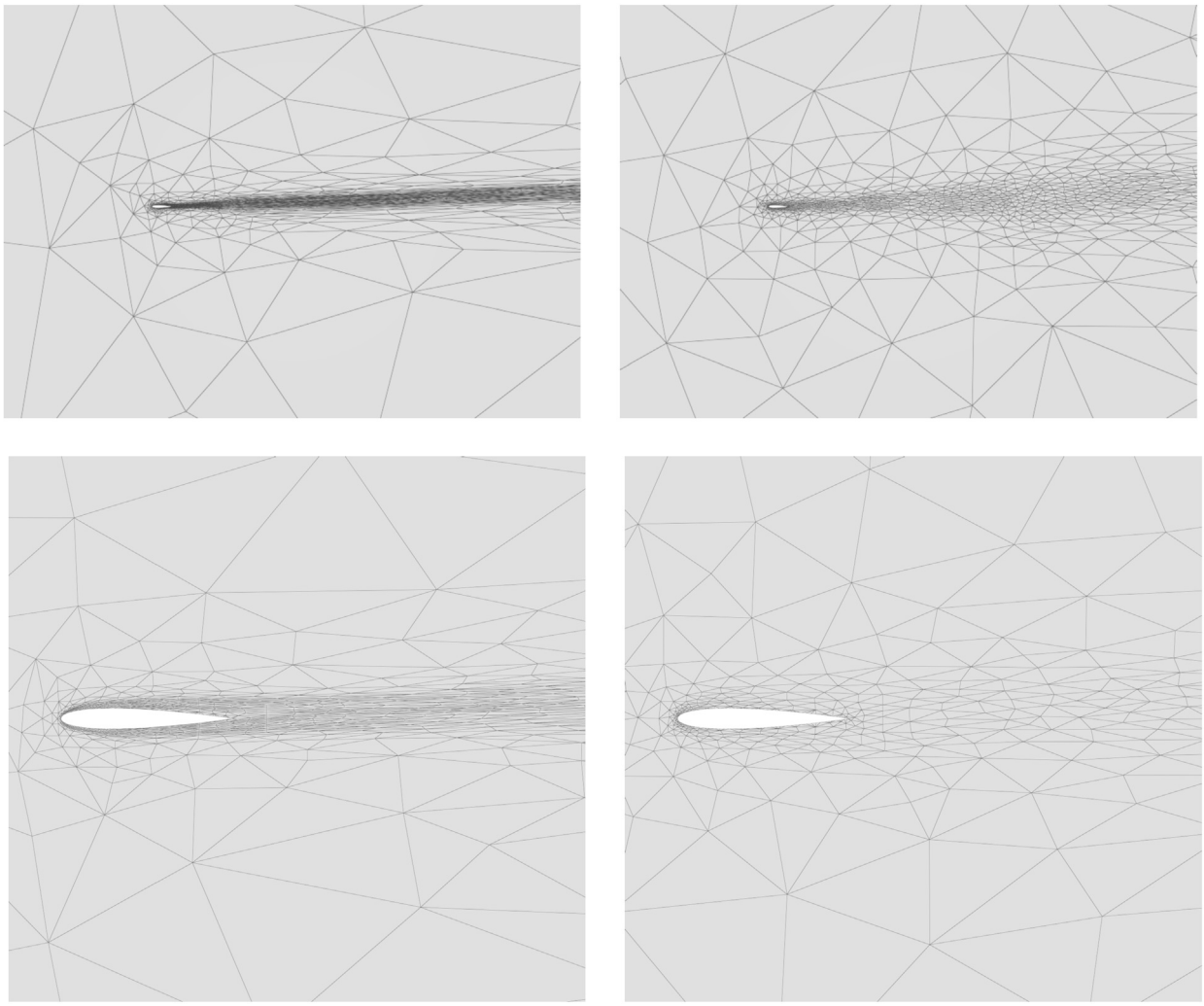


Fig. 16. Mesh adaptation in the wake of a NACA012 airfoil for $\mathbb{P}_1$ and $\mathbb{P}_4$ solutions. Left: adapted mesh with respect to $\mathbb{P}_1$ finite elements. Right: adapted mesh with respect to $\mathbb{P}_4$ finite elements. Both meshes have roughly the same number of triangles.

that increasing frequency could lead to higher mesh to mesh interpolation error. An interpolation of $k = 3$ is used for the conservative variables. The log-simplex method is using the vorticity to generate the metric hence adaptation is performed on a $k = 2$ field.

