Adaptation de maillages pour des schémas numériques d'ordre très élevé

Estelle Carine Mbinky

Adaptation de maillages pour des schémas numériques d'ordre très élevé

Mesh adaptation for very high order numerical schemes

THÈSE

présentée pour obtenir le titre de
DOCTEUR
de l'Université Pierre et Marie Curie, Paris VI

École doctorale : Sciences Mathématiques de Paris Centre
Spécialité : Mathématiques Appliquées

par

Estelle Carine MBINKY

soutenue le 20 12 2013 devant le jury composé de :

Directeurs:		
Frédéric Alauzet	Chargé de recherche	INRIA Paris Rocquencourt
Adrien Loseille	Chargé de recherche	INRIA Paris Rocquencourt
Rapporteurs:		
Marco Picasso	Professeur	Ecole Polytechnique Fédérale de Lausanne
Thierry Coupez	Professeur	Mines-ParisTech
Examinateurs:		
Paul-Louis George	Directeur de recherche	INRIA Paris Rocquencourt
Alain Dervieux	Directeur de recherche	INRIA Sophia Antipolis
Pascal Frey	Professeur	Université Pierre et Marie Curie

Equipe projet Gamma3, INRIA Paris-Rocquencourt, 78 153 Le Chesnay.

Remerciements

Je tiens à adresser en premier lieu mes plus chaleureux remerciements à Paul-Louis George, responsable scientifique de l'équipe Gamma3, à mon directeur de thèse Frédéric Alauzet et mon co-directeur de thèse Adrien Loseille.

Je remercie Paul-Louis George et Frédéric Alauzet de m'avoir permis d'intégrer une équipe dynamique, chaleureuse et dont je partage les valeurs d'innovation. Je remercie Paul-Louis George dont j'ai pu apprécier sa dimension mathématique, mais aussi sa dimension humaine. Je remercie Frédéric Alauzet, qui n'a pas simplement accepté de diriger ma thèse; il m'a transmis sa passion, sa rigueur scientifique et son expérience en tant que chercheur. Je le remercie également pour son soutien et ses encouragements tout au long de cette thèse.

Je remercie Adrien Loseille qui représente pour moi un modèle de gentillesse, de patience, de motivation, d'enthousiasme et de rigueur. Je le remercie pour ses conseils, son dynamisme, sa passion et son soutien inconditionnelle durant ces trois dernières années de recherche.

Je remercie très sincèrement Thierry Coupez et Marco Picasso d'avoir accepté de rapporter ma thèse. Je les remercie d'avoir pris de leur temps pour comprendre, analyser mes travaux de recherche et soumettre des remarques pertinentes. Je les remercie également d'avoir accepté de faire partie de mon jury.

Je suis très honorée qu'Alain Dervieux et Pascal Frey aient accepté d'intégrer mon jury. Je tiens à leur adresser mes sincères remerciements. Je remercie tout particulièrement Alain Dervieux qui a suivi mes travaux de recherche et qui m'a apporté son soutien et son aide.

Je tiens également à remercier les membres de l'équipe Galaad à Sophia Antipolis pour leur accueil chaleureux. Je tiens à remercier en particulier Bernard Mourrain pour sa disponibilité, son aide et ses précieuses remarques qui m'ont permis d'avancer dans mes travaux.

Je souhaite remercier mes professeurs de l'Université Montpellier II, en particulier Bruno Koobus, Fabien Marche et Mohammadi Bijan, grâce à qui j'ai su développer mes connaissances et renforcer ma passion pour les Mathématiques Appliquées et le Calcul Scientifique.

Durant ces trois dernières années, j'ai eu la chance et le plaisir de cotôyer de nombreuses personnes chaleureuses et attachantes. Que tous soient remerciés pour les bons moments partagés. Je pense notamment aux membres de l'équipe Tropics à Sophia Antipolis, en particulier à Alain Dervieux, à Stephen Wornom, Alexandre Carabias, Hubert Alcin. J'adresse un immense remerciement aux membres de l'équipe Gamma3

et à Maryse, pour leur accueil, leur disponibilité, nos échanges scientifiques mais aussi et surtout leur gentillesse. Je n'oublie pas d'y inclure Julien Castelneau et Stéphanie Chaillat-Loseille.

Un remerciement particulier pour mes deux jeunes collègues Victorien et Nicolas qui m'ont soutenu et qui, par leur bonne humeur, m'ont permis de piquer des fous rires au moment où j'en avais tant besoin. Merci d'avoir supporté ma manie du rangement un peu excessive je l'avoue.

Je souhaite remercier mes frères Christian, Bertin, Stéphane, Richard et ma petite soeur Viviane, ma cousine Nardine et son mari Eugène Manta, mes oncles et tantes en France, au Portugal et au Sénégal pour leur soutien et leurs encouragements constants. Cette thèse, aboutissement de longues années d'études, je la dois avant tout à une personne d'exception: ma maman Caroline, décédée il y a 5 ans et sans qui je n'aurai trouvé la force et le courage d'aller jusqu'au bout. Il m'est impossible de trouver les mots pour dire à quel point je suis fière d'avoir eu une maman comme elle, pleine de valeurs et dévouée à ses enfants, à quel point je l'aime et surtout à quel point elle me manque. Cette thèse lui est dédiée.

Je souhaite également remercier mes amis: Anca, Afaf, Ibtihel, Alice, Richard James, Mina, Cécile, Martine, Sadrack, Murielle, Hassina, Sihem, Jessica, Néné Dia, Paul Byandé, ... pour avoir répondu présent à tout moment lorsque j'avais besoin d'eux et pour leurs encouragements. Merci pour vos conseils et votre soutien.

Pour conclure, je tiens bien évidemment à remercier Guillaume pour son soutien et ses encouragements quotidiens.

A la mémoire de
ma chère maman Caroline partie trop tôt.

Contents

Résumé

Le but de l'adaptation de maillages est de générer le meilleur maillage pour un problème donné, i.e, celui qui permettra d'obtenir la meilleure solution possible avec un nombre fixé de degrés de liberté. Pour cela, on met en place un processus itératif où l'on fait converger en même temps le couple maillage adapté-solution. Autrement dit, le processus d'adaptation de maillages consiste à changer localement la taille et l'orientation du maillage en fonction du comportement de la solution physique étudiée. Les méthodes d'adaptation de maillages ont prouvé qu'elles pouvaient être extrêmement efficaces en:

- réduisant significativement la taille des maillages pour une précision donnée (plusieurs ordres de grandeur),
- en atteignant rapidement une convergence asymptotique d'ordre 2 pour des problèmes contenant des singularités (ondes de choc, discontinuités de contact, points et lignes d'arrêt, ...) lorsqu'elles sont couplées à des méthodes numériques d'ordre élevé (i.e., ordre 2).

Dans les techniques d'adaptation de maillages basées sur les métriques, deux approches ont été proposées: les méthodes multi-échelles basées sur un contrôle de l'erreur d'interpolation en norme $\mathbf{L}^p$ et les méthodes ciblées à une fonctionnelle qui contrôle l'erreur d'approximation sur une fonctionnelle d'intérêt via l'utilisation de l'état adjoint. Cependant, avec l'émergence de méthodes numériques d'ordre très élevé, i.e, de schémas numériques d'ordre $\geqslant 3$ telles que la méthode de Galerkin discontinue ou les schémas aux résidus distribués, il devient nécessaire de prendre en compte l'ordre du schéma numérique dans le processus d'adaptation de maillages. En effet, pour un solveur d'ordre 3, on désire aussi contrôler un modèle d'erreur d'ordre 3 et non plus un modèle quadratique. Il est à noter que l'adaptation de maillages devient encore plus cruciale pour de tels schémas car ils ne convergent qu'à l'ordre 1 dans les singularités de l'écoulement. Par conséquent, le raffinement du maillage au niveau des singularités de la solution doit être d'autant plus important que l'ordre de la méthode est élevé.

L'objectif de cette thèse sera d'étendre les résultats numériques et théoriques obtenus dans le cas de l'adaptation pour des solutions linéaires par morceaux à l'adaptation pour des solutions d'ordre élevé qui sont polynomiales par morceaux. Ces solutions sont représentées sur le maillage par des éléments finis de Lagrange d'ordre $k \geqslant 2$ (isoparamétriques). A cette fin, cette thèse portera sur la modélisation de l'erreur

d'interpolation locale d'ordre $\mathbb{P}^{k+1}$ avec $k \geqslant 2$ dans le formalisme du maillage continu. En d'autres termes, on définit une erreur d'interpolation continue d'ordre $\mathbb{P}^{k+1}$ sur un maillage continu. Dans le cas de l'erreur d'interpolation, le modèle d'erreur est un polynôme homogène de degré $k \geqslant 3$. Or, les méthodes d'adaptation de maillages basées sur les métriques nécessitent que le modèle d'erreur soit une forme quadratique, laquelle fait apparaître intrinsèquement un espace métrique. Par conséquent, pour pouvoir exhiber un tel espace, il est nécessaire de décomposer le polynôme homogène et de l'approcher par une forme quadratique à la puissance $\frac{k}{2}$. Cette modélisation permet ainsi de révéler un champ de métriques indispensable pour communiquer avec le générateur de maillages. La méthode de décomposition utilisée est une extension de la méthode de diagonalisation au cas des polynômes homogènes de degré élevé.

En deux dimensions, la décomposition de Sylvester nous permettra d'approcher localement les variations de la fonction exacte d'erreur par un modèle d'erreur quadratique à la puissance $\frac{k}{2}$. Ensuite, ce modèle d'erreur local est utilisé pour contrôler globalement l'erreur en norme $\mathbf{L}^p$. Le maillage optimal est obtenu en minimisant cette erreur. On définit ainsi la méthode d'adaptation multi-échelle d'ordre élevé. Le cas de la dimension trois s'appuie sur la même méthodologie mais la modélisation de l'erreur locale est basée sur des méthodes de décomposition de tenseurs symmétriques: le modèle PARAFAC et l'extension la méthode de Sylvester à des dimensions plus grandes, en vue d'approcher les variations du modèle d'erreur.

Dans cette thèse, on s'attachera à démontrer la convergence à l'ordre k de la méthode d'adaptation de maillages pour des fonctions analytiques et pour des simulations numériques utilisant des solveurs d'ordre $k \geqslant 3$.

Introduction

In the context of scientific computing, there have been many efforts to extend the standard computational pipeline based on (linear) unstructured meshes to very high order meshes and solution. The emergence of high order numerical schemes has motivated the desire to develop appropriate mesh adaptation methods to fully benefit of the contribution of high order in terms of accuracy. However, the complete achievement of the higher-order computational pipeline is still a challenge as many problematics must be addressed.
First, from a mesh generation point of view, it turns out that using high order curved meshes to fit the geometry at hand is mandatory to reach the theoretical order of the underlying scheme. However, this task is tedious and is an active field of research.
Then, it is necessary to extend the underlying numerical schemes to higher-order interpolation, see for instance Discontinuous Galerkin methods.
Finally, the extension of mesh adaptation for a very high order method is only in 2D and the extension to 3D is barely tractable.

0.1 STATE OF THE ART

The development of high order numerical schemes such as Discontinuous Galerkin methods [8, 10, 81] or Residual distribution schemes [1] led to adaptation methods essentially based on a posteriori estimates that didn't allow to take into account the error in an anisotropic manner. In the case of second-order schemes and a priori estimates based on interpolation error, the TROPICS project and the GAMMA3 project have made several important advances in Multi-scale and Goal-Oriented mesh adaptation [2, 13, 69]. In terms of higher-order estimates for anisotropic mesh adaptation, there are few results but it is necessary to take them into account. However, the improvement of mesh adaptation to high order is limited to 2D [21, 22, 74, 54]. This thesis is a contribution for the development of **very high order anisotropic** mesh adaptation in 2D and 3D.

0.2 OUR APPROACH

We focus on the **error estimate** part and on the derivation of a proper metric field in 2D and 3D when a third order accurate interpolation is used. Our approach is based on a priori estimates based on interpolation errors. The idea is to model the high

order interpolation error in each point of the domain using the space of homogeneous polynomials. From this local error model, we approximate local optimal metric that will be used to generate the final optimal anisotropic mesh. To do so, we use symmetric tensor decomposition in order to diagonalize the local error model and deduce the best directions of the optimal metric. This is the same idea of what has been done in the linear interpolation case where the diagonalization of Hessian matrix is the main key of the method.

0.3 MY CONTRIBUTION

In this thesis, I focused on generating anisotropic adapted meshes using a priori estimates based on high order interpolation errors in $\mathbf{L}^p$-norm.
First, I used Sylvester method to diagonalize the local error model in 2D. I made some changes in Sylvester algorithm proposed by Comon and Mourrain [33] to decompose every homogeneous polynomial of degree 2 in two variables and treat every degenerated case. I have implemented this algorithm in **Metrix** [5] for the construction of metric fields.
Then, I used an extension of Sylvester method to decompose the local error model in 3D. I have implemented the symmetric tensor decomposition algorithm in **Metrix** for the approximation of metrics fields. This algorithm has been partially modify to better approximate the local optimal metrics.
Finally, I proceeded to multi-scale mesh adaptation based on high order error model to analyze and validate our approach.

0.4 ORGANIZATION AND CONTENT OF THESIS

The present thesis is organized as follow:

- In **Chapter 1**, we recall differential geometry concepts that will be a key component for the generation of adapted meshes. We begin by a review of metric tensors, Riemannian metric spaces and we detail metric based mesh generation. Then, operators on metric tensors, which are of main interest in mesh adaptation, are presented. We end by a review of the **continuous mesh framework** that has been proposed to mathematically model unstructured meshes.

- In **Chapter 2**, we present two **decomposition algorithms of symmetric tensors** of any degree and any dimension as a sum of powers of linear terms: the $\mathrm{CP3}_{alsls}$ algorithm and Sylvester's algorithms. These decompositions are the high order counter-part of symmetric matrix diagonalization as used in Hessian-based mesh adaptation. These decomposition methods will constitute the basic idea to construct metrics from high order interpolation error on each node of the mesh during the adaptive process.
  The main difficulty will be to assess the robustness of these decompositions for

numerically computed tensors. We start the chapter with a short review of multilinear algebra notions that will be used in the decomposition algorithms.

- In **Chapter 3**, we address the construction of local anisotropic metrics from high order interpolation error in 2D and 3D for mesh adaptation. Starting from high order local error model, the main idea is to approach locally the variations of this error by the variations of a quadratic definite positive form. Optimality conditions for the quadratic forms are derived and several algorithms based on symmetric tensor decomposition of **Chapter 2** are introduced.

- In **Chapter 4**, we extend the **multi-scale mesh adaptation** approach to high order interpolation. This approach aims at controlling the accuracy of the solution in the whole domain by a minimization of the global interpolation error in $\mathbf{L}^p$-norm. This global optimization problem is based on the results of the local optimization problem solved in **Chapter 3**.
  We begin by a review of the global optimization problem that has to be solved in the higher-order case. Then, the optimal mesh which minimizes the global interpolation error in $\mathbf{L}^p$-norm is exhibited. Afterwards, one of the classical recovery technique to find the third-order derivatives of the high order error model in 2D and 3D is presented. We finally present analytical examples for which optimal third-order anisotropic mesh adaptation, asymptotic convergence are reached with a high level of anisotropy.

SCIENTIFIC COMMUNICATIONS

Proceedings, Conferences, Workshops and Seminars

- **A priori-based mesh adaptation for higher-order accurate Euler simulation**, A. Carabias and E. Mbinky, European Workshop on High Order Nonlinear Numerical Methods for Evolutionary PDEs (HONOM 2013), Bordeaux, France, March 2013.

- **Higher-order interpolation for mesh adaptation**, E. Mbinky, seminar, INRIA Sophia-Antipolis, France, February 2013 (Oral).

- **Higher-order interpolation for mesh adaptation**, E. Mbinky, F. Alauzet and A. Loseille, 21^{th} International Mesh Roundtable (IMR 2012), Springer, San José, California (San Francisco Bay Area), October 2012 (Oral).

- 20^{th} International Mesh Roundtable (IMR 2012), Paris, France, Octobre 2011.

- **Multi-scale anisotropic mesh adaptation for a third-order accurate interpolation**, E. Mbinky, F. Alauzet, A. Loseille and A. Dervieux, European Congress on Computional Methods in Applied Sciences ans Engineering (ECCOMAS 2012), Vienna, Austria, September 2012 (Oral).

- **Interpolations d'order élevé et adaptation de maillages**, E. Mbinky, Congrés d'Analyse Numérique (CANUM 2012), Clermont-Ferrand, France, May 2012 (Poster).
- **High order interpolation and mesh adaptation**, E. Mbinky, junior seminar, INRIA-Rocquencourt, France, May 2012 (Oral).

1

Metric-based mesh adaptation

Contents

1.1 STATE OF THE ART

1.1.1 A short history of metric-based mesh adaptation

The idea of adapting the mesh associated with a numerical solution is very old. Since the 1960's, a rather large number of papers have been published on the subject. For instance, the query "Mesh Adaptation" on Google Scholar exhibits 7 130 000 results! In most of these works, the adaptation is isotropic and done by successive refinements of the elements according to predefined patterns. For instance, a square is split into four squares or a triangle is split into four triangles. The seminal idea of anisotropic mesh adaptation emerged later at the end of the 80's with error estimate and mesh generation concerns.

In 1987, Peraire et al. proposed a first attempt in 2D by providing error measures involving directions [78]. They pointed out the directional properties of the interpolation error and initiated the idea of generating elements with aspect ratios. They considered a local mapping procedure to generate elongated elements. They coupled this with an advancing front technique to generate slightly anisotropic meshes, i.e., elements having a $1:5$ ratio.
Similar approaches were considered in [67] and [83]. The first attempts in 3D were proposed in the early 1990's in [68] and [77], but numerical results were almost isotropic and the mesh anisotropy was not clearly visible. In 1994, Zienkiewicz gave a qualified status on the subject [93]. Despite some great successes with this new approach, they emphasized that: *"Unfortunately the amount of elongation which can be used in a typical mesh generation by such mapping is small..."*.