6. Conclusion

An adaptation procedure to generate anisotropic meshes with $\mathbb{P}_k$ interpolation has been introduced. It is based on an iterative algorithm to derive a local optimal metric which approximates a given $(k + 1)$ differential form. At each step, a simple linear log-simplex problem is solved in the logarithm space of metric fields. This optimal local metric is then globally optimized *via* a calculus of variations to obtain the optimal distribution of the DoFs in L^p norm. This strategy has been tested on various 2D and 3D examples where the optimal rate of convergence expected from (5.1) was exhibited, for both analytical and numerical solutions. For all the adaptive cases, the adapted meshes have a lower level error and reach faster the asymptotic rate of convergence. In addition, for a comparable number of DoFs, the total CPU time of the high order adaptation is generally much smaller than its equivalent for the $\mathbb{P}_1$ adaptation, due to the fewer number of elements which are dealt by the remesher in the high order case.

One limitation of the current work relies on the fact that it is efficient when the solution u is smooth. The most obvious perspective, which is currently investigated is to extend this strategy to $h - p$ adaptation and then coupling it with automatic shock detection for solutions involving both smooth and sharp features. This extension is currently investigated within the solver Argo and may be the topic of a further article. Notice also that the log-simplex algorithm can be used to produce curved meshes, adapted to a given parameterized geometrical model. Indeed, when having a parameterized surface, the gap between the mesh and the surface can be measured on the parameters space or on the tangent plane. Thus, a surface based mesh adaptation reduces to a 2D solution based

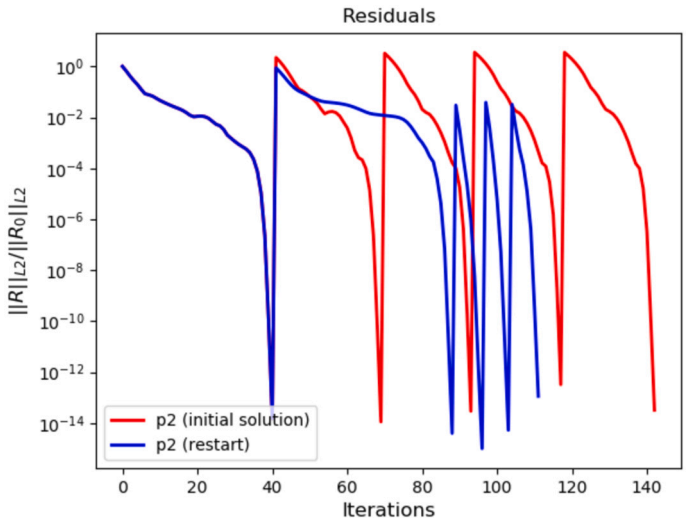


Fig. 17. Residual convergence for viscous flow past a NACA0012 at $k = 2$. Several adaptation steps are performed using the converged solution of the previous mesh (blue curve) or the initial solution (red curve). At the first iteration, the two curves coincide because both approaches start from the same initial solution.