Almost at the same time on the meshing side, Mavriplis [73] suggested to generate stretched elements using a Delaunay approach in two dimensions in order to obtain high-aspect ratio triangles in boundary layers and wake regions as required by aeronautic numerical simulations. According to him, the Delaunay triangulation had to be performed in a locally stretched space: the idea of metric almost emerged. The year after, George, Hecht and Vallet [46] introduced the use of Riemannian metric tensor in a 2D Delaunay mesh generator to handle anisotropic adapted meshes. They exhibited that the absolute value of the Hessian of a given scalar solution is a metric. The edges length and elements volume inside the mesh generator were computed in the Riemannian metric space defined by the given metric field. They proposed to generate a uniform mesh in the Riemannian metric space, this mesh being adapted and anisotropic in the physical space.

The fruitful idea of metric was widely exploited for 2D anisotropic mesh adaptation in the 90's and even more today. For instance, among many others, see the works of [42, 26, 55, 39, 20]. In 1997, Baker gave a state-of-art [11] and wrote: *"Mesh generation in three dimensions is a difficult enough task in the absence of mesh adaptation and it is only recently that satisfactory three-dimensional mesh generators have become available*

[...]. Mesh alteration in three dimensions is therefore a rather perilous procedure that should be undertaken with care". Indeed, 3D meshing is much more complicated as new pathologies occur. The bare existence of such 3D meshes is not guaranteed. Doing 3D anisotropic mesh adaptation is even more complicated.

These bottlenecks have been partly solved by the development of local re-meshing techniques, which try to adapt the mesh by performing local modifications (insertion/deletion of vertices, vertices displacements, connectivity changes). One great asset of these techniques is to intrinsically get rid of the previous existence problem. At the beginning of the 2000's, first results with truly 3D anisotropic mesh adaptation were published [89, 76, 15, 12, 44, 48, 66].
In the meantime, new more accurate anisotropic error estimates have been proposed: a posteriori estimates [79, 40], a priori estimates [41, 6, 56] and goal-oriented estimates for functional outputs [90, 58, 71].

1.1.2 Current impact in scientific computing

Thanks to its generality, metric-beased mesh adaptation has been applied to various research fields and also used with a large panel of numerical methods. In all cases, it has brought large improvement in terms of accuracy and CPU performances. Just to give some 3D examples, it has been applied successfully to the the sonic boom simulation [4], multi-fluid flows [34, 49], blast problems [3], Stefan problems [12], metal forming processes [19],... It has also been coupled among which the Finite Volume [4], Finite Element [7], Stabilized Finite Element[19] and Discontinuous Galerkin Finite Element [81] methods.

Nowadays, there are a lot of meshing softwares based on the metric concept. Let us cite **Bamg** [52] and **BL2D** [64] in 2D, **Yams** [43] for discrete surface mesh adaptation and **Feflo.a** [72], **Forge3d** [36], **Fun3d** [58], **Gamanic3d** [47], **MAdLib** [35], **MeshAdap** [66], **Mmg3d** [38], **Mom3d** [89], **Tango** [15] and **LibAdaptivity** [76] in 3D. It is worth mentioning that all these softwares have arisen from different mesh generation methods. The method used in [47, 52] is based on a global constrained Delaunay kernel. In [64], the Delaunay method and the frontal approaches are coupled. [43, 72, 35, 38] are based on local mesh modifications and [36] is based on the minimal volume principle.

Nowadays, metric-based mesh adaptation has become a mature field of research which has now proved its relevance for steady and unsteady industrial problems.

1.2 BASICS OF METRIC-BASED MESH ADAPTATION

This section gives an overview of the most relevant concepts in differential geometry. For metric based mesh adaptation, we recall some essential notions of metric spaces which will be used in later chapters and will play a central role in the adaptation process.

For the sake of clarity, we recall the differential geometry notions that are used in the sequel. In the sequel, we use the following notations: bold face symbols, as $\mathbf{a}$, $\mathbf{b}$, $\mathbf{u}$, $\mathbf{v}$, $\mathbf{x}$, $\mathbf{e}$, ..., denote vectors or points of $\mathbb{R}^n$. Vectors coordinates are denoted by $\mathbf{x} = (x_i)_{i=1,\dots,n}$. The natural dot product between two vectors $\mathbf{u}$ and $\mathbf{v}$ of $\mathbb{R}^n$ is : $\mathbf{u} \,.\, \mathbf{v} = (\mathbf{u}, \mathbf{v})_{\mathcal{I}_n} = \sum_{i=1}^{n} u_i \, v_i$, with $\mathcal{I}_n$ the identity matrix.

1.2.1 Euclidian metric space

Definition 1.1 *An* ***Euclidian metric space*** $(\mathbb{R}^n, \mathcal{M})$ *is a vector space of finite dimension where the dot product is defined by means of a* ***Symmetric Definite Positive*** *tensor* $\mathcal{M}$*:*

$$\begin{array}{rccl} (.\,,.)_{\mathcal{M}}: & \mathbb{R}^n \times \mathbb{R}^n & \longrightarrow & \mathbb{R}^+ \\ & (\mathbf{u}, \mathbf{v}) & \longmapsto & \mathbf{u} \,._{\mathcal{M}}\, \mathbf{v} = (\mathbf{u}, \mathbf{v})_{\mathcal{M}} = (\mathbf{u}, \mathcal{M}\mathbf{v}) = {}^t\mathbf{u}\,\mathcal{M}\mathbf{v}. \end{array}$$

The matrix $\mathcal{M}$ is simply called a **metric tensor** or a **metric**.

The dot product defined by $\mathcal{M}$ makes $\mathbb{R}^n$ becomes a normed vector space $(\mathbb{R}^n, ||.||_{\mathcal{M}})$ and a metric vector space $(\mathbb{R}^n, d_{\mathcal{M}}(.\,,.))$ supplied by the following **norm** and **distance** definitions:

$$\begin{array}{rccl} ||\,.\,||_{\mathcal{M}}: & \mathbb{R}^n & \longrightarrow & \mathbb{R}^+ \\ & \mathbf{u} & \longmapsto & ||\mathbf{u}||_{\mathcal{M}} = \sqrt{(\mathbf{u}, \mathcal{M}\mathbf{v})}, \end{array}$$

and

$$\begin{array}{rccl} d_{\mathcal{M}}(.\,,.): & \mathbb{R}^n & \longrightarrow & \mathbb{R}^+ \\ & (\mathbf{u}, \mathbf{v}) & \longmapsto & d_{\mathcal{M}}(\mathbf{u}, \mathbf{v}) = ||\mathbf{u} - \mathbf{v}||_{\mathcal{M}}. \end{array}$$

In these spaces, we define geometric quantities which are of main interest when dealing with meshing:

- the **length** $\ell_{\mathcal{M}}$ of an edge $\mathbf{e} = \mathbf{ab}$ is given by:

$$\ell_{\mathcal{M}}(\mathbf{e}) = d_{\mathcal{M}}(\mathbf{a}, \mathbf{b}),$$

- the **angle** between two non zero-vectors $\mathbf{u}$ and $\mathbf{v}$ is defined by the unique real value $\theta \in [0, \pi]$ verifying:

$$cos(\theta) = \frac{(\mathbf{u}, \mathbf{v})_{\mathcal{M}}}{||\mathbf{u}||_{\mathcal{M}} ||\mathbf{v}||_{\mathcal{M}}},$$

- the **volume** of element K computed with respect to a metric tensor $\mathcal{M}$ is:

$$|K|_{\mathcal{M}} = \sqrt{\det \mathcal{M}}\, |K|_{\mathcal{I}_n},$$

where $|K|_{\mathcal{I}_n}$ is the Euclidian volume of the element K.

Spectral decomposition. As metric tensor $\mathcal{M}$ is a symmetric definite positive matrix, it is diagonalizable in a orthonormal basis:

$$\mathcal{M} = \mathcal{R}\, \Lambda\, {}^t\mathcal{R},$$

- $\mathcal{R}$ is an orthonormal matrix composed of the **eigenvectors** $(\mathbf{v}_i)_{i=1,\dots,n}$ of $\mathcal{M}$ verifying ${}^t\mathcal{R}\mathcal{R} = \mathcal{R}\,{}^t\mathcal{R} = \mathcal{I}_n$.
- $\Lambda = diag(\lambda_i)$ is the diagonal matrix composed of the **eigenvalues** of $\mathcal{M}$, denoted $(\lambda_i)_{i=1,\dots,n}$ and which are strictly positive.

Geometric interpretation of a metric tensor. We will often refer to the geometric interpretation of a metric tensor. In the vicinity of $\mathcal{V}(\mathbf{a})$ of a point $\mathbf{a}$, the set of points that are at distance ϵ, is given by:

$$\Phi_{\mathcal{M}}(\epsilon) = \{\mathbf{x} \in \mathcal{V}(\mathbf{a}) \,|\, {}^t(\mathbf{x}-\mathbf{a})\, \mathcal{M}\, (\mathbf{x}-\mathbf{a}) = \epsilon^2\}.$$

The above relation defines an ellipsoid centered at $\mathbf{a}$ with its axes aligned with the eigen directions of $\mathcal{M}$. Sizes along these directions are given by $h_i = \lambda_i^{-\frac{1}{2}}$. In the sequel, the set $\Phi_{\mathcal{M}}(1)$ is called the **unit ball** of $\mathcal{M}$ and we denote by $\mathcal{B}_{\mathcal{M}}$. This ellipsoid in 2D and 3D is depicted in **Figure 1.1**.

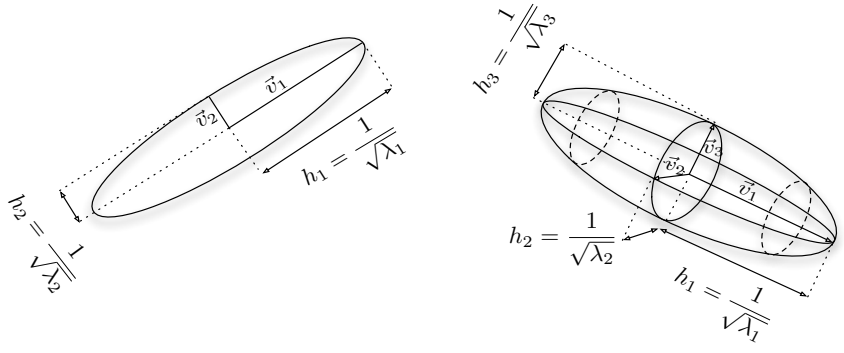

Figure 1.1: *Unit balls associated with metric $\mathcal{M} = \mathcal{R}\,\Lambda\,{}^t\mathcal{R}$ in 2D and 3D.*

Natural Mapping. From the previous definition and the spectral decomposition of $\mathcal{M}$, we deduce that application $\Lambda^{\frac{1}{2}}\,\mathcal{R}$ where $\Lambda^{\frac{1}{2}} = diag(\lambda_i^{\frac{1}{2}})$ defines the **mapping**

from the physical space $(\mathbb{R}^n, \mathcal{I}_n)$, where $\mathcal{I}_n$ is the identity matrix, to the Euclidean metric space $(\mathbb{R}^n, \mathcal{M})$:

$$\begin{aligned} \Lambda^{\frac{1}{2}}\mathcal{R}: \quad (\mathbb{R}^n, \mathcal{I}_n) &\longrightarrow (\mathbb{R}^n, \mathcal{M}) \\ \mathbf{x} &\longmapsto (\Lambda^{\frac{1}{2}}\mathcal{R})\,\mathbf{x}. \end{aligned}$$

And, we trivially recover: $\mathbf{u} \,._{\mathcal{M}}\, \mathbf{v} = {}^t\Big((\Lambda^{\frac{1}{2}}\mathcal{R})\,\mathbf{u}\Big)\,.\,\Big((\Lambda^{\frac{1}{2}}\mathcal{R})\,\mathbf{v}\Big) = {}^t\mathbf{u}\,\mathcal{M}\mathbf{v}$.

Notice that application $\Lambda^{\frac{1}{2}}\mathcal{R}$ maps $\mathcal{B}_{\mathcal{M}}$ from physical space into the unit ball in the metric space and, conversely, application ${}^t\mathcal{R}\,\Lambda^{-\frac{1}{2}}$ maps the unit ball into $\mathcal{B}_{\mathcal{M}}$, see **Figure 1.2**.

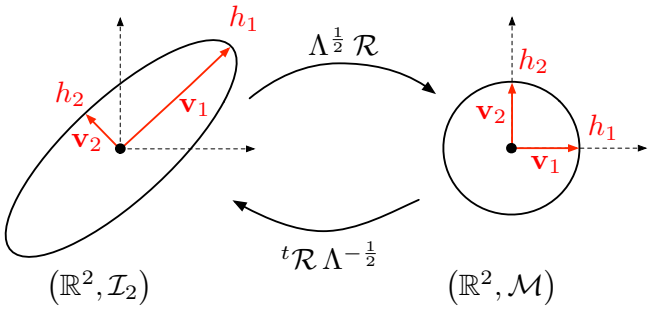

Figure 1.2: *Mappings between physical space $(\mathbb{R}^2, \mathcal{I}_n)$ and Euclidean metric space $(\mathbb{R}^2, \mathcal{M})$.*

1.2.2 Riemannian metric space

Definition 1.2 *In differential geometry, a **Riemannian manifold** or **Riemannian space** $(M, \mathcal{M})$ is a smooth manifold M in which each tangent space is equipped with a dot product defined by a metric tensor $\mathcal{M}$, a Riemannian metric space, in manner which varies smoothly from point to point. In other word, a Riemannian manifold is a smooth manifold in which the tangent space $T_{\mathbf{a}}M$ at each point $\mathbf{a}$ is a finite-dimensional Euclidean metric space $(T_{\mathbf{a}}M, \mathcal{M}(\mathbf{a}))$.*
*We denote the **Riemannian metric space** by $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$, with $\Omega \subset \mathbb{R}^n$ our computational domain.*

Even if no global definition of scalar product exists, in a Riemannian metric space, we can define various geometric notions $\mathbf{M}$ that takes into account the spatial variations of the metric:

- the **length** of edge $\mathbf{e} = \mathbf{ab}$ in Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is computed using the straight line parametrization in Ω, $\gamma(t) = \mathbf{a} + t\,\mathbf{ab}$, where $t \in [0, 1]$:

$$\ell_{\mathcal{M}}(\mathbf{e}) = \int_0^1 ||\gamma'(t)||_{\mathcal{M}}\,\mathrm{d}t = \int_0^1 \sqrt{{}^t\mathbf{ab}\,\mathcal{M}(\mathbf{a}+t\,\mathbf{ab})}\,\mathrm{d}t\,,$$

- the **angle** between two non-zero vectors $\mathbf{u}$ and $\mathbf{v}$ of Ω in Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is the unique real value $\theta \in [0, \pi]$ verifying:

$$cos(\theta) = \frac{(\mathbf{u}, \mathbf{v})_{\mathcal{M}(.)}}{||\mathbf{u}||_{\mathcal{M}(.)}||\mathbf{v}||_{\mathcal{M}(.)}},$$

- the **volume** of element K computed with respect to Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is:

$$|K|_{\mathcal{M}} = \int_K \sqrt{\det \mathcal{M}(\mathbf{x})}\, \mathrm{d}\mathbf{x}. \tag{1.1}$$

1.3 METRIC-BASED MESH ADAPTATION

Previously, Riemannian metric spaces have been introduced. Now, we illustrate how metric fields can be used in the context of mesh adaptation. Indeed, to generate anisotropic meshes, we have to prescribe at each point of the domain privileged sizes and orientations for each element. The information will be transmitted to the mesher which works in such spaces and changes locally the way of computing length, distances, angles and volumes.

The main idea of metric-based mesh adaptation, that has been initially introduced in [46], is to generate the **unit mesh** in the prescribed Riemannian metric space.

1.3.1 Unit element and unit mesh

***Definition 1.3** (**Unit element**)*
*An element K, defined by its list of edges $(\mathbf{e}_i)$, is **unit** with respect to a metric tensor $\mathcal{M}$ if the length of all its edges is unit in metric $\mathcal{M}$:*

$$\forall\, \mathbf{e}_i, \quad \ell_{\mathcal{M}}(\mathbf{e}_i) = \sqrt{{}^t\mathbf{e}_i\, \mathcal{M}\, \mathbf{e}_i} = 1.$$

If K is composed only of unit edges, then it is regular, e.g, its volume $|K|_{\mathcal{M}}$ in metric $\mathcal{M}$ is constant equal to:

$$|K|_{\mathcal{M}} = \begin{cases} \frac{\sqrt{3}}{4} \ \textit{in } 2D, \\ \\ \frac{\sqrt{2}}{12} \ \textit{in } 3D. \end{cases}$$

The notion of unit mesh is far more complicated than the notion of unit element as the existence of a mesh composed only of unit elements with respect to a given Riemannian metric space is not guaranteed. Consequently, this notion of unit mesh has to be relaxed. First, we give the following definition of **quasi-unit elements** that is also in practice used by mesh generators,

***Definition 1.4** (**Quasi-unit elements**)*
*An element K, defined by its list of edges $(\mathbf{e_i})$, is said to be **quasi-unit** for Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ if the following bounds are enforced:*

- *the length of the edges is given by:*

$$\forall\, \mathbf{e}_i, \quad \forall\, i \in [0, \frac{n(n+1)}{2}], \quad \ell_{\mathcal{M}}(\mathbf{e}_i) \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right],$$

- *the volume of the element is controlled via a quality function $Q_{\mathcal{M}}$ and a given bound $\alpha > 0$ by:*

$$Q_{\mathcal{M}}(K) = \frac{12}{3^{\frac{1}{2}}} \frac{|K|_{\mathcal{M}}}{\sum\limits_{i=1}^{3} \ell^2_{\mathcal{M}}(\mathbf{e_i})} \in [\alpha, 1] \quad \textit{in } 2D.$$

$$Q_{\mathcal{M}}(K) = \frac{36}{3^{\frac{1}{3}}} \frac{|K|^{\frac{2}{3}}_{\mathcal{M}}}{\sum\limits_{i=1}^{6} \ell^2_{\mathcal{M}}(\mathbf{e_i})} \in [\alpha, 1] \quad \textit{in } 3D,$$

$Q_{\mathcal{M}}(K) = 1$ corresponds to a perfect regular element, whatever its edges length, while $Q_{\mathcal{M}}(K) = 0$ indicates a null or degenerated element. Hence, the mesh adaptation software will try create elements with a quality near to 1.