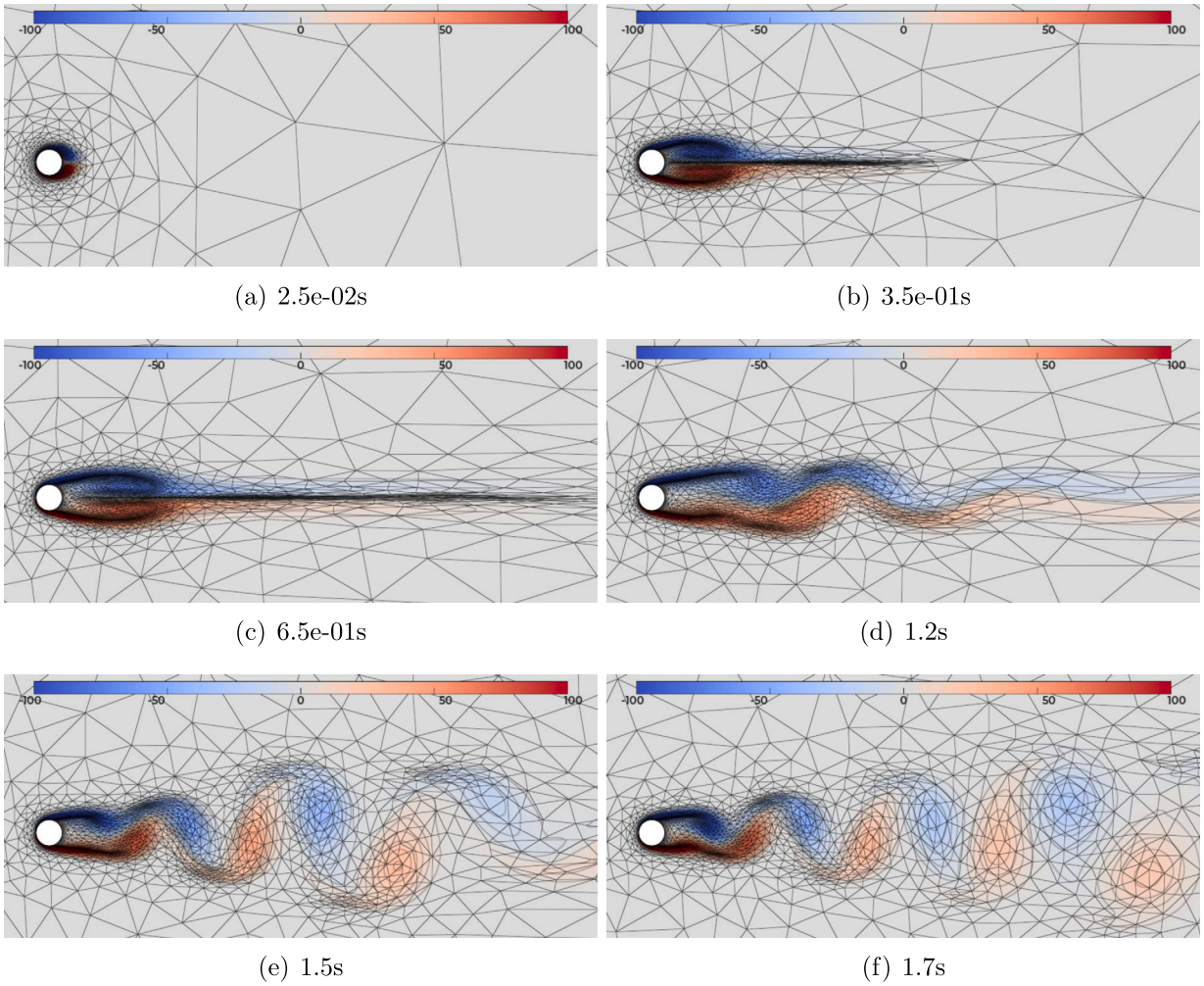
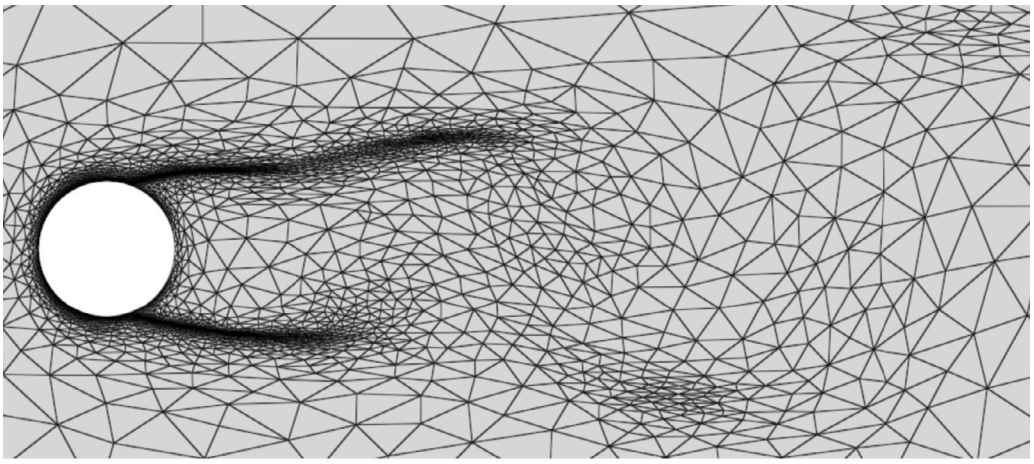
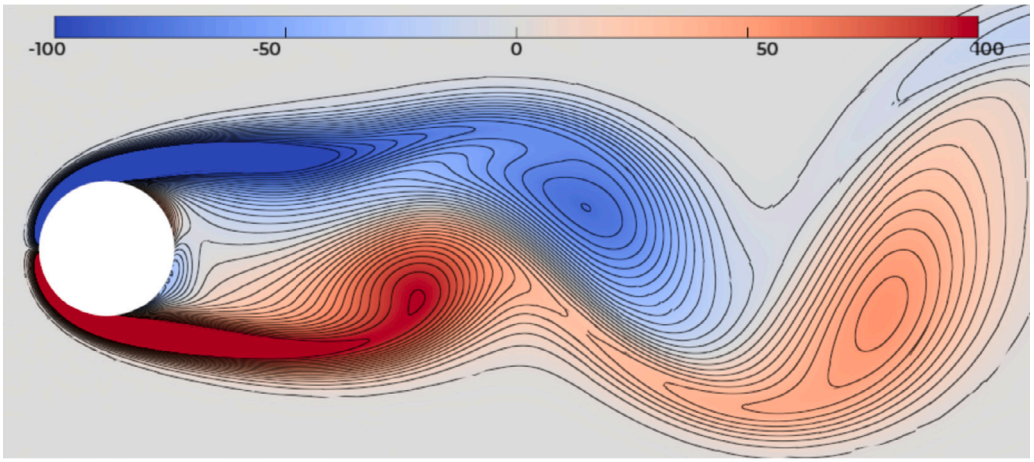