We can now give the following definition of unit mesh.

***Definition 1.5** (**Unit mesh**)*
*A discrete mesh $\mathcal{H}$ of a domain $\Omega \subset \mathbb{R}^n$ is a **unit mesh** with respect to Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ if all its elements are a quasi-unit.*

Whatever the kind of desired mesh (uniform, adapted isotropic, adapted anisotropic), the mesh generator will always generate a unit mesh in the prescribed Riemannian metric space [46]. Consequently, the generated mesh is uniform and isotropic in the Riemannian metric space while it is adapted and anisotropic in the Euclidian space.

1.3.2 Useful operations on metrics

The main advantage when working with metric spaces is the well-posedness of operations on metric tensors, among which the **metric intersection** and **metric interpolation**. These operations have a straightforward geometric interpretation when considering the ellipsoid associated with a metric.

Metric intersection

When several metrics are specified at a point of the domain, all these metric tensors must be reduced to a single one. The **metric intersection** consists in keeping the most restrictive size constraint in all directions imposed by this set of metrics.

Formally speaking, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two metric tensors given at a point. The metric tensor $\mathcal{M}_{1\cap 2}$ corresponding to the intersection of $\mathcal{M}_1$ and $\mathcal{M}_2$ is the one prescribing the largest possible size under the constraint that the size in each direction is always smaller than the sizes prescribed by $\mathcal{M}_1$ and $\mathcal{M}_2$. Let us give a geometric interpretation of this operator. Metric tensors are geometrically represented by an ellipse in $2D$ and an ellipsoid in $3D$. But the intersection between two metrics is not directly the intersection between two ellipsoids as their geometric intersection is not an ellipsoid. Therefore, we seek for the largest ellipsoid representing $\mathcal{M}_{1\cap 2}$ included in the geometric intersection of the ellipsoids associated with $\mathcal{M}_1$ and $\mathcal{M}_2$. The ellipsoid (metric) verifying this property is obtained by using the simultaneous reduction of two metrics.

Simultaneous reduction. The simultaneous reduction enables to find a common basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ such that $\mathcal{M}_1$ and $\mathcal{M}_2$ are congruent to a diagonal matrix in this basis, and then to deduce the intersected metric. To do so, the matrix $\mathcal{N} = \mathcal{M}_1^{-1}\,\mathcal{M}_2$ is introduced. $\mathcal{N}$ is diagonalizable with real-eigenvalues. The normalized eigenvectors of $\mathcal{N}$ denoted by $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ constitute a common diagonalization basis for $\mathcal{M}_1$ and $\mathcal{M}_2$. The entries of the diagonal matrices, that are associated with the metrics $\mathcal{M}_1$ and $\mathcal{M}_2$ in this basis, are obtained with the Rayleigh formula[1]:

$$\lambda_i = {}^t\mathbf{e}_i\,\mathcal{M}_1\,\mathbf{e}_i \quad \text{and} \quad \mu_i = {}^t\mathbf{e}_i\,\mathcal{M}_2\,\mathbf{e}_i, \quad \text{for} \quad i = 1, ..., 3.$$

Let $\mathcal{P} = (\mathbf{e}_1\,\mathbf{e}_2\,\mathbf{e}_3)$ be the matrix the columns of which are the eigenvectors $\{\mathbf{e}_i\}_{i=1,...,3}$ of $\mathcal{N}$. $\mathcal{P}$ is invertible as $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ is a basis of $\mathbb{R}^3$. We have:

$$\mathcal{M}_1 = \mathcal{P}^{-t}\begin{pmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{pmatrix}\mathcal{P}^{-1} \quad \text{and} \quad \mathcal{M}_2 = \mathcal{P}^{-t}\begin{pmatrix}\mu_1 & 0 & 0\\ 0 & \mu_2 & 0\\ 0 & 0 & \mu_3\end{pmatrix}\mathcal{P}^{-1}.$$

Computing the metric intersection. The resulting intersected metric $\mathcal{M}_{1\cap 2}$ is then analytically given by:

$$\mathcal{M}_{1\cap 2} = \mathcal{M}_1 \cap \mathcal{M}_2 = \mathcal{P}^{-t}\begin{pmatrix}\max(\lambda_1,\mu_1) & 0 & 0\\ 0 & \max(\lambda_2,\mu_2) & 0\\ 0 & 0 & \max(\lambda_3,\mu_3)\end{pmatrix}\mathcal{P}^{-1}.$$

[1] λ_i and μ_i are not the eigenvalues of $\mathcal{M}_1$ and $\mathcal{M}_2$. They are spectral values associated with the basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$

The ellipsoid associated with $\mathcal{M}_{1\cap 2}$ is the largest ellipsoid included in the geometric intersection region of the ellipsoids associated with $\mathcal{M}_1$ and $\mathcal{M}_2$, the proof is given in [2].

Numerically, to compute $\mathcal{M}_{1\cap 2}$, the real-eigenvalues of $\mathcal{N}$ are first evaluated with a Newton algorithm. Then the eigenvectors of $\mathcal{N}$, which define $\mathcal{P}$, are computed using the algebra notions of image and kernel spaces.

REMARK 1.1 The intersection operation is not associative. Consequently, when more than two metrics are intersected, the result depends on the order of intersection. In this case, the resulting intersected metric is not anymore optimal. If, we seek for the largest ellipsoid included in the geometric intersection region of several (> 2) metrics, the John ellipsoid has to be found thanks to an optimization problem [69].

Metric interpolation

In practice, the metric field is only known discretely at mesh vertices. The definition of an interpolation procedure on metrics is therefore mandatory to be able to compute the metric at any point of the domain. For instance, the computation of the volume of an element using quadrature formula with (1.1) requires the computation of some interpolated metrics inside the considered element.
Several interpolation schemes have been proposed in [2] which are based on the simultaneous reduction. The main drawback of these approaches is that the interpolation operation is not commutative. Hence, the result depends on the order in which the operations are performed when more than two metrics are involved. Moreover, such interpolation schemes do not satisfy useful properties such as the maximum principle. Consequently, to design an interpolation scheme on these objects, one needs a consistent operational framework. We suggest to consider the log-Euclidean framework introduced in [9].

Log-Euclidean framework. We first define the notion of metric logarithm and metric exponential.
The **metric logarithm** is defined on the set of metric tensors. For metric tensor $\mathcal{M} = \mathcal{R}\,\Lambda\,{}^t\mathcal{R}$, it is given by:

$$\ln(\mathcal{M}) := \mathcal{R}\ \ln(\Lambda)\,{}^t\mathcal{R},$$

where $\ln(\Lambda) = diag(\ln(\lambda_i))$. The **matrix exponential** is defined on the set of symmetric matrices. For any symmetric matrix $\mathcal{S} = \mathcal{Q}\,\Xi\,{}^t\mathcal{Q}$, it is given by:

$$\exp(\mathcal{S}) := \mathcal{Q}\ \exp(\Xi)\,{}^t\mathcal{Q},$$

where $\exp(\Xi) = diag(\exp(\xi_i))$. We can now define the **logarithmic addition** $\oplus$ and the **logarithmic scalar multiplication** $\odot$:

$$\mathcal{M}_1 \oplus \mathcal{M}_2 := \exp(\ln(\mathcal{M}_1) + \ln(\mathcal{M}_2))$$

$$\alpha \odot \mathcal{M} := \exp(\alpha . \ln(\mathcal{M})) = \mathcal{M}^\alpha.$$

The logarithmic addition is commutative and coincides with matrix multiplication whenever the two tensors $\mathcal{M}_1$ and $\mathcal{M}_2$ commute in the matrix sense. The space of metric tensors, supplied with the logarithmic addition $\oplus$ and the logarithmic scalar multiplication $\odot$ is a vector space.

Remark 1.2 This framework allows more general computations to be carried out on metric tensors, such as statistical studying or the resolution of PDE's on metric tensors.

Metric interpolation in the log-Euclidean framework. We propose to use the linear interpolation operator derived from the log-Euclidean framework. Let $(\mathbf{x}_i)_{i=1...k}$ be a set of vertices and $(\mathcal{M}(\mathbf{x}_i))_{i=1...k}$ their associated metrics. Then, for a point $\mathbf{x}$ of the domain such that:

$$\mathbf{x} = \sum_{i=1}^{k} \alpha_i \, \mathbf{x}_i \quad \text{with} \quad \sum_{i=1}^{k} \alpha_i = 1,$$

the interpolated metric is defined by:

$$\mathcal{M}(\mathbf{x}) = \bigoplus_{i=1}^{k} \alpha_i \odot \mathcal{M}(\mathbf{x}_i) = \exp\left(\sum_{i=1}^{k} \alpha_i \ln(\mathcal{M}(\mathbf{x}_i))\right). \tag{1.2}$$

This interpolation is commutative, but its bottleneck is to perform k diagonalizations and to request the use of the logarithm and the exponential functions which are CPU consuming. However, this procedure is essential to define continuously the metric map on the entire domain. Moreover, it has been demonstrated in [9] that this interpolation preserves the maximum principle, i.e., for an edge $\mathbf{pq}$ with endpoints metrics $\mathcal{M}(\mathbf{p})$ and $\mathcal{M}(\mathbf{q})$ such that $\det(\mathcal{M}(\mathbf{p})) < \det(\mathcal{M}(\mathbf{q}))$ then we have $\det(\mathcal{M}(\mathbf{p})) < \det(\mathcal{M}(\mathbf{p} + t\,\mathbf{pq})) < \det(\mathcal{M}(\mathbf{q}))$ for all $t \in [0, 1]$.

Remark 1.3 The interpolation formulation (1.2) reduces to

$$\mathcal{M}(\mathbf{x}) = \prod_{i=1}^{k} \mathcal{M}(\mathbf{x}_i)^{\alpha_i}, \tag{1.3}$$

if all the metrics commute. Therefore, an arithmetic mean in the log-Euclidean framework could be interpreted as a geometric mean in the space of metric tensors.

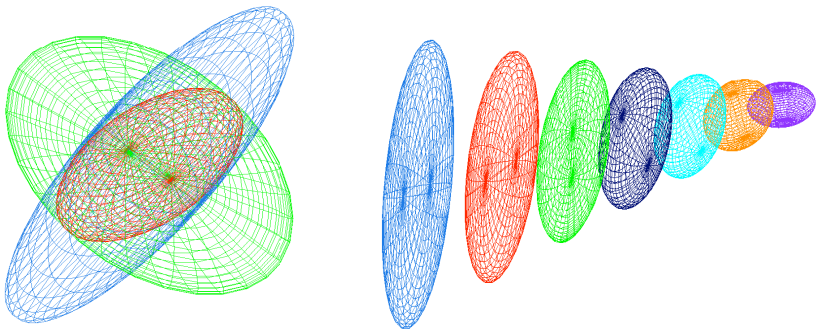

Figure 1.3: *Left, view illustrating the metric intersection procedure with the simultaneous reduction in three dimensions. In red, the resulting metric of the intersection of the blue and the green metrics. Right, metric interpolation along a segment where the endpoints metrics are the blue and the purpe ones.*

1.4 CONTINUOUS MESH FRAMEWORK

Previously, we have emphasized the role of metric tensor and Riemannian metric spaces as useful mathematical tools to prescribed sizes and directions to adaptive meshers. As a matter of fact, these differential geometry notions are really more than just a simple tool for mesh generation.
In this section, we go further in this analysis and we demonstrate that there is a comprehensive duality between discrete meshes and Riemannian metric spaces. More precisely, Riemannian metric spaces can be seen as continuous models representing meshes. This section summarizes works that have been done in [69].

To build the continuous framework, the study is first done locally for a single element of a given mesh and then generalized to the whole computational domain Ω. The notions of *unit element* and *unit mesh* with respect to a metric field play a central role in this perspective.

1.4.1 The continuous element model

We have seen in the previous sections that an arbitrary element K of positive Euclidean volume $|K| > 0$ defined by its list of edges $(\mathbf{e}_i)_i$ is **unit** with respect to a constant metric tensor $\mathcal{M}$ if the lengths of all its edges are unit in metric $\mathcal{M}$. In fact, the function *unit with respect to* defines a classes of equivalence of discrete elements.

Proposition 1.1 (Equivalence classes)
Let $\mathcal{M}$ be a metric tensor, there exists a non-empty infinite set of unit elements with respect to $\mathcal{M}$. Conversely, given an element K such that $|K| \neq 0$, there is a unique metric tensor $\mathcal{M}$ for which K is unit with respect to $\mathcal{M}$.

All the discrete representatives of a given equivalence class $\mathcal{M}$ share some common properties, which can be described using only metric tensor $\mathcal{M}$. These properties

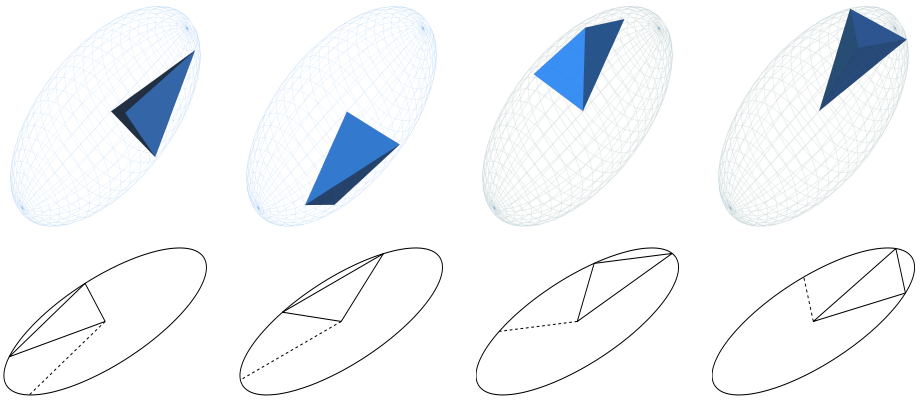

Figure 1.4: *Several unit elements with respect to a continuous element in 2D and 3D.*

connect the geometric properties of unit elements to the linear algebra properties of metric tensors. The following proposition gives some geometric invariants that hold for all unit element with respect to metric tensor. Other geometric invariants can be found in [69].

Proposition 1.2 (Geometric invariants)
Let $\mathcal{M}$ be a metric tensor and K be a unit element with respect to $\mathcal{M}$. We denote by $(\mathbf{e}_i)_{1\leqslant i\leqslant \frac{n(n+1)}{2}}$ its edges list ($n = 2$ or 3) and $|K|$ its Euclidean volume. Then the following invariants hold:

- *standard invariants:*

$$\forall(\mathbf{e}_i,\mathbf{e}_j),\quad \begin{cases} \ell_{\mathcal{M}}(\mathbf{e}_i)^2 = {}^t\mathbf{e}_i\,\mathcal{M}\,\mathbf{e}_i = 1 \\ \\ 2\,{}^t\mathbf{e}_i\mathcal{M}\mathbf{e}_j + 1 = 0 \quad if\ \ i \neq j \end{cases}$$

- *invariant related to the Euclidean volume $|K|$:*

$$|K| = \frac{\sqrt{3}}{4}\,\det(\mathcal{M}^{-\frac{1}{2}})\ \textit{in } 2D \quad \textit{and} \quad |K| = \frac{\sqrt{2}}{12}\,\det(\mathcal{M}^{-\frac{1}{2}})\ \textit{in } 3D.$$

- *invariant related to the square length of the edges for all symmetric definite positive matrix H:*

$$\begin{aligned} \sum_{i=1}^{3}\ell_H(\mathbf{e}_i)^2 &= \sum_{i=1}^{3}{}^t\mathbf{e}_i\,H\,\mathbf{e}_i = \frac{3}{2}\,trace(\mathcal{M}^{-\frac{1}{2}}\,H\,\mathcal{M}^{-\frac{1}{2}})\ \textit{in } 2D, \\ \sum_{i=1}^{6}\ell_H(\mathbf{e}_i)^2 &= \sum_{i=1}^{6}{}^t\mathbf{e}_i\,H\,\mathbf{e}_i = 2\,trace(\mathcal{M}^{-\frac{1}{2}}\,H\,\mathcal{M}^{-\frac{1}{2}})\ \textit{in } 3D. \end{aligned}$$

Proposition 1.1 highlights a duality between discrete and continuous elements. **Proposition 1.2** illustrates a duality between geometric quantities. We thus introduce the following terminology:

Definition 1.6 (Continuous element)
In the continuous mesh framework, a metric tensor $\mathcal{M}$ is called ***continuous element****. It is used to model all discrete elements that are unit for $\mathcal{M}$.*

1.4.2 The continuous mesh model

In this subsection, $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ represents a Riemannian metric space. As for the local duality, we would like to define equivalence classes of meshes, each class being represented by a single continuous object. But the main complexity is to take into account the variations of function $\mathbf{x} \mapsto \mathcal{M}(\mathbf{x})$. To simplify the analysis, $\mathbf{M}$ is first rewritten in order to distinguish local properties from global ones:

Proposition 1.3 *A Riemannian metric space $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ locally writes:*

$$\forall\, \mathbf{x} \in \Omega, \quad \mathcal{M}(\mathbf{x}) = d^{\frac{2}{n}}(\mathbf{x})\, \mathcal{R}(\mathbf{x})\, diag\left(r_1^{\frac{2}{n}}(\mathbf{x}), ..., r_n^{\frac{2}{n}}(\mathbf{x})\right)\, {}^t\mathcal{R}(\mathbf{x}),$$

where:

- *density d is equal to: $d = \left(\prod_{i=1}^{n} \lambda_i\right)^{\frac{1}{2}} = \left(\prod_{i=1}^{n} h_i\right)^{-1}$, with λ_i the eigenvalues of $\mathcal{M}$,*
- *anisotropic quotients r_i are equal to: $r_i = h_i \left(\prod_{k=1}^{n} h_k\right)^{-\frac{1}{n}}$,*
- *$\mathcal{R}$ is the eigenvectors matrix of $\mathcal{M}$ representing the orientation.*

The density d controls only the local level of accuracy of $\mathbf{M}$. Increasing or decreasing d does not change the anisotropic properties or the orientation. The anisotropy is given by the anisotropic quotients and the orientation by matrix $\mathcal{R}$.

Remark 1.4 The set of initial parameters $(h_1, ..., h_n)$ that define locally a metric is replaced by the new set of parameters $(d, r_1, ..., r_{n-1})$.

We also define the **complexity** $\mathcal{C}$ of $\mathbf{M}$:

$$\mathcal{C}(\mathbf{M}) = \int_\Omega d(\mathbf{x})\, \mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\, \mathrm{d}\mathbf{x}.$$

This real-value parameter is useful to quantify the global level of accuracy of $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. It can also be interpreted as the continuous counterpart of the number of vertices of a discrete mesh.
This quantity also leads to the definition of a sequence of embedded Riemannian spaces:

Embedded Riemannian spaces. Two Riemannian spaces, saying $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ and $\mathbf{N} = (\mathcal{N}(\mathbf{x}))_{\mathbf{x}\in\Omega}$, are embedded if a constant c exists such that:

$$\forall\, \mathbf{x} \in \Omega, \quad \mathcal{N}(\mathbf{x}) = c\, \mathcal{M}(\mathbf{x}).$$

Conversely, from $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$, we can deduce $\mathbf{N} = (\mathcal{N}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ of complexity N with the same anisotropic properties (anisotropic orientations and ratios) by considering:

$$\mathcal{N}(\mathbf{x}) = \left(\frac{N}{\mathcal{C}(\mathbf{M})}\right)^{\frac{2}{n}} \mathcal{M}(\mathbf{x}).$$

In the context of error estimation, this notion enables to perform convergence order studies with respect to an increasing complexity.

Proposition 1.3 underlines a duality between meshes and Riemannian metric spaces. In particular, this duality is locally justified by strict analogy between discrete and continuous notions: orientation vs. $\mathcal{R}$, stretching vs. r_i and size vs. d. For a mesh, we point out the duality between the number of vertices and $\mathcal{C}(\mathbf{M})$. However, as already explained in Subsection 1.3.1, the set of discrete meshes represented by $\mathbf{M}$ is more complex to describe than the class of unit elements. The problem arises from the impossibility to tesselate $\mathbb{R}^3$ uniquely with the regular elements. Consequently, the notion of unit element does not extend as well to a mesh. In order to ensure existence, the notion of quasi-unit element is devised, see Definition 1.4. This definition takes into account the continuous mesh variations. We thus introduce the following terminology:

Definition 1.7 (Continuous mesh)
In the continuous mesh framework, a ***continuous mesh*** *of a domain Ω is defined by a collection of continuous elements $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$, i.e., a Riemannian metric space. It is used to model all meshes that are unit for $\mathbf{M}$.*

1.5 CONCLUSION

In this chapter, we recalled a continuous framework to model elements and meshes. This continuous mesh framework pushes further the duality between Riemannian metric spaces and discrete meshes. This also demonstrated the well-foundedness of metric-based mesh adaptation. The last chapters will emphasize the fertility of this new concept in the context of error estimation and in the aim of seeking for the optimal mesh for a given problem.

2

Symmetric tensor decomposition

Contents

2.1 INTRODUCTION

Multilinear algebra is the algebra of higher-order tensors. Higher-order tensors can intuitively be imagined as the multidimensional equivalent of vector (first order) or matrices (second order), i.e., as "blocks" of numbers, in three or more dimensions. The entries of an N^{th}-order tensor are defined with respect to the bases chosen in N reference vector spaces. By looking for coordinate transformations that induce an interesting representation of tensor, one can define several types of multilinear decompositions; similar questions can be raised for higher-order tensors.

Tensors are objects which appear in many contexts and different applications. They have been widely used in Electrical Engineering since the nineties [88], and in particular in Antenna Array Processing [27] or Telecommunications [63, 85]. Even earlier in the seventies, tensors have been used in Chemometrics [18] or Psychometrics [59].

Higher-order tensor decomposition has proven to be useful in a number of application fields. For instance, in Arithmetic Complexity with the use of third-order tensors to represent bilinear maps [60, 87]. Another important application field is Factor Analysis. For instance, Independent Component Analysis, initially introduced for symmetric tensors whose rank did not exceed the dimension [29]. Statisticians early identified difficult problems, tackling the limits of linear algebra. The difficulty lies in the fact that such arrays may have more factors than their dimensions. Next, data are often arranged in many-way arrays and the reduction to two-way arrays sometimes results in a loss of information. Lastly, the solution of some problems generally requires the use of High-Order Statistics (HOS) which are intrinsically tensors objects (McCullagh 1987). Now, it has become possible to estimate more factors than the dimension [57].

In this chapter, we study the decomposition of symmetric tensors into a minimal linear combination of rank-one terms. Higher-order tensors will play a central role for higher-order interpolation in mesh adaptation as rank-2 tensors in the linear case.

The decomposition of a tensor was first introduced and studied by Frank L. Hitchcook in 1927, and then was discovered in 1970's by psychometricians. Bergman [14] and Harshman [51] were the first to notice that the concept of rank was difficult to extend from matrices to higher-order tensors. Harshman [50] and Carrol [25] developed the first *Canonical Polyadic Decomposition* algorithms of a third-order tensor and its extension to higher-order, later referred to as *(PARAFAC)* model or *CANDECOMP* model. Several years later, Kruskal [60] conducted a detailed analysis of uniqueness, and related several definitions of rank.

Here, we consider two methods of decomposition of any symmetric tensor of any degree and any dimension:

The first one is a rank determinant problem which extends the Singular Value De-

composition (SVD) problem for symmetric matrices. It is based on Sylvester's theorem for the 2D case and its extension to higher dimensions. The symmetric tensor decomposition algorithms have been developed by Comon, Mourrain et al. [17, 33] so that any homogeneous polynomial of any variables and arbitrary degree associated with a symmetric tensor of arbitrary order and dimension can be decomposed as a sum of powers of linear terms. The number of powers in the linear form can be generic, i.e., it corresponds to the minimal number of terms that is required in general. As it can be non-generic, i.e., the number of powers can be larger than in the generic case.

The second one is the *Multi-way Parallel Factor* (PARAFAC) model fitting by the Alternating Least Square (ALS) algorithm, a canonical decomposition method based on a functional minimization problem. The PARAFAC/ALS decomposition is an iterative numerical method proposed to decompose third-order tensors (later extended to higher-order tensors) in higher dimensions as a sum of rank-1 tensors. It has most often been applied in psychometrics, chemometrics and the signal processing area [62, 84]. This technique requests the rank to be much smaller than the generic one. But of course, we can be confronted to tensors of rank much larger than the generic one. In this case, this iterative numerical method encounters difficulties to compute the corresponding tensor decomposition and suffers from a lack of a guarantee of global convergence.

2.2 INTRODUCTION TO MULTILINEAR ALGEBRA

This section contains the basic materials on multilinear algebra. We will start by giving a proper definition of higher-order tensors. Our next concern is the development of tools to work with higher-order tensors. We introduce some basic matrix and tensor operations, we establish a convention to represent higher-order tensors in terms of matrices; such a format is required to express tensor techniques in terms of matrix tools and software. These tools will be useful to introduce homogeneous polynomials and make the connexion with symmetric tensors. We also present a multilinear generalization of Singular Value Decomposition (SVD). All these tools will be used in the next section and chapter.

2.2.1 Higher-order tensors

Definition 2.1 *Let $(\mathbb{V}^{(\ell)})_{1\leqslant\ell\leqslant N}$ be N Euclidean vector spaces with finite dimensions $(I_\ell)_{1\leqslant\ell\leqslant N}$. An element of the tensor vector space $\mathbb{V}^{(1)}\otimes\mathbb{V}^{(2)}\otimes\ ...\ \otimes\mathbb{V}^{(N)}$, where $\otimes$ denotes the outer (tensor) product (see Definition 2.2 below), is called a* $\mathbf{N^{th}}$***-order*** *or* ***higher-order tensor****.*
Let us choose $(\mathbf{e}_{i_\ell}^{(\ell)})_{1\leqslant i_\ell\leqslant I_\ell}$ a basis in each of the N vector spaces $\mathbb{V}^{(\ell)}$. Then any Nth-order tensor $\mathbf{X}$ of that vector space of dimensions $\prod_{\ell=1}^{N} I_\ell$ has coordinates $X_{i_1 i_2 ... i_N}$ and

is defined by the relation:

$$\mathbf{X} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} ... \sum_{i_N=1}^{I_N} X_{i_1 i_2 ... i_N} \, \mathbf{e}_{i_1}^{(1)} \otimes \mathbf{e}_{i_2}^{(2)} \otimes \ ... \ \otimes \mathbf{e}_{i_N}^{(N)}. \tag{2.1}$$

For $\mathbb{V}^{(\ell)} = \mathbb{R}^{I_\ell}$, $\mathbf{X}$ *is a real-valued* $(I_1 \times I_2 \times ... \times I_N)$*-tensor and for* $\mathbb{V}^{(\ell)} = \mathbb{C}^{I_\ell}$, $\mathbf{X}$ *is a complex-valued* $(I_1 \times I_2 \times ... \times I_N)$*-tensor.*

A tensor of order N is also called a N-way array; i.e., it enjoys the multi-linearity property after a change of coordinate system. For instance, consider a third-order tensor $\mathbf{X}$ with entries X_{ijk}, and a change of coordinates defined by three square invertible matrices, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. Then, in the new coordinates system, the tensor $\mathbf{X}^{bis}$ can be written as a function of tensor $\mathbf{X}$ as:

$$X_{ijk}^{bis} = \sum_{abc} A_{ia} \, B_{jb} \, C_{kc} \, X_{abc}. \tag{2.2}$$

2.2.2 Review of matrix and tensor standard operations

We define some vector and matrix products like the Outer product, the Kronecker product and the Khatri-Rao product that will be used in the sequel of the chapter.

Definition 2.2 *The Outer product of vectors* $\mathbf{a} = (a_i)_i \in \mathbb{C}^I$ *and* $\mathbf{b} = (b_j)_j \in \mathbb{C}^J$ *is denoted by* $\mathbf{a} \otimes \mathbf{b}$ *and its* $I \times J$ *result is a matrix defined by:*

$$\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\,^{\mathbf{t}}\mathbf{b} = \begin{pmatrix} a_1 b_1 & a_1 b_2 & ... & a_1 b_J \\ a_2 b_1 & a_2 b_2 & ... & a_2 b_J \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ a_I b_1 & a_I b_2 & ... & a_I b_J \end{pmatrix}.$$

Definition 2.3 *The Kronecker product of matrices* $\mathbf{A} = (a_{ij})_{i,j} \in \mathbb{C}^{I \times J}$ *and* $\mathbf{B} = (b_{k\ell})_{k,\ell} \in \mathbb{C}^{K \times L}$ *is denoted by* $\mathbf{A} \otimes \mathbf{B}$ *and the* $IK \times JL$ *result is a matrix defined by:*

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & ... & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & ... & a_{2J}\mathbf{B} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & ... & a_{IJ}\mathbf{B} \end{pmatrix}.$$

Note that the Kronecker product and the Outer product are denoted in a similar manner, which might be confusing. In fact, this usual practice has some reasons: $\mathbf{A} \otimes \mathbf{B}$ *is the array of coordinates of the Outer product of the two associated linear operators, in some canonical basis.*

Definition 2.4 *The Khatri-Rao product of matrices* $\mathbf{A} = (a_{ik})_{i,k} \in \mathbb{C}^{I\times K}$ *and* $\mathbf{B} = (b_{jk})_{j,k} \in \mathbb{C}^{J\times K}$ *is denoted by* $\mathbf{A} \odot \mathbf{B}$ *and its* $IJ \times K$ *result is a matrix defined by:*

$$\mathbf{A} \odot \mathbf{B} = \left(\begin{array}{cccc} a_{:1} \otimes b_{:1} & a_{:2} \otimes b_{:2} & \ldots & a_{:K} \otimes b_{:K} \end{array} \right).$$

The Khatri-Rao product is nothing else but the column-wise Kronecker product.

We now define the "*matrix unfoldings*" of a given tensor, i.e., the matrix representations of that tensor in which all the column (row, ...) vectors are stacked one after the other. To avoid confusion, we will stick to one particular ordering of the column (row, ...). An Nth-order tensor $\mathbf{X}$ admits N different matrix unfoldings. These matrix unfoldings are also called "*matrix flattenings*", or equivalently "*modes*".

Definition 2.5 (Matrix unfolding)
Assume an Nth-order tensor $\mathbf{X} \in \mathbb{C}^{I_1\times I_2\times \ldots \times I_N}$. *For a particular choice of* $\overline{N} = \{1, 2, ..., N\}$, *the matrix unfolding*

$$\mathbf{X}_{I_{P(1)}I_{P(2)}\ldots I_{P(\overline{N})}\times I_{P(\overline{N}+1)}I_{P(\overline{N}+2)}\ldots I_{P(\overline{N})}} \in \mathbb{C}^{I_{P(1)}I_{P(2)}\ldots I_{P(\overline{N})}\times I_{P(\overline{N}+1)}I_{P(\overline{N}+2)}\ldots I_{P(N)}}$$

of the tensor $\mathbf{X}$, *linked with a permutation* $P(1, 2, ..., N)$ *of* $(1, 2, ..., N)$ *contains the element* $X_{i_1 i_2 \ldots i_N}$ *at the position with row index*
$(i_{P(1)} - 1)I_{P(2)}I_{P(3)} \ldots I_{P(\overline{N})} + (i_{P(2)} - 1)I_{P(3)}I_{P(4)} \ldots I_{P(\overline{N})} + \ldots + i_{P(\overline{N})}$
and column index
$(i_{P(\overline{N}+1)} - 1)I_{P(\overline{N}+2)}I_{P(\overline{N}+3)} \ldots I_{P(N)} + (i_{P(\overline{N}+2)} - 1)I_{P(\overline{N}+3)}I_{P(\overline{N}+4)} \ldots I_{P(N)} + \ldots + i_{P(N)}$.

Notice that the definitions of the matrix unfoldings involve the tensor dimensions $I_1, I_2, ..., I_N$ *in a cyclic way.*

A standard matrix unfolding, corresponding to $\overline{N} = 1$ *and permuted indices* $(n, 1, ..., n - 1, n + 1, ..., N)$ *will be briefly represented by* $\mathbf{X}_{(n)}$.

To show how it really works, let us take a $I \times J \times K$ third-order complex or real-valued array $\mathbf{X}$. For k fixed in the third mode, we have a $I \times J$ matrix that can be denoted as $\mathbf{X}_{::k}$. The collection of these K such matrices can be arranged in a $K \times IJ$ block matrix :

$$\mathbf{X}_{K\times IJ} = \left[\begin{array}{c} \mathbf{X}_{::1} \\ \vdots \\ \mathbf{X}_{::k} \\ \vdots \\ \mathbf{X}_{::K} \end{array} \right].$$

The three sections of the tensor $\mathbf{X}$: $\mathbf{X}_{i::}$, $\mathbf{X}_{:j:}$ and $\mathbf{X}_{::k}$ are respectively called *horizontal, lateral and frontal slices.*
The three matrix representations or matrix unfoldings of $\mathbf{X}$ are:
- $\mathbf{X}_{I\times JK}$: obtained by stacking all $J \times K$ up-down slices of $\mathbf{X}$ one above each other;

- $\mathbf{X}_{J\times KI}$: obtained by stacking all $K \times I$ left-right slices of $\mathbf{X}$ one above each other;
- $\mathbf{X}_{K\times IJ}$: obtained by stacking all $I \times J$ front-bottom slices of $\mathbf{X}$ one above each other.

The concept of matrix unfolding will be easy to understand using an example.

Example 2.1
Let the frontal slices of $\mathbf{X} \in \mathbb{R}^{3\times3\times2}$ be

$$\mathbf{X}_{::1} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}, \qquad \mathbf{X}_{::2} = \begin{bmatrix} 10 & 13 & 16 \\ 11 & 14 & 17 \\ 12 & 15 & 18 \end{bmatrix}.$$

Then, the three unfolding matrices of $\mathbf{X}$ are:

$$\mathbf{X}_{(1)} = \mathbf{X}_{I\times JK} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 \\ 2 & 5 & 8 & 11 & 14 & 17 \\ 3 & 6 & 9 & 12 & 15 & 18 \end{bmatrix},$$

$$\mathbf{X}_{(2)} = \mathbf{X}_{J\times KI} = \begin{bmatrix} 1 & 2 & 3 & 10 & 11 & 12 \\ 4 & 5 & 6 & 13 & 14 & 15 \\ 7 & 8 & 9 & 16 & 17 & 18 \end{bmatrix},$$

$$\mathbf{X}_{(3)} = \mathbf{X}_{K\times IJ} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \end{bmatrix}.$$

We now define the n-mode product, i.e., the multiplication of a higher-order tensor by a matrix and the Frobenius norm.