Fig. 18. Vortex shedding past a cylinder at Re = 185. Adaptation is performed on a $k = 2$ vorticity of the flow. Vorticity on the adapted mesh is shown at different time.



(a) mesh



(b) vorticity

Fig. 19. Adapted mesh close to the cylinder during vortex shedding at $t = 1.6$ s.

adaptation, and the order of the generated curved elements is linked to the order of the considered solution. This topic has been dealt in [16] and is still under investigation.

CRediT authorship contribution statement

Olivier Coulaud: Conceptualization, Investigation, Formal analysis, Writing, Software, Visualization. **Pierre Schrooyen**: Investigation, Writing, Software, Visualization. **Adrien Loseille**: Supervision, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] F. Alauzet, Size gradation control of anisotropic meshes, Finite Elem. Anal. Des. 46 (2010) 181–202.
[2] F. Alauzet, L. Frazza, D. Papadogiannis, Periodic adjoints and anisotropic mesh adaptation in rotating frame for high-fidelity rans turbomachinery applications, J. Comput. Phys. 450 (2022).
[3] F. Alauzet, A. Loseille, A decade of progress on anisotropic mesh adaptation for computational fluid dynamics, Comput. Aided Des. 72 (2016) 13–39.

[4] A. Balan, M. Michael Park, S. Wood, W. Anderson, Verification of Anisotropic Mesh Adaptation for Complex Aerospace Applications, 2020.
[5] W. Cao, An interpolation error estimate on anisotropic meshes in $\mathbb{R}^n$ and optimal metrics for mesh refinement, SIAM J. Numer. Anal. 45 (6) (2007) 2368–2391 (electronic).
[6] W. Cao, An interpolation error estimate in $\mathbb{R}^2$ based on the anisotropic measures of higher order derivatives, Math. Comput. 77 (261) (2008) 265–286 (electronic).
[7] M.J. Castro-Díaz, F. Hecht, B. Mohammadi, O. Pironneau, Anisotropic unstructured mesh adaption for flow simulations, Int. J. Numer. Methods Fluids 25 (4) (1997) 475–491.
[8] A. Chakraborty, A. Rangarajan, G. May, An anisotropic h-adaptive strategy for discontinuous Petrov-Galerkin schemes using a continuous mesh model, Comput. Math. Appl. 106 (2022) 1–17.
[9] P.G. Ciarlet, The Finite Element Method for Elliptic Problems, Classics in Applied Mathematics, vol. 40, SIAM, Philadelphia, PA, 2002.
[10] O. Coulaud, A. Loseille, Very high order anisotropic metric-based mesh adaptation in 3d, Proc. Eng. 163 (353–365) (12 2016) 353–365.
[11] T. Coupez, Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing, JCP 230 (7) (2011) 2391–2405.
[12] G.B. Dantzig, M.N. Thapa, Linear Programming 2: Theory and Extensions, Springer-Verlag, 2003.
[13] V. Dolejší, Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes, Comput. Vis. Sci. 1 (1998) 165–178.
[14] V. Dolejší, Anisotropic hp-adaptive method based on interpolation error estimates in the L^q-norm, Appl. Numer. Math. 82 (2014) 80–114.
[15] V. Dolejší, Anisotropic hp-adaptive discontinuous Galerkin method for the numerical solution of time dependent PDEs, Appl. Math. Comput. 267 (2015) 682–697, The Fourth European Seminar on Computing (ESCO 2014).
[16] R. Feuillet, O. Coulaud, A. Loseille, Anisotropic error estimate for high-order parametric surface mesh generation, in: 28th International Meshing Roundtable, 2019.
[17] P.J. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, Comput. Methods Appl. Mech. Eng. 194 (48–49) (2005) 5068–5082.