Definition 2.6 *The n-mode product of a tensor $\mathbf{X} \in \mathbb{C}^{I_1\times I_2\times\ldots\times I_N}$ by a matrix $\mathbf{A} \in \mathbb{C}^{J_n\times I_n}$, denoted by $\mathbf{X} \times_n \mathbf{A}$, is an $(I_1 \times I_2 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots \times I_N)$-tensor of which the entries are given by*

$$(\mathbf{X} \times_n \mathbf{A})_{i_1 i_2 \ldots j_n \ldots i_N} = \sum_{i_n=1}^{I_n} X_{i_1 i_2 \ldots i_n \ldots i_N}\, a_{j_n i_n},$$

for all index values.

Definition 2.7 *The Frobenius norm of a Nth-order tensor $\mathbf{X} = X_{i_1 i_2 \ldots i_N} \in \mathbb{C}^{I_1\times I_2\times\ldots\times I_N}$ is the square root of the sum of the squares of the absolute values of all its elements, i.e.,*

$$||\mathbf{X}||_F = \sqrt{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\ldots\sum_{i_N=1}^{I_N} |X_{i_1 i_2 \ldots i_N}|^2}.$$

The following part is devoted to the definition and properties of the rank of a higher-order tensor. When rank properties are concerned, there are major differences between matrices and higher-order tensors. As we will explain in subsection 2.2.3, these differences directly affect the way in which tensorial decompositions differ from matrix decomposition. We use the following definition:

Definition 2.8 *An Nth-order tensor* $\mathbf{X}$ *has rank* 1 *when it equals the outer product of* N *vectors* $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, ..., \mathbf{U}^{(N)}$ *with* $\mathbf{U}^{(\ell)} = (U^{(\ell)}_{i_\ell})_{i_\ell}$*:*

$$\mathbf{X} = \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \otimes ... \otimes \mathbf{U}^{(N)}, \tag{2.3}$$

i.e.,

$$X_{i_1 i_2 ... i_N} = U^{(1)}_{i_1} U^{(2)}_{i_2} ... U^{(N)}_{i_N}.$$

Definition 2.9 *The rank of an arbitrary Nth-order tensor* $\mathbf{X}$*, represented by* $R = rank(\mathbf{X})$*, is the* ***minimal*** *number of rank-1 tensors such that the following equality holds true:*

$$\mathbf{X} = \sum_{p=1}^{R} U^{(1)}_p \otimes U^{(2)}_p \otimes ... \otimes U^{(N)}_p. \tag{2.4}$$

Tensor rank R *always exists and is well defined.*

Definition 2.10 *The n-mode vectors of an Nth-order tensor* $\mathbf{X} \in \mathbb{C}^{I_1 \times I_2 \times ... \times I_N}$ *are the* I_n*-dimensional vectors obtained from* $\mathbf{X}$ *by varying the index* i_n *and keeping the other indices fixed, i.e., they are the column vectors of the matrix unfolding* $\mathbf{X}_{(n)}$*.*

The n-rank of the tensor $\mathbf{X}$*, represented by* $R_n = rank_n(\mathbf{X})$*, is the dimension of the vector space generated by the n-mode vectors of* $\mathbf{X}$ *and*

$$rank_n(\mathbf{X}) = rank(\mathbf{X}_{(n)}).$$

This definition generalizes the notion of "column (row) vector" and "column (row) rank" of matrices to Nth-order tensors.

2.2.3 Higher-Order Singular Value Decomposition

In this subsection, we present a multilinear generalization of the Singular Value Decomposition (SVD) for N^{th} order tensors, called *Higher-Order Singular Value Decomposition* (HOSVD). The HOSVD will play an important role in the next section. Indeed, the HOSVD will be used for dimensionality reduction in the Multi-way Parallel Factor (PARAFAC) model.
For convenience, we first repeat the model for matrices using a similar notation:

Theorem 2.1 (Matrix SVD)
Every complex $(I_1 \times I_2)$*-matrix* $\mathbf{A}$ *can be written as the product*

$$\mathbf{A} = \mathbf{U}^{(1)} \,.\, \mathbf{S} \,.\, \mathbf{V}^{(2)^H} = \mathbf{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{V}^{(2)^*} = \mathbf{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)},$$

in which:

- $\mathbf{U}^{(1)} = \left[U_1^{(1)} U_2^{(1)} \ldots U_{I_1}^{(1)}\right]$ *is a unitary* $(I_1 \times I_1)$*-matrix,*
- $\mathbf{U}^{(2)} = \left[U_1^{(2)} U_2^{(2)} \ldots U_{I_2}^{(2)}\right]$ $(= \mathbf{V}^{(2)^*})$ *is a unitary* $(I_2 \times I_2)$*-matrix,*
- $\mathbf{S}$ *is an* $(I_1 \times I_2)$*-matrix with the properties of*

  - pseudodiagonality:

$$\mathbf{S} = diag(\sigma_1, \sigma_2, \ldots, \sigma_{\min(I_1, I_2)}),$$

  - ordering:

$$\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_{\min(I_1, I_2)} \geqslant 0.$$

The σ_i *are singular values of* $\mathbf{A}$ *and the vectors* $U_i^{(1)}$ *and* $U_i^{(2)}$ *are respectively an* i^{th} *left and an* i^{th} *right singular vector. The symbol* $*$ *denotes the complex conjugation.*

Now we state the following theorem:

Theorem 2.2 (*$\mathbf{N^{th}}$*-order SVD)
Every complex $(I_1 \times I_2 \times \ldots \times I_N)$*-tensor* $\mathbf{X}$ *can be written as the product*

$$\mathbf{X} = \mathbf{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \ldots \times_N \mathbf{U}^{(N)}, \tag{2.5}$$

in which:

- $\mathbf{U}^{(n)} = \left[U_1^{(n)} U_2^{(n)} \ldots U_{I_n}^{(n)}\right]$ *is a unitary* $(I_n \times I_n)$*-matrix,*
- $\mathbf{S}$ *is a complex* $(I_1 \times I_2 \times \ldots \times I_N)$*-tensor of which the subtensors* $\mathbf{S}_{i_n=\alpha}$*, obtained by fixing the nth index to* α*, have the properties of*

  - all-orthogonality: two subtensors $\mathbf{S}_{i_n=\alpha}$ *and* $\mathbf{S}_{i_n=\beta}$ *are orthogonal for all possible values of* n*,* α *and* β *subject to* $\alpha \neq \beta$*:*

$$(\mathbf{S}_{i_n=\alpha}, \mathbf{S}_{i_n=\beta}) = 0 \quad when \quad \alpha \neq \beta,$$

  - ordering:

$$\|\mathbf{S}_{i_n=1}\|_F \geqslant \|\mathbf{S}_{i_n=2}\|_F \geqslant \ldots \geqslant \|\mathbf{S}_{i_n=I_n}\|_F \geqslant 0$$

for all possible values of n.

The number $(\mathbf{X}, \mathbf{Y})$ denotes the tensor scalar product of two tensors $\mathbf{X}$ and $\mathbf{Y}$. It is just the extension of the classical scalar product of two vectors. The Frobenius-norm $\|\mathbf{S}_{i_n=i}\|_F$, symbolized by $\sigma_i^{(n)}$, are n-mode singular values of $\mathbf{X}$ and the vector $U_i^{(n)}$ is an ith n-mode singular vector.

2.2.4 Symmetric tensors and homogeneous polynomials

For the sake of simplicity and to avoid confusion, in the sequel, the order of a tensor will be denoted by k and the dimension of a tensor will be denoted by n.

Definition 2.11 *According to Definition 2.1, a tensor $\mathbf{X}$ of dimension n and order k is an object defined in a n-dimensional coordinate system by a table with k indices, $\{X_{i_1 i_2 \ldots i_k}\}_{1 \leqslant i_\ell \leqslant n}$, that follows a particular transformation formula if the coordinate system is changed.*
Then, a k^{th}-order tensor $\mathbf{X}$ of dimension n is symmetric if $\sigma(\mathbf{X}) = \mathbf{X}$, i.e., $X_{\sigma(i_1 i_2 \ldots i_k)} = X_{i_1 i_2 \ldots i_k}$ for all permutation σ.

Definition 2.12 *Let $\mathbf{X}$ be a k^{th}-order tensor of dimension n. A homogeneous polynomial of degree k in n variables can be associated to $\mathbf{X}$ by the following expression:*

$$p(x_{i_1}, x_{i_2}, ..., x_{i_k}) = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} ... \sum_{i_k=1}^{n} X_{i_1 i_2 \ldots i_k} \, x_{i_1} x_{i_2} ... x_{i_k}. \tag{2.6}$$

In Expression (2.6), it is clear that because of the symmetry of $\mathbf{X}$, some terms appears several times. There is another way of writing polynomials by resorting a standard compact notation, widely used in invariant theory.

Let $\mathbf{i} \in \mathbb{N}^n$ be a vector of n indices. The length of $\mathbf{i}$ is defined as $|\mathbf{i}| = \sum_{\ell=1}^{n} i_\ell$. By convention, if $\mathbf{x} \in \mathbb{C}^n$, $\mathbf{x}^\mathbf{i}$ denoted the product $\prod_{\ell=1}^{n} x_\ell^{i_\ell}$ and $(\mathbf{i})! = \prod_{\ell=1}^{n} (i_\ell!)$. Lastly, $c(\mathbf{i})$ denotes the multinomial coefficient, namely $c(\mathbf{i}) = \frac{|\mathbf{i}|!}{(\mathbf{i})!}$.

With this notation, any homogeneous polynomial can be written as:

$$p(\mathbf{x}) = \sum_{|\mathbf{i}|=k} c(\mathbf{i}) \, c_\mathbf{i} \, \mathbf{x}^\mathbf{i}, \tag{2.7}$$

$c_\mathbf{i}$ are the coefficients characterizing polynomial $p(.)$ and each one are associated with one entry of the corresponding symmetric tensor $\mathbf{X}$.

An *algebraic form* or simply *form* is another name for a homogeneous polynomial.

Example 2.2
A homogeneous polynomial of degree k in two variables is called a *binary form* and is defined by:

$$p(x,y) = \sum_{i=0}^{k} \binom{k}{i} c_i \, x^i y^{k-i},$$

where $\binom{k}{i} = \frac{k!}{i!(k-i)!}$ is the binomial coefficient.

We present the polarization of algebraic forms. It is a technique for expressing a homogeneous polynomial or algebraic form in a simpler fashion adjoining more variables. It produces a multilinear form from which the original polynomial can be recovered by evaluating along a certain diagonal. From this multilinear form, we deduce the corresponding symmetric tensor.

Definition 2.13 *Let $f(\mathbf{x})$ be a polynomial in n variables $\mathbf{x} = (x_1, x_2, ..., x_n)$. Suppose that f is homogeneous of degree k. Let $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, ... , $\mathbf{x}^{(k)}$ be a collection of indeterminates with $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_n^{(i)})$.*

The polar form of f is a polynomial $\mathbf{F}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(k)})$ which is linear in each indeterminate $\mathbf{x}^{(i)}$ (i.e., $\mathbf{F}$ is multilinear), symmetric in $\mathbf{x}^{(i)}$ and such that $\mathbf{F}(\mathbf{x}, \mathbf{x}, ..., \mathbf{x}) = f(\mathbf{x})$. It is given by the following construction:

$$\mathbf{F}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(k)}) = \frac{1}{k!} \frac{\partial}{\partial \lambda_1} ... \frac{\partial}{\partial \lambda_k} f(\lambda_1 \mathbf{x}^{(1)} + ... + \lambda_k \mathbf{x}^{(k)})_{|\lambda_i = 0}. \tag{2.8}$$

Example 2.3
A quadratic form. Let $f(x,y) = x^2 + 3xy + 2y^2$. The polarization of f is the function in $\mathbf{x}^{(1)} = (x^{(1)}, y^{(1)})$ (rows) and $\mathbf{x}^{(2)} = (x^{(2)}, y^{(2)})$ (columns) given by:

$$\mathbf{F}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = x^{(1)}x^{(2)} + \frac{3}{2} y^{(1)}x^{(2)} + \frac{3}{2} x^{(1)}y^{(2)} + 2\, y^{(1)}y^{(2)}.$$

Using a matrix form, we can rewrite $\mathbf{F}$ as follow:

$$\mathbf{F} = \begin{bmatrix} 1 & \frac{3}{2} \\ \frac{3}{2} & 2 \end{bmatrix}.$$

A cubic form. Let $f(x,y) = x^3 + 2xy^2$. The polarization of f is the function in $\mathbf{x}^{(1)} = (x^{(1)}, y^{(1)})$ (rows), $\mathbf{x}^{(2)} = (x^{(2)}, y^{(2)})$ (columns) and $\mathbf{x}^{(3)} = (x^{(3)}, y^{(3)})$ (slices) given by:

$$\mathbf{F}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}) = x^{(1)}x^{(2)}x^{(3)} + \frac{2}{3} x^{(1)}y^{(2)}y^{(3)} + \frac{2}{3} y^{(1)}y^{(2)}x^{(3)} + \frac{2}{3} y^{(1)}x^{(2)}y^{(3)}.$$

$\mathbf{F}$ is a third-order symmetric tensor that we can rewrite using one of his matrix representations or matrix unfolding (see Definition 2.5):

$$\mathbf{F}_{(1)} = \left[\begin{array}{cc|cc} 1 & 0 & 0 & \frac{2}{3} \\ 0 & \frac{2}{3} & \frac{2}{3} & 0 \end{array}\right].$$

We give the general formula of the polarization of a homogeneous polynomial of degree 3 in two and three variables, respectively.

- **Two-dimensional case**.

Consider the following homogeneous polynomial of degree 3 in two variables:

$$f(x,y) = a_1x^3 + a_2x^2y + a_3xy^2 + a_4y^3.$$

The polarization of f is the function in $\mathbf{x}^{(1)} = (x^{(1)}, y^{(1)})$, $\mathbf{x}^{(2)} = (x^{(2)}, y^{(2)})$ and $\mathbf{x}^{(3)} = (x^{(3)}, y^{(3)})$ given by:

$$\begin{aligned}\mathbf{F}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}) &= a_1\, x^{(1)}x^{(2)}x^{(3)} + \frac{a_2}{3}y^{(1)}x^{(2)}x^{(3)} + \frac{a_2}{3}x^{(1)}x^{(2)}y^{(3)} + \frac{a_2}{3}x^{(1)}y^{(2)}x^{(3)} \\ &+ \frac{a_3}{3}x^{(1)}y^{(2)}y^{(3)} + \frac{a_3}{3}y^{(1)}y^{(2)}x^{(3)} + \frac{a_3}{3}y^{(1)}x^{(2)}y^{(3)} + a_4\, y^{(1)}y^{(2)}y^{(3)}.\end{aligned}$$

Thus, we can rewrite $\mathbf{F}$ using the first matrix unfolding of f given by:

$$\mathbf{F}_{(1)} = \frac{1}{6}\left[\begin{array}{cc|cc} 6a_1 & 2a_2 & 2a_2 & 2a_3 \\ 2a_2 & 2a_3 & 2a_3 & 6a_4 \end{array}\right].$$

- **Three-dimensional case**.

Consider the following homogeneous polynomial of degree 3 in three variables:

$$\begin{aligned}f(x,y,z) &= a_1x^3 + a_2x^2y + a_3xy^2 + a_4y^3 + a_5x^2z + a_6xz^2 + a_7z^3 \\ &+ a_8y^2z + a_9yz^2 + a_{10}xyz.\end{aligned}$$

Using the same approach as the two-dimensional case, we first give the polarization $\mathbf{F}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)})$ of f, with $\mathbf{x}^{(i)} = (x^{(i)}, y^{(i)}, z^{(i)})$, $i = 1, 2, 3$. Then, we rewrite it under a matrix form. To simplify the calculations, we directly give the result.

The first matrix unfolding of f is given by:

$$\mathbf{F}_{(1)} = \frac{1}{6}\left[\begin{array}{ccc|ccc|ccc} 6a_1 & 2a_2 & 2a_5 & 2a_2 & 2a_3 & a_{10} & 2a_5 & a_{10} & 2a_6 \\ 2a_2 & 2a_3 & a_{10} & 2a_3 & 6a_4 & 2a_8 & a_{10} & 2a_8 & 2a_9 \\ 2a_5 & a_{10} & 2a_6 & a_{10} & 2a_8 & 2a_9 & 2a_6 & 2a_9 & 6a_7 \end{array}\right].$$

2.3 SYMMETRIC TENSOR DECOMPOSITION ALGORITHMS

In this section, we present two methods of *Canonical Polyadic Decomposition*: the algorithm of Sylvester and the Multi-way PARAFAC model to decompose symmetric tensors of arbitrary dimension and order into a sum of rank-1 terms. Many authors adopted the "CP" acronym, which stands for "Canonical Polyadic", or the "CANDECOMP".

2.3.1 Sylvester's algorithm: the binary form decomposition

As already pointed out earlier, a rank-1 tensor is associated with a linear form raised to k^{th} powers. In terms of polynomials, the CP decomposition can thus be rephrased: how can one decompose a quantic into a sum of k^{th} powers of linear forms [33]? It is this topic that addresses Sylvester's theorem, restricted however to the binary case (i.e., two variables).

We first recall one important definition.

Definition 2.14 *Given a homogeneous polynomial p of degree k in n variables, the width (also called tensor rank) of p refers to the minimal number of forms, r, necessary to write p as a sum of k^{th} powers of linear forms. For a generic homogeneous polynomial, the width is denoted by $g(n, k)$.*
A generic rank refers to the minimal number of forms needed for a dense set of symmetric tensors with arbitrary dimension and order.

Generic case. It has recently been shown by Reznick [82] that for all homogeneous polynomial p in n variables of degree k,

$$g(n, k) = r \leqslant \binom{n+k-2}{k-1}. \tag{2.9}$$

But there is no general expression that gives the exact value of $g(n, k)$. To be accurate, it is necessary to study each case separately.
The following table summarizes known values of $g(n, k)$ in the generic case. As we can notice in **Table 2.1**, the rank can exceed the dimension, which is not true for matrices.

$k\backslash n$	2	3	4	5	6	7
2	2	3	4	5	6	7
3	2	4	5	8	10	12
4	3	6	10	15	22	30

Table 2.1: Generic rank r of symmetric tensors (homogeneous polynomials) as a function of the dimension n and the order k.

Non-generic case. Contrary to matrices (i.e., second-order tensor), the generic rank is not always maximal. In other word, the rank can exceed the generic value. Unfortunately, the maximal achievable rank is not known for all pairs (P, n) with P the number of sources in the decomposition and n the dimension. We illustrate this fact. For instance, for $n = 2$ and $k = 3$, the maximal rank is 3. The polynomial x^2y can be written as:

$$6\,x^2y = (x + y)^3 + (-x + y)^3 - 2y^3.$$

As we can see, the rank exceeds its generic value 2 (indeed, refer to **Table 2.1**, $k = 3$ and $n = 2$, thus rank $= 2$) but the Reznick bound (2.9) is reached $r_{max} = 3$.

Now, we recall Sylvester's theorem before presenting the resulting algorithm.

Theorem 2.3 Sylvester (1886)
A binary quantic $p(x, y) = \sum_{i=0}^{k} \binom{k}{i} c_i \, x^i y^{k-i}$ *can be written as a sum of* k^{th} *powers of* r *distinct linear forms in* $\mathbb{C}$*:*

$$p(x, y) = \sum_{j=1}^{r} \lambda_j \, (\alpha_j \, x + \beta_j \, y)^k, \tag{2.10}$$

if and only if

- *there exists a vector* $\mathbf{q} \in \mathbb{C}^{r+1}$*, with components* q_ℓ*, such that :*

$$M.\mathbf{q} = 0, \tag{2.11}$$

with

$$M = \begin{bmatrix} c_0 & c_1 & \dots & c_r \\ c_1 & c_2 & \dots & c_{r+1} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ c_{k-r} & c_{k-r+1} & \dots & c_k \end{bmatrix},$$

a catalecticant or hankel matrix (see ***Appendix A.4****) which elements are the coefficients* c_i *of the binary quantic* p*.*

- *the polynomial* $q(x, y) = \sum_{\ell=0}^{r} q_\ell \, x^\ell \, y^{r-\ell}$ *admits* r *distincts roots, i.e., it can be written as:*

$$q(\mathbf{x}) = \prod_{j=1}^{r} (\beta_j^* \, x - \alpha_j^* \, y),$$

where r *is the decomposition rank, i.e., the minimal number of linear terms such that Equality (2.10) holds true.*

The proof of this theorem is constructive [31, 33] and yields to Sylvester's algorithm [17, 33]. This theorem not only proves the existence of the r forms, but also gives a mean to compute them.

Algorithm 1: BINARY FORM DECOMPOSITION

Input: *A binary form* $p(x,y) = \sum_{i=0}^{k} \binom{k}{i} c_i \, x^i y^{k-i}$ *of degree* k.

Output: *A decomposition of* p *as* $p(x,y) = \sum_{j=1}^{r} \lambda_j \, \ell_j(x,y)^k$ *with* r *minimal.*

(1) Initialize $r = 0$.
(2) Increment $r \leftarrow r + 1$.
(3) If the column rank of $H[r]$ *is full, then go to step* (2)
(4) Else compute a basis $\{\ell_1, ..., \ell_i\}$ *of the right kernel of* $H[r]$.
(5) Specialization:

- *Take a generic vector* $\mathbf{q}$ *in the kernel, e.g.* $\mathbf{q} = \sum_i \mu_i \, \ell_i$.
- *Compute the roots of the associated polynomial* $q(x,y) = \sum_{i=0}^{r} q_i \, x^i \, y^{r-i}$.
- *If the roots are not distinct in* $\mathbb{C}^2$, *try another specialization. If distinct roots cannot be obtained, go to step* (2).
- *Else if* $q(x,y)$ *admits* r *distinct roots* (α_j, β_j) *then compute coefficients* λ_j, $1 \leqslant j \leqslant r$, *by solving the linear system below, where* a_i *denotes* $\binom{k}{i} c_i$

$$\begin{bmatrix} \alpha_1^k & ... & \alpha_r^k \\ \alpha_1^{k-1}\beta_1 & ... & \alpha_r^{k-1}\beta_r \\ . & . & . \\ . & . & . \\ . & . & . \\ \beta_1^k & ... & \beta_r^k \end{bmatrix} \lambda = \begin{bmatrix} a_0 \\ a_1 \\ . \\ . \\ . \\ a_k \end{bmatrix}.$$

(6) The decomposition is $p(x,y) = \sum_{j=1}^{r} \lambda_j \, (\alpha_j \, x + \beta_j \, y)^k$.

The associated Matlab code of **Algorithm 1** developed by Comon and Mourrain is given in **Appendix B.1**.

We give two examples to illustrate the previous algorithm and show how it really works.

Example 2.4
We apply Sylvester's algorithm to the third-order polynomial:

$$p(x,y) = 39x^3 + 102x^2y + 48xy^2 + 24y^3.$$

For $r = 1$, we have the following Hankel matrix:

$$M = \begin{bmatrix} c_0 & c_1 \\ c_1 & c_2 \\ c_2 & c_3 \end{bmatrix} = \begin{bmatrix} 39 & 34 \\ 34 & 16 \\ 16 & 24 \end{bmatrix}.$$

This matrix is full column rank, i.e., each of the columns of the matrix are linearly independent (for a square or non-square matrix). Therefore, we build the Hankel matrix for $r = 2$:

$$M = \begin{bmatrix} c_0 & c_1 & c_2 \\ c_1 & c_2 & c_3 \end{bmatrix} = \begin{bmatrix} 39 & 34 & 16 \\ 34 & 16 & 24 \end{bmatrix}.$$

This matrix is not full column rank. Its rank is equal to 2, therefore we compute a basis of the kernel. To do so, we use the singular value decomposition and we get the following decomposition of the matrix:

$$M = U\Sigma V^*,$$

where $rank(\Sigma) = 2$ and we know that $Ker(M)$ is the third column of V: $\mathbf{v} = [-0.6465, 0.4526, -0.6142]$.
We compute the roots of the univariate polynomial of degree 2, $g(x) = v_0x^2 + v_1x + v_2$. We find $x = (1.3856, -0.6856)$. Then, the roots of $q(x, y) = \sum_{i=0}^{2} v_i\, x^i y^{2-i}$ are $(\alpha_1, \beta_1) = (1.3856, 1)$ and $(\alpha_2, \beta_2) = (-0.6856, 1)$. Lastly, we compute λ_1 and λ_2 by equating coefficients in the same monomials and we get the following final decomposition:

$$p(x, y) = 15.6693(1.3856x + y)^3 + 8.3307(-0.6856x + y)^3.$$

Example 2.5
Consider the following fourth-order polynomial:

$$p(x, y) = 17x^4 + 48x^3y + 120x^2y^2 + 264xy^3 + 257y^4.$$

For $r = 1$, we have the following Hankel matrix:

$$M = \begin{bmatrix} c_0 & c_1 \\ c_1 & c_2 \\ c_2 & c_3 \\ c_3 & c_4 \end{bmatrix} = \begin{bmatrix} 17 & 12 \\ 12 & 20 \\ 20 & 66 \\ 66 & 257 \end{bmatrix}.$$

This matrix is full column rank. Therefore, we build the Hankel matrix for $r = 2$:

$$M = \begin{bmatrix} c_0 & c_1 & c_2 \\ c_1 & c_2 & c_3 \\ c_2 & c_3 & c_4 \end{bmatrix} = \begin{bmatrix} 17 & 12 & 20 \\ 12 & 20 & 66 \\ 20 & 66 & 257 \end{bmatrix}.$$

This matrix is not full column rank, i.e., $\det(M) = 0$ (for a square matrix). Its rank is equal to 2, therefore we compute a basis of the kernel using the singular value decomposition $M = U\Sigma V^*$, where $rank(\Sigma) = 2$. $Ker(M)$ is the third column of V: $\mathbf{v} = [-0.3980, 0.8955, -0.1990]$.
We compute the roots of the univariate polynomial of degree 2, $g(x) = v_0x^2 + v_1x + v_2$. We find $x = (2, 0.25)$. Then, the roots of $q(x, y) = \sum_{i=0}^{2} v_i\, x^i y^{2-i}$ are $(\alpha_1, \beta_1) = (2, 1)$ and $(\alpha_2, \beta_2) = (0.25, 1)$. Lastly, we compute λ_1 and λ_2 by equating coefficients in the same monomials and we get the following final decomposition:

$$p(x, y) = (2x + y)^4 + 256(0.25x + y)^4.$$

Remark 2.1 **Degenerated cases**
Let $p(x, y)$ be a homogeneous polynomial of degree 3 in two variables. If the length of the column vector $\mathbf{v}$ is equal to $\ell = 3$, we define the univariate polynomial of degree 2 associated to $\mathbf{v}$: $g(x) = v_0\, x^2 + v_1\, x + v_2$.
Using Sylvester's algorithm, we have the following results according to the nature of the coefficients $b_i = v_i$:
- if $\forall i = 1, 2, 3$, $b_i \neq 0$ then the polynomial $g(x)$ admits two real or complex roots. The rank is equal to 2.
- if $v_0 \neq 0$, $v_1 \neq 0$ and $v_2 = 0$ then the polynomial $g(x)$ admits two roots but one of them is null. The rank is equal to 2.
- if $v_0 \neq 0$, $v_2 \neq 0$ and $v_1 = 0$ then the polynomial $g(x)$ admits two real or complex roots. The rank is equal to 2.
- if $v_1 \neq 0$, $v_2 \neq 0$ and $v_0 = 0$ then $g(x)$ is a monomial and admits one real root. The rank is equal to 1.
- if $v_0 = 0$, $v_2 = 0$ and $v_1 \neq 0$ then $g(x)$ is a monomial and admits one null root. The rank is equal to 1.
- if $v_1 = 0$, $v_2 = 0$ and $v_0 \neq 0$ then $g(x)$ is a monomial and admits two null roots. The rank is equal to 2.
- if $v_0 = 0$, $v_1 = 0$ and $v_2 \neq 0$ then $g(x)$ is a constant and has no root. No rank found.

In Remark 2.1, we discuss the different cases that we can be confronted with when we use Sylvester's algorithm to decompose a homogeneous polynomial of degree 3 in two variables. When the rank given by the decomposition algorithm is smaller than the generic rank, this can mean that the rank of this decomposition is in fact greater than the generic one.
For those homogeneous polynomial which decomposition rank is greater than the generic one, we can try to decompose them "by hand". Otherwise, to achieve the desired generic rank $r = 2$ for a homogeneous polynomial p of degree 3 in two variables, we can use a "noise effect" on p, i.e., we add at each coefficient of p a small value $\epsilon_i \approx 0$. That will not change the nature of the initial homogeneous polynomial but will help to avoid parse matrices in the singular value decomposition of the Hankel matrix

M. This idea allows to deal with all the generic and non generic cases that we have encountered in the binary case. We use the same reasoning for higher-degree, ($k \geqslant 4$).

We give an example to illustrate this remark.

Example 2.6
Consider the following third-order polynomial:

$$p(x, y) = x^3 + y^3.$$

For $r = 1$, we have the following Hankel matrix:

$$M = \begin{bmatrix} c_0 & c_1 \\ c_1 & c_2 \\ c_2 & c_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

This matrix is full column rank. Therefore, we build the Hankel matrix for $r = 2$:

$$M = \begin{bmatrix} c_0 & c_1 & c_2 \\ c_1 & c_2 & c_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

This matrix is not full column rank. Its rank is equal to 2, therefore we compute a basis of the kernel. We use the singular value decomposition and we get the following decomposition of the matrix:

$$M = U\Sigma V^*,$$

with

$$U = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}.$$

$Ker(M)$ is the third column of V: $\mathbf{v} = [0, -1, 0]$. Then, we compute the roots of $q(x, y) = \sum_{i=0}^{2} v_i\, x^i y^{2-i}$. $q(x, y)$ admits one root. $(\alpha_1, \beta_1) = (0, 1)$. We get one term instead of two terms for the decomposition. The decomposition of p that we get is not a good decomposition. We can not access to the value of (α_2) numerically. But manually, we find $(\alpha_2, \beta_2) = (1, 0)$.
A solution is to use a "noise effect". We propose to add a small value $\epsilon = 10^{-4}$ to the coefficients of $p(x, y)$. Then, we have the following polynomial:

$$p_{mod}(x, y) = 1.0001\, x^3 + 0.0001\, x^2 y + 0.0001\, xy^2 + 1.0001\, y^3.$$

For $r = 1$, we have the following Hankel matrix:

$$M = \begin{bmatrix} c_0 & c_1 \\ c_1 & c_2 \\ c_2 & c_3 \end{bmatrix} = \begin{bmatrix} 1.000100 & 0.000033 \\ 0.000033 & 0.000033 \\ 0.000033 & 1.000100 \end{bmatrix}.$$

This matrix is full column rank. Therefore, we build the Hankel matrix for $r = 2$:

$$M = \begin{bmatrix} c_0 & c_1 & c_2 \\ c_1 & c_2 & c_3 \end{bmatrix} = \begin{bmatrix} 1.000100 & 0.000033 & 0.000033 \\ 0.000033 & 0.000033 & 1.000100 \end{bmatrix}.$$

This matrix is not full column rank. Its rank is equal to 2, therefore we compute a basis of the kernel. We use the singular value decomposition and we get the following decomposition of the matrix:

$$M = U\Sigma V^*,$$

with

$$U = \begin{bmatrix} -0.7071067 & -0.707106 \\ -0.7071067 & 0.707106 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 1.000133 & 0 \\ 0 & 1.000066 \end{bmatrix},$$

$$V = \begin{bmatrix} -0.707106 & -0.707106 & 0.000033 \\ -0.000047 & 0.000000 & -0.999999 \\ -0.707106 & 0.707106 & 0.000033 \end{bmatrix}.$$

The third column of V is: $Ker(M) = \mathbf{v} = [0.000033, -0.999999, 0.000033]$. The roots of $q(x, y) = \sum_{i=0}^{2} v_i\, x^i y^{2-i}$ are $(\alpha_1, \beta_1) = (30003.99, 1)$ and $(\alpha_2, \beta_2) = (0.000033, 1)$. Lastly, we compute λ_1 and λ_2 by equating coefficients in the same monomials and we get the following final decomposition:

$$p(x, y) \sim p_{mod}(x, y) = 3.7e^{-14}(30003.99\, x + y)^3 + 1.0000999(0.000033\, x + y)^3.$$

Remark 2.2 The disappointing fact is that Sylvester's theorem cannot be extended to dimensions higher than 2. In fact, a key step in the proof [33] is that for any polynomial p of degree k, and any monomial m of degree $k - r$, there exists polynomial q of degree r such that qm is orthogonal to p. Equation (2.11) expresses that orthogonality in terms of polynomial coefficients. It is clear that this holds true only when $k \geqslant r$, which is unfortunately satisfied only in the binary case, according to **Table 2.1** [30].

2.3.2 Extension of Sylvester's approach to higher dimensions

In this section, we describe a new algorithm able to decompose a homogeneous polynomial of arbitrary degree and dimension as a sum of powers of linear forms. It has been proposed by Brachat et al. [17]. It generalizes the algorithm of Sylvester and extends the principle of Sylvester's theorem to higher dimensions. In the binary case, the decomposition problem is solved directly by computing ranks of catalecticant or hankel matrix. In higher dimensions, this is not so simple. An extension step is required to find the decomposition. This leads to the resolution of a polynomial system of small degree, from which we deduce the decomposition by solving a simple eigenvalue problem, thanks to linear algebra manipulations.

Algorithm 2: SYMMETRIC TENSOR DECOMPOSITION

Input: *A homogeneous polynomial* $f(\mathbf{x}) = \sum_{i=0}^{k} \sum_{j=0}^{k-i} c_{ij} x^i y^j z^{k-i-j}$ *of degree* k.

Output: *A decomposition of* f *as* $f(\mathbf{x}) = \sum_{i=1}^{r} \lambda_i \, \ell_i(\mathbf{x})^k$ *with* r *minimal.*

(1) Compute the coefficients of the dual $f^* : c_\alpha = a_\alpha \binom{k}{\alpha}^{-1}$, *for* $|\alpha| \leqslant k$.
(2) Initialize $r = 0$.
(3) Increment $r \leftarrow r + 1$.
(4) Specialization:

- *Take any basis* B *of monomials of degree* $\leqslant k$ *connected to (1) with* $|B| \leqslant r$.
- *Build the matrix* $\mathbb{H}^{B^+}_{f^*(\mathbf{h})}$ *with the coefficients* c_α.
- *If there exist a non-zero minor of order* $r+1$ *in* $\mathbb{H}^{B^+}_{f^*(\mathbf{h})}$, *without coefficients depending on* $\mathbf{h}$, *restart the loop with* $r := r + 1$*; i.e., go to step* (3) *and try another specialization.*
- *Else if any minors of order* $r+1$ *in* $\mathbb{H}^{B^+}_{f^*(\mathbf{h})}$, *without coefficients depending on* $\mathbf{h}$, *vanish, compute* $\mathbf{h}$ *such that* $\det(\mathbb{H}^{B}_{f^*(\mathbf{h})}) \neq 0$ *and the formal multiplication operators* $\mathbb{M}_i = \mathbb{H}^{B}_{x_i f^*(\mathbf{h})} (\mathbb{H}^{B}_{f^*(\mathbf{h})})^{-1}$ *commute.*
  If there is no solution, restart the loop with $r := r + 1$*; i.e., go to step* (3)
- *Else compute the* $n \times r$ *eigenvalues* $\zeta_{i,j}$ *and the eigenvectors* $\mathbf{v}_j$ *such that* $\mathbb{M}_i \mathbf{v}_j = \zeta_{i,j} \mathbf{v}_j$, *with* $i = 1, ..., n$ *and* $j = 1, ..., r$*;*
  until *the eigenvalues* $\mathbf{v}_j$ *are simple.*

(5) Solve the linear system in $(\lambda_j)_{j=1,...,r}$*:* $f(\mathbf{x}) = \sum_{j=1}^{r} \lambda_j \, \ell_j(\mathbf{x})^k$. *The coefficients of* ℓ_j *are the eigenvectors* $\mathbf{v}_j$ *found in step* (4).