[18] E. Gauci, A. Belme, A. Carabias, A. Loseille, F. Alauzet, A. Dervieux, A priori error-based mesh adaptation in cfd, Methods Appl. Anal. 26 (2019) 195.
[19] F. Hecht, R. Kuate, An approximation of anisotropic metrics from higher order interpolation error for triangular mesh adaptation, J. Comput. Appl. Math. 258 (2014) 99–115.
[20] K. Hillewaert, Development of the discontinuous Galerkin method for large scale/high resolution CFD and acoustics in industrial geometries, PhD thesis, Université Catholique de Louvain, 2013.
[21] L. Kamenski, W. Huang, How a nonconvergent recovered Hessian works in mesh adaptation, SIAM J. Numer. Anal. 52 (4) (2014) 1692–1708.
[22] D.A. Kopriva, J.H. Kolias, A coonservative staggered-grid Chebyshev multidomain method for compressible flows, J. Comput. Phys. 125 (1996) 244–261.
[23] T. Leicht, R. Hartmann, Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulation, J. Comput. Phys. 229 (2010) 7344–7360.
[24] A. Loseille, Phd Thesis: Adaptation de maillage anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluides: Application à la prédiction haute-fidélité du bang sonique, PhD thesis, Université Pierre et Marie Curie - Paris VI, 2008.
[25] A. Loseille, F. Alauzet, Continuous mesh framework Part I: Well-posed continuous interpolation error, SIAM J. Numer. Anal. 49 (1) (2011) 38–60.
[26] A. Loseille, F. Alauzet, Continuous mesh framework Part II: Validations and applications, SIAM J. Numer. Anal. 49 (1) (2011) 61–86.
[27] A. Loseille, R. Feuillet Vizir, High-order mesh and solution visualization using opengl 4.0 graphic pipeline, in: Xiangmin Jiao, Jean-Christophe Weill (Eds.), 56th AIAA Aerospace Sciences Meeting, AIAA Scitech, 2008.
[28] D.J. Marvriplis, Multigrid strategies for viscous flow solver on anisotropic unstructured meshes, J. Comput. Phys. 145 (1998) 141–165.
[29] J.M. Mirebeau, Optimal meshes for finite elements of arbitrary order, Constr. Approx. 32 (2) (2010) 339–383.
[30] D. Pagnutti, C. Ollivier-Gooch, A generalized framework for-high order anisotropic mesh adaptation, Comput. Struct. 87 (11–12) (2009) 670–679, Fifth MIT Conference on Computational Fluid and Solid Mechanics.
[31] A. Rangarajan, A. Chakraborty, G. May, V. Dolejsi, A continuous-mesh optimization technique for piecewise polynomial approximation on tetrahedral grids, AIAA Fluid Dynamics Conference (2018).
[32] A. Rangarajan, A. Balan, G. May, Mesh optimization for discontinuous Galerkin methods using a continuous mesh model, AIAA J. 56 (10) (2018) 4060–4073.
[33] A. Rangarajan, G. May, V. Dolejsi, Adjoint-based anisotropic hp-adaptation for discontinuous Galerkin methods using a continuous mesh model, J. Comput. Phys. 409 (2020) 109321.
[34] R.J. Reed, T. Hill, Triangular mesh methods for the neutron transport equation, Technical report la-ur 73-473, Los Alamos, 1973.
[35] P. Schrooyen, Numerical Simulation of Aerothermal Flows through Ablative Thermal Protection Systems, PhD thesis, Université Catholique de Louvain, 2015.
[36] M.G. Vallet, C.M. Manole, J. Dompierre, S. Dufour, F. Guibault, Numerical comparison of some Hessian recovery techniques, Int. J. Numer. Methods Eng. 72 (2007) 987–1007.
[37] D.A. Venditti, D.L. Darmofal, Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows, J. Comput. Phys. 187 (2003) 22–46.
[38] M. Yano, D.L. Darmofal, An optimization-based framework for anisotropic simplex mesh adaptation, J. Comput. Phys. 231 (22) (2012) 7626–7649.