A partial Matlab code has been proposed in **Appendix B.2** to decompose third-order symmetric tensors in three variables into a sum of three linear terms. We will see in the sequel and in the next chapter the reason why we impose three linear terms instead of four linear terms.
For more details about algebraic tools used in this algorithm, see **Appendix A**.

We briefly explain **Algorithm 2** before giving some examples to show how it works.

Consider a homogeneous polynomial $f(\mathbf{x})$ of degree k and dimension n:

$$f(\mathbf{x}) = \sum_{j_0+...+j_n=k} a_{j_0,...,j_n} x_0^{j_0} ... x_n^{j_n}$$

that we want to decompose. We may assume without loss of generality, that for at least one variable, say x_0, all its coefficients in the decomposition are non-zero coefficients, i.e $\ell_{i,0} \neq 0$, $1 \leqslant i \leqslant r$. We dehomogenize f with respect to this variable and we denote this polynomial by $f^a := f(1, x_1, ..., x_n)$. We want to decompose f^a or equivalently, its corresponding dual element f^* as a sum of powers of linear forms, i.e.,

$$f^*(\mathbf{x}) = \sum_{i=1}^{r} \lambda_i \left(1 + \ell_{i,1}x_1 + ... + \ell_{i,n}x_n\right)^k = \sum_{i=1}^{r} \lambda_i \, \ell_i(\mathbf{x})^k,$$

where $\ell_i(\mathbf{x}) = 1 + \ell_{i,1}x_1 + ... + \ell_{i,n}x_n$.
Assume that we know the value of r. In this case, knowing the value of f^* on polynomials of degree high enough allows us to compute the table of multiplications modulo the kernel of $\mathbb{H}_{f^*}$. By solving the generalized eigenvalue problem $(\mathbb{H}_{x_i f^*} - \lambda \mathbb{H}_{f^*})\mathbf{v} = 0$, we will recover the points of evaluation ℓ_j. By solving a linear system, we will then deduce the value of $\lambda_1, ..., \lambda_r$. Thus the goal of this algorithm is to extend f^* on a large enough set of polynomials, in order to be able to run the eigenvalue problem.

Example 2.7
Consider a symmetric tensor of dimension 3 and order 3, which corresponds to the following homogeneous polynomial:

$$\begin{aligned} f(x,y,z) &= 4x^3 + 9x^2y + 39xy^2 + 9y^3 - 3x^2z + 93xz^2 - 19z^3 - 39y^2z \\ &\quad +111yz^2 - 78xyz. \end{aligned}$$

- We compute the coefficients $(c_{i_0,i_1,...,i_n})$ of the dual element f^* in the dual basis B^+ from the coefficients $(a_{i_0,i_1,...,i_n})$ of the polynomial f in the monomial basis B:

$$c_{j_0,j_1,...,j_n} := a_{j_0,j_1,...,j_n}. \binom{k}{j_0, ..., j_n}^{-1}, \; n = 3, \; k = 3.$$

- We form a $\binom{n+k-1}{k} \times \binom{n+k-1}{k}$ matrix, the rows and the columns of which correspond to the coefficients of the polynomial f^* in the dual basis. This matrix $\mathbb{H}^{B^+}_{f^*(\mathbf{h})}$ is the formal Hankel matrix associated to f^* and is called *quasi-Hankel* or *Catalecticant*.

Taking a connected basis with $r = 1$ and $r = 2$ elements, we find non-zero minors of degree 2 and 3 respectively, in $\mathbb{H}^{B^+}_{f^*(\mathbf{h})}$. Hence, f has no rank equal to 1 or 2.

For $B = \{1, y, z\}$, then $B^+ = \{1, y, z, y^2, yz, z^2\}$. We show only the 6×6 principal

minor of the matrix $\mathbb{H}^{B^+}_{f^*(\mathbf{h})}$:

$$\left(\begin{array}{c|cccccc} & 1 & y & z & y^2 & yz & z^2 \\ \hline 1 & 4 & 3 & -1 & 13 & -13 & 31 \\ y & 3 & 13 & -13 & 9 & -13 & 37 \\ z & -1 & -13 & 31 & -13 & 37 & -19 \\ y^2 & 13 & 9 & -13 & h_{040} & h_{031} & h_{022} \\ yz & -13 & -13 & 37 & h_{031} & h_{022} & h_{013} \\ z^2 & 31 & 37 & -19 & h_{022} & h_{013} & h_{044} \end{array}\right),$$

where h_i are unknown parameters.

- We extract from $\mathbb{H}^{B^+}_{f^*(\mathbf{h})}$ a principal minor of full rank (i.e., determinant $\neq 0$ for a square matrix). We should re-arrange the rows and the column of the matrix so that there is a principal minor of full rank. We call this minor Δ_0. To do so, we try to put the matrix in row echelon form, using elementary row and column operations.
In our example, the 3×3 principal minor is of full rank, so there is no need for re-arranging the matrix:

$$\Delta_0 = \mathbb{H}^B_{f^*} = \begin{pmatrix} 4 & 3 & -1 \\ 3 & 13 & -13 \\ -1 & -13 & 31 \end{pmatrix}, \quad \det(\Delta_0) \neq 0.$$

- We compute the matrices $\Delta_1 = y\Delta_0$ and $\Delta_2 = z\Delta_0$. The columns of Δ_1 and Δ_2 correspond to the monomials $\{y, y^2, yz\}$ and $\{z, yz, z^2\}$ respectively. They are just the corresponding monomials of the columns of Δ_0, i.e., $\{1, y, z\}$ multiplied by y and $\{1, y, z\}$ multiplied by z respectively.

$$\Delta_1 = \mathbb{H}^B_{yf^*} = \begin{pmatrix} 3 & 13 & -13 \\ 13 & 9 & -13 \\ -13 & -13 & 37 \end{pmatrix},$$

$$\Delta_2 = \mathbb{H}^B_{zf^*} = \begin{pmatrix} -1 & -13 & 31 \\ -13 & -13 & 37 \\ 31 & 37 & -19 \end{pmatrix}.$$

We need that the multiplication operators $\mathbb{M}_i = \mathbb{H}^B_{x_i f^*(\mathbf{h})}(\mathbb{H}^B_{f^*(\mathbf{h})})^{-1}$ commute. That is $\mathbb{M}^B_y \mathbb{M}^B_z = \mathbb{M}^B_z \mathbb{M}^B_y$. We have:

$$\mathbb{M}^B_y = \mathbb{H}^B_{yf^*(\mathbf{h})}(\mathbb{H}^B_{f^*(\mathbf{h})})^{-1} = \begin{pmatrix} 0 & 3.6842 & -4.1053 \\ 1 & -0.7895 & 1.7368 \\ 0 & -0.6316 & 1.7895 \end{pmatrix},$$

$$\mathbb{M}^B_z = \mathbb{H}^B_{zf^*(\mathbf{h})}(\mathbb{H}^B_{f^*(\mathbf{h})})^{-1} = \begin{pmatrix} 0 & -4.1053 & 6.6316 \\ 0 & 1.7368 & 1.5789 \\ 1 & 1.7895 & 0.2632 \end{pmatrix}.$$

Thus we have:

$$\mathbb{M}_y^B\mathbb{M}_z^B = \mathbb{M}_z^B\mathbb{M}_y^B = \begin{pmatrix} -4.1053 & -0.9474 & 4.7368 \\ 1.7368 & -2.3684 & 5.8421 \\ 1.7895 & 2.1053 & -0.5263 \end{pmatrix}.$$

It holds that the multiplication operators commute. It should be noted that in this step the algorithm has to compute the parameters $\mathbf{h}$ such that the multiplication operators commute but in this case all our entries are known.

- Now, we solve the generalized eigenvalue problem $(\Delta_1 - \lambda\Delta_0)\mathbf{v} = 0$. We normalized the elements of the eigenvectors so that the first element is equal to 1. Then, we get the following generalized eigenvectors:

$$\begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}.$$

The coordinates of the eigenvectors correspond to the elements $\{1, y, z\}$. Thus we can recover the coefficients of y and z in the decomposition from these coordinates.

- It remains to compute the coefficients λ_i of the decomposition:

$$f(x, y, z) = \lambda_0(x - 2y + 2z)^3 + \lambda_1(x + 2y + 3z)^3 + \lambda_2(x + y + 3z)^3.$$

We do this easily by solving a linear system which has a solution, since the decomposition exists. Doing that, we deduce $\lambda_0 = 1$, $\lambda_1 = 2$ and $\lambda_2 = 1$.

-Finally, the homogeneous polynomial f admits the following decomposition:

$$f(x, y, z) = (x - 2y + 2z)^3 + 2(x + 2y + 3z)^3 + (x + y + 3z)^3.$$

Remark 2.3 In this example, all the elements of the matrices Δ_0 and Δ_1 are known. If this is not the case, we can compute the unknown entries h_i of the matrix using either necessary and sufficient conditions of commutation $\mathbb{M}_i\mathbb{M}_j - \mathbb{M}_j\mathbb{M}_i = 0$ for any $i, j \in \{1, ..., n\}$, with $\mathbb{M}_i = \mathbb{H}^B_{x_i f^*(\mathbf{h})}(\mathbb{H}^B_{f^*(\mathbf{h})})^{-1} = \Delta_i\Delta_0^{-1}$. This leads to the resolution of polynomial equations of small degree in non-generic cases. We give an example to illustrate this remark. But for more details, see [16, 17].

The following example has been proposed and studied in [16, 17].

Example 2.8
- Consider a symmetric tensor of dimension 3 and order 4, that corresponds to the following homogeneous polynomial

$$f(x, y, z) = 79xy^3 + 56x^2z^2 + 49y^2z^2 + 4xyz^2 + 57x^3y.$$

According to the **Table 2.1** of generic ranks, the rank of this function is 6.

- As previously, we have to compute the formal Hankel matrix associated to the formal linear form $f^*(\mathbf{h})$. We show only the 10×10 principal minor of the matrix $\mathbb{H}^{B^+}_{f^*(\mathbf{h})}$:

$$\left(\begin{array}{c|cccccccccc} & 1 & y & z & y^2 & yz & z^2 & y^3 & y^2z & yz^2 & z^3 \\ \hline 1 & 0 & \frac{57}{4} & 0 & 0 & 0 & \frac{28}{3} & \frac{79}{4} & 0 & \frac{1}{3} & 0 \\ y & \frac{57}{4} & 0 & 0 & \frac{79}{4} & 0 & \frac{1}{3} & 0 & 0 & \frac{49}{6} & 0 \\ z & 0 & 0 & \frac{28}{3} & 0 & \frac{1}{3} & 0 & 0 & \frac{49}{6} & 0 & 0 \\ y^2 & 0 & \frac{79}{4} & 0 & 0 & 0 & \frac{49}{6} & h_{500} & h_{410} & h_{320} & h_{230} \\ yz & 0 & 0 & \frac{1}{3} & 0 & \frac{49}{6} & 0 & h_{410} & h_{320} & h_{230} & h_{140} \\ z^2 & \frac{28}{3} & \frac{1}{3} & 0 & \frac{49}{6} & 0 & 0 & h_{320} & h_{230} & h_{140} & h_{050} \\ y^3 & \frac{79}{4} & 0 & 0 & h_{500} & h_{410} & h_{320} & h_{600} & h_{510} & h_{420} & h_{330} \\ y^2z & 0 & 0 & \frac{49}{6} & h_{410} & h_{320} & h_{230} & h_{510} & h_{420} & h_{330} & h_{240} \\ yz^2 & \frac{1}{3} & \frac{49}{6} & 0 & h_{320} & h_{230} & h_{140} & h_{420} & h_{330} & h_{240} & h_{150} \\ z^3 & 0 & 0 & 0 & h_{230} & h_{140} & h_{050} & h_{330} & h_{240} & h_{150} & h_{060} \end{array}\right),$$

where h_i are unknown parameters.

- For $B^+ = \{1, y, z, y^2, yz, z^2\}$, the 6×6 principal minor is of full rank. Then the matrix Δ_0 is:

$$\Delta_0 = \begin{pmatrix} 0 & \frac{57}{4} & 0 & 0 & 0 & \frac{28}{3} \\ \frac{57}{4} & 0 & 0 & \frac{79}{4} & 0 & \frac{1}{3} \\ 0 & 0 & \frac{28}{3} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{79}{4} & 0 & 0 & 0 & \frac{49}{6} \\ 0 & 0 & \frac{1}{3} & 0 & \frac{49}{6} & 0 \\ \frac{28}{3} & \frac{1}{3} & 0 & \frac{49}{6} & 0 & 0 \end{pmatrix}.$$

- We compute the matrices $\Delta_1 = y\Delta_0$ and $\Delta_2 = z\Delta_0$ whose the columns correspond to the monomial $\{y, y^2, yz, y^3, y^2z, yz^2\}$ and $\{z, yz, z^2, y^2z, yz^2, z^3\}$ respectively.

$$\Delta_1 = \mathbb{H}^B_{yf^*} = \begin{pmatrix} \frac{57}{4} & 0 & 0 & \frac{79}{4} & 0 & \frac{1}{3} \\ 0 & \frac{79}{4} & 0 & 0 & 0 & \frac{49}{6} \\ 0 & 0 & \frac{1}{3} & 0 & \frac{49}{6} & 0 \\ \frac{79}{4} & 0 & 0 & h_{500} & h_{410} & h_{320} \\ 0 & 0 & \frac{49}{6} & h_{410} & h_{320} & h_{230} \\ \frac{1}{3} & \frac{49}{6} & 0 & h_{320} & h_{230} & h_{140} \end{pmatrix},$$

$$\Delta_2 = \mathbb{H}^B_{zf^*} = \begin{pmatrix} 0 & 0 & \frac{28}{3} & 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} & 0 & \frac{49}{6} & 0 \\ \frac{28}{3} & \frac{1}{3} & 0 & \frac{49}{6} & 0 & 0 \\ 0 & 0 & \frac{49}{6} & h_{410} & h_{320} & h_{230} \\ \frac{1}{3} & \frac{49}{6} & 0 & h_{320} & h_{230} & h_{140} \\ 0 & 0 & 0 & h_{230} & h_{140} & h_{050} \end{pmatrix}.$$

We have to determine the unknown variables h_i. To do so, we consider the following method. We form all the possible matrix equations $\mathbb{M}_i\mathbb{M}_j^B - \mathbb{M}_j^B\mathbb{M}_i^B = 0$. There are $\binom{n}{2}$ equations and we equate their elements to zero. Since the dimension of the matrices is $r \times r$, this leads to at most $\binom{n}{2}r^2$ or $\mathcal{O}(n^2r^2)$ equations. Note that the equations are, at most of degree 2.

Then, the matrices Δ_1 and Δ_2 have to satisfy the matrix equation $\mathbb{M}_y^B\mathbb{M}_z^B - \mathbb{M}_z^B\mathbb{M}_y^B = 0$, for any x,y, i.e., the matrices of multiplication commute.

$$\mathbb{M}_y^B\mathbb{M}_z^B - \mathbb{M}_z^B\mathbb{M}_y^B = \Delta_1\Delta_0^{-1}\Delta_2\Delta_0^{-1} - \Delta_2\Delta_0^{-1}\Delta_1\Delta_0^{-1} = 0.$$

This matrix relation involves polynomial equations of degree 2. Many of the resulting equations are trivial. After discarding them, we have 6 unknowns $\{h_{500}, h_{410}, h_{320}, h_{230}, h_{140}, h_{050}\}$. One solution of this system of equations is:

$$\{h_{500} = 1, h_{410} = 2, h_{320} = 3, h_{230} = 1.506, h_{140} = 4.96, h_{050} = 0.056\}.$$

We substitute these values to Δ_1 and we continue the algorithm as in the previous example.

- We solve the generalized eigenvalue problem $(\Delta_1 - \lambda\Delta_0)\mathbf{v} = 0$. Then, we get the following normalized eigenvectors:

$$\begin{bmatrix} 1 \\ -0.830 + 1.593i \\ -0.326 - 0.0501 \\ -1.849 - 2.645i \\ 0.350 - 0.478i \\ 0.103 + 0.0327i \end{bmatrix}, \begin{bmatrix} 1 \\ -0.830 - 1.593i \\ -0.326 + 0.0501 \\ -1.849 + 2.645i \\ 0.350 + 0.478i \\ 0.103 - 0.0327i \end{bmatrix}, \begin{bmatrix} 1 \\ 1.142 \\ 0.836 \\ 1.305 \\ 0.955 \\ 0.699 \end{bmatrix},$$

$$\begin{bmatrix} 1 \\ 10.956 \\ -0.713 \\ 0.914 \\ -0.682 \\ 0.509 \end{bmatrix}, \begin{bmatrix} 1 \\ -0.838 + 0.130i \\ 0.060 + 0.736i \\ 0.686 - 0.219i \\ -0.147 - 0.610i \\ -0.539 + 0.089i \end{bmatrix}, \begin{bmatrix} 1 \\ -0.838 - 0.130i \\ 0.060 - 0.736i \\ 0.686 + 0.219i \\ -0.147 + 0.610i \\ -0.539 - 0.089i \end{bmatrix}.$$

The coordinates of the eigenvectors correspond to the elements $\{1, y, z, y^2, yz, z^2\}$ and we can recover the coefficients of y and z in the decomposition.

After solving the over-constrained linear system for the coefficients of the linear form, we deduce the following decomposition of f:

$$\begin{aligned} f(x,y,z) &= (0.517 + 0.044i)(x - (0.830 - 1.593i)y - (0.326 + 0.0501)z)^4 \\ &+ (0.517 - 0.044i)(x - (0.830 + 1.593i)y - (0.326 - 0.0501)z)^4 \\ &+ 2.958(x + 1.142y + 0.836z)^4 + 4.583(x + 0.956y - 0.713z)^4 \\ &- (4.288 + 1.119i)(x - (0.838 - 0.13i)y + (0.060 + 0.736i)z)^4 \\ &- (4.288 - 1.119i)(x - (0.838 + 0.13i)y + (0.060 - 0.736i)z)^4. \end{aligned}$$

Remark 2.4 Example 2.8 is an example of tensor decomposition where we have a system of equations with a large number of unknowns. This kind of systems of equations doesn't admit a unique solution, we can have an infinity of solutions. But more than the non-uniqueness, it is not so easy to solve these systems. For the moment, the only results found in the theory and the practice are given by "maple", the formal calculation software. As we can see, we are very limited in terms of numerical applications.
However, concerning third-order tensor decomposition, we proposed a solution to get an approximation of the desired decomposition in 3D. This solution will be exposed in the next chapter and will lead to good numerical results.

2.3.3 The multi-way CANDECOMP/PARAFAC ("CP") model

In the previous two sections, we work with homogeneous polynomials. But we can also work with symmetric tensors of arbitrary order and dimension. In this section, we present one among the various SVD-based algorithms to compute Canonical Decomposition of k^{th}-order tensors in higher dimensions. This algorithm is the *Multi-way Parallel Factor* (PARAFAC) model.

In order to fix the ideas, we take a third-order symmetric tensor $\mathbf{X} = (X_{ijk})_{i,j,k}$ of any dimension. Once the three bases are chosen in each of the three vector spaces, the tensor is defined by a 3-way array. Its "CP" decomposition takes the following linear form:

$$\mathbf{X} \approx (\mathbf{A}, \mathbf{B}, \mathbf{C}).\mathbf{\Lambda}, \tag{2.12}$$

$$X_{ijk} \approx \sum_{\ell=1}^{R} \lambda_\ell \, A_{i\ell} \, B_{j\ell} \, C_{k\ell}, \tag{2.13}$$

with R the tensor rank, i.e., the minimal number of rank-1 terms such that Equality (2.12) holds true.

The previous model (2.12) can be written in a compact form using the Khatri-Rao product $\odot$ (column-wise Kronecker product) as, possibly up to an error term,

$$\mathbf{X}_{I \times JK} \approx \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T, \tag{2.14}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are called loading matrices of $\mathbf{X}$ of size $I \times R$, $J \times R$, and $K \times R$ and $\mathbf{X}_{I \times JK}$ is the matrix of size $I \times JK$ obtained by unfolding the array array $\mathbf{X}$ of size $I \times J \times K$.
There exists several algorithms that fit the "CP" model. We focus on the most widely used among all: the Alternating Least squares (ALS) algorithm.

The first step consists in reducing the dimensions of the problem, and at the same time reducing the rank of $\mathbf{X}$, by truncating the Singular Value Decomposition of unfolding matrices.

The second step (the ALS part) consists in estimating one of the three matrices at each step by minimizing in the least squares sense and in an alternating way the following cost function:

$$\varphi = ||\mathbf{X}_{I\times JK} - \mathbf{A}(\mathbf{C}\odot\mathbf{B})^T||_F^2, \tag{2.15}$$

where $||\bullet||_F$ is the Frobenius norm.
In an "Alternating way" means that, at each iteration, we minimize φ with respect to $\mathbf{A}$, given $\mathbf{B}$ and $\mathbf{C}$ fixed, then update $\mathbf{B}$, given $\mathbf{A}$ and $\mathbf{C}$ fixed and finally update $\mathbf{C}$, given $\mathbf{A}$ and $\mathbf{B}$ fixed. For more details, see [18, 85, 86].

With $\mathbf{B}$ and $\mathbf{C}$ fixed to initial values, the estimate of $\mathbf{A}$ in the least squares sense is given by:

$$\mathbf{A}^T = (\mathbf{B}\odot\mathbf{C})^\dagger\,\mathbf{X}_{I\times JK}.$$

Similarly for $\mathbf{B}$ and $\mathbf{C}$, we write:

$$\mathbf{B}^T = (\mathbf{C}\odot\mathbf{A})^\dagger\,\mathbf{X}_{J\times KI},$$

$$\mathbf{C}^T = (\mathbf{A}\odot\mathbf{B})^\dagger\,\mathbf{X}_{K\times IJ},$$

where $\mathbf{M}^\dagger$ is the pseudo-inverse of $\mathbf{M}$, i.e., $\mathbf{M}^\dagger = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$. In **Figure 2.1** below, we note $+$, the pseudo-inverse of a matrix. This figure is from the article [80].

It has been pointed out the important fact that, when the convergence of the ALS algorithm is slow (very large number of iterations are needed to reach the final solution of a given cycle), we need to speed up the "CP" model. To do so, the unknowns can be linearly interpolated at each iteration and the interpolated matrices are used as inputs of the current ALS update. This is the *Line Search* (LS) method. Several line Search techniques are possible, see [32, 75, 80]. One of the proposed modifications of the ALS algorithm we use here is the *Enhanced Line Search* (ELS) [75, 80].

As previously stated, the idea of the Line Search method consists in predicting the value of the loading factor a certain number of iterations ahead by computing a sort of linear regression:

$$\mathbf{A}^{new} = \mathbf{A}^{it-2} + R_{LS}\,\mathbf{G}_a^{it},$$

$\mathbf{A}^{it-1}$ is the estimation of matrix $\mathbf{A}$ obtained in the ALS iteration $it-1$, and $\mathbf{A}^{new}$ is the matrix that will be used in the it^{th} iteration of $\mathbf{A}^{it-1}$. $\mathbf{G}_a^{it} = \mathbf{A}^{it-1} - \mathbf{A}^{it-2}$ defines the direction of the cycle. Matrices $\mathbf{B}^{new}$ and $\mathbf{C}^{new}$ are obtained in an equivalent way using the same relaxation factor R_{LS}.

The Enhanced Line Search is performed at the beginning of the ALS algorithm and consists in seeking the optimal relaxation factor R_{LS} that leads to the final solution of a given cycle in only one step. So, for iteration it, we look for the optimal triplet (R_a, R_b, R_c) that minimizes:

$$\gamma_{ELS} = ||\mathbf{X}_{I\times JK} - (\mathbf{A}^{it-2} + R_a\,\mathbf{G}_a^{it})((\mathbf{B}^{it-2} + R_b\,\mathbf{G}_b^{it})\odot(\mathbf{C}^{it-2} + R_c\,\mathbf{G}_c^{it}))||_F^2. \tag{2.16}$$

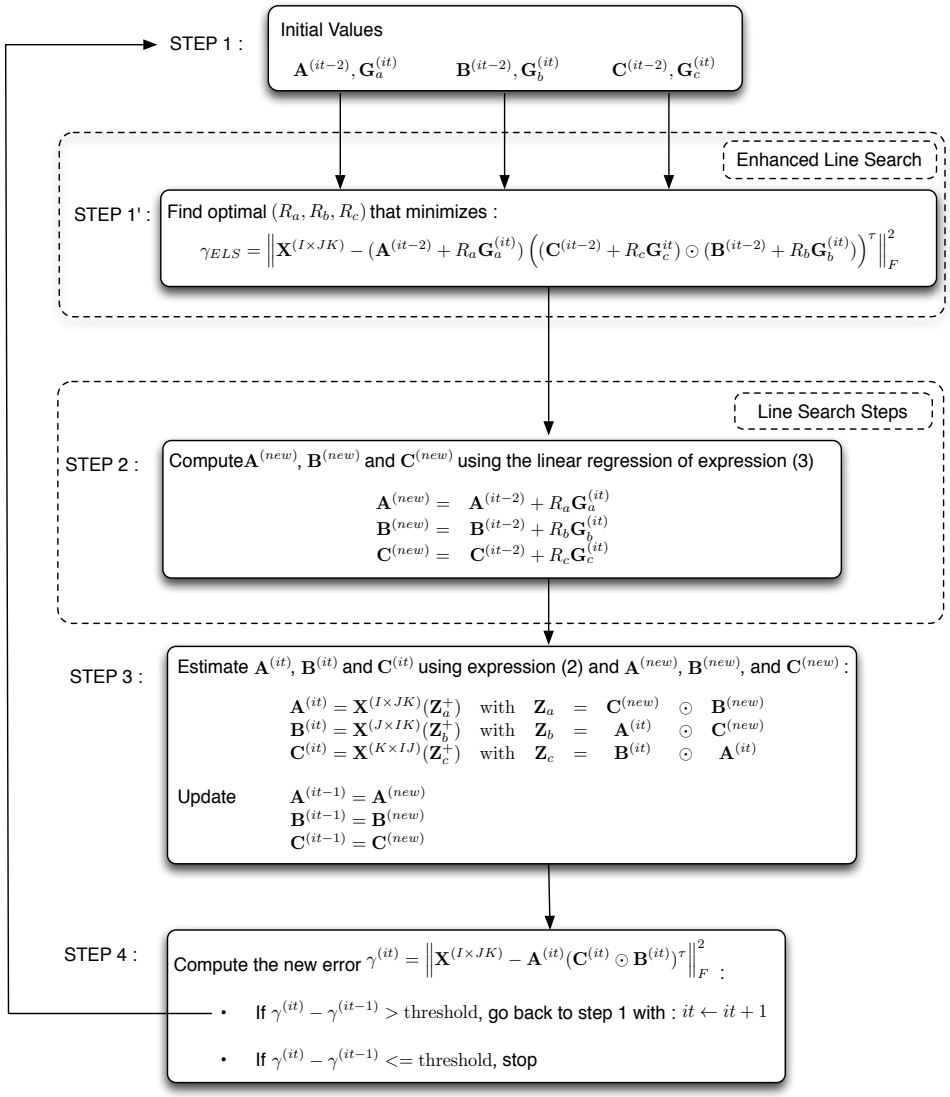

Figure 2.1: Steps of the Alternating Least Squares algorithm with Enhanced Line Search [80].

The optimal solution is obtained when we jointly minimize γ_{ELS} with respect to three different factors (R_a, R_b, R_c). In this case, the problem consists in solving a system of three polynomials in three unknowns, which leads to a higher numerical complexity. To reduce this complexity, we choose to work with the same factors for all modes $R_a = R_b = R_c = R_{LS}$. It involves a polynomial in a single unknown R of degree 6:

$$\begin{aligned} \gamma_{ELS}(R) &= \sum_{ijk}(X_{ijk} - \sum_{f=1}^{R}(A_{if} + R_{LS}\, G_{a,if})(B_{jf} + R_{LS}\, G_{b,jf})(C_{kf} + R_{LS}\, G_{c,kf}))^2 \\ \gamma_{ELS}(R) &= \sum_{d=0}^{6} p_d\, R_{LS}^d. \end{aligned} \tag{2.17}$$

where p_d are functions of observed values stored array $\mathbf{X}$ and coefficients of loading matrices of iterations $it-1$ and $it-2$. To find the optimal R, we determine the roots of polynomial $\gamma'_{ELS}(R)$, which provides five possible values of R. We feed those values into Expression (2.17).

Once the three matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are found, we can rewrite the previous decomposition of $\mathbf{X}$ in polynomial form, i.e., into a sum of third powers of r distinct linear forms in $\mathbb{C}$.

Remark 2.5 In the ALS algorithm, a given number of initializations are tested.

(a) - If the dimensions of the tensor allow it, namely if $\mathbf{X}$ has two dimensions higher than the rank R, say $I \geqslant R$ and $J \geqslant R$, the first initialization is built by exploiting the tensor itself. Indeed, the third dimension K, is first reduced to 2 by a singular value decomposition, after which a Generalized Eigenvalue Decomposition called DTLD (Direct Trilinear Decomposition [65]) is applied on the two $I \times J$ slices of the matrix pencil. The other initializations are all random.

(b) - Otherwise, all the initializations are random.

(c) - If $\mathbf{A}$ and/or $\mathbf{B}$ and/or $\mathbf{C}$ are provided as input arguments to enforce the use of this (these) matrix(ces) as starting point(s), thus, the number of initializations is set to 1 and is the only one used.

In the sequel, we explain how we get the polynomial form of the tensor decomposition from the three matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ in 2D and 3D.

• **Two-dimensional case**

$\mathbf{X}$ is a third-order symmetric tensor $\mathbf{X} = (X_{ijk})_{i,j,k} \in \mathbb{C}^{2\times 2\times 2}$ of dimension 2. We have found three matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C} \in \mathbb{C}^{2\times 2}$ such that:

$$X_{ijk} \approx \sum_{\ell=1}^{R} \lambda_\ell \, A_{i\ell} \, B_{j\ell} \, C_{k\ell},$$

with $R = 2$ the tensor rank.

We use the Khatri-Rao product $\odot$ to write the symmetric tensor $\mathbf{X}$ as a sum of 2 symmetric tensors:

$$\mathbf{X}_{I\times JK} = \mathbf{X}_1 + \mathbf{X}_2,$$

with $\mathbf{X}_i = A(:,i).(C(:,i) \odot B(:,i))^T$, for $i = 1, 2$. $A(:,i)$, $C(:,i)$, $B(:,i)$ are the i^{th} column vectors of matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ respectively.

As each $\mathbf{X}_i$ is a symmetric tensor, each of them can be written as a homogeneous polynomial form p_i of degree 3 in two variables and then as a third power of a linear form. These linear forms are distinct in $\mathbb{C}$. So,

$$p_i = \mu_i(\alpha_i x + \beta_i y)^3,$$

with

$$\mu_i = \mathbf{X}_i(I, JK),$$

$$\alpha_i = sign\left(\frac{\mathbf{X}_i(1,1)}{\mu_i}\right)\left|\frac{\mathbf{X}_i(1,1)}{\mu_i}\right|, \quad \beta_i = 1,$$

with $i = 1, 2$ and $I = J = K = 2$ the dimensions of the tensor $\mathbf{X}$. $X_i(1,1)$ is the element of the first line and first column of the matrix unfolding $\mathbf{X}_i$.
Then we get the following polynomial form of $\mathbf{X}$ in 2D:

$$\begin{aligned} p_{\mathbf{X}} &= p_1 + p_2, \\ p_{\mathbf{X}} &= \mu_1(\alpha_1 x + y)^3 + \mu_2(\alpha_2 x + y)^3. \end{aligned}$$

- **Three-dimensional case**

The reasoning is the same as the two-dimensional case. $\mathbf{X}$ is a third-order symmetric tensor $\mathbf{X} = (X_{ijk})_{i,j,k} \in \mathbb{C}^{3\times3\times3}$ of dimension 3. We have found three matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C} \in \mathbb{C}^{3\times3}$ that satisfies:

$$X_{ijk} \approx \sum_{\ell=1}^{R} \lambda_\ell \, A_{i\ell} \, B_{j\ell} \, C_{k\ell},$$

with $R = 3$ the tensor rank.

Using the Khatri-Rao product $\odot$, we write the symmetric tensor $\mathbf{X}$ as a sum of three symmetric tensors:

$$\mathbf{X}_{I\times JK} = \mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3,$$

with $\mathbf{X}_i = A(:,i).(C(:,i) \odot B(:,i))^T$, for $i = 1, 2, 3$. $A(:,i)$, $C(:,i)$, $B(:,i)$ are the i^{th} column vectors of matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ respectively.
As each $\mathbf{X}_i$ is a symmetric tensor, each of them can be written as a homogeneous polynomial form p_i of degree 3 in three variables and then as a third power of a linear form. These linear forms are distinct in $\mathbb{C}$. Then,

$$p_i = \mu_i(\alpha_i x + \beta_i y + \gamma_i z)^3,$$

with

$$\mu_i = \mathbf{X}_i(I, JK),$$

$$\alpha_i = sign\left(\frac{\mathbf{X}_i(1,1)}{\mu_i}\right)\left|\frac{\mathbf{X}_i(1,1)}{\mu_i}\right|, \quad \beta_i = sign\left(\frac{\mathbf{X}_i(2,5)}{\mu_i}\right)\left|\frac{\mathbf{X}_i(2,5)}{\mu_i}\right|,$$

$$\gamma_i = 1.$$

with $i = 1, 2, 3$ and $I = J = K = 3$ the dimensions of the tensor $\mathbf{X}$. $\mathbf{X}_i(2, 5)$ is the element of the second line and fifth column of the matrix unfolding $\mathbf{X}_i$.
Then we get the following polynomial form of $\mathbf{X}$ in 3D:

$$\begin{aligned} p_{\mathbf{X}} &= p_1 + p_2 + p_3, \\ p_{\mathbf{X}} &= \mu_1(\alpha_1 x + \beta_1 y + z)^3 + \mu_2(\alpha_2 x + \beta_2 y + z)^3 + \mu_3(\alpha_3 x + \beta_3 y + z)^3. \end{aligned}$$

2.4 CONCLUSION

In this chapter, two algorithms have been presented:

- Sylvester's algorithm and its extension to higher dimensions to decompose homogeneous polynomials in any variables and arbitrary degree into a sum of powers of linear forms,
- the CANDECOMP/PARAFAC ("CP") model to decompose symmetric tensors of arbitrary order and dimension as a sum of rank-1 symmetric tensors. In the sequel, we use the CP3_{alsls} (third-order CANDECOMP/PARAFAC) decomposition algorithm developed by Nion and Lathauwer [75]. For more details, you can visit Nion's website (*http://perso-etis.ensea.fr/∼nion/*).

Sylvester's algorithm extended to higher dimensions has two important impacts. First, it permits an efficient computation of the decomposition of any tensor of subgeneric rank, as opposed to widely used iterative algorithms with unproved global convergence like Alternating Least Squares fitting the "CP" model. Second, it gives tools for understanding uniqueness conditions and for detecting the rank. Symmetric tensor decompositions are going to play an important role in higher-order mesh adaptation.

3

Higher-order interpolation error estimate

Contents

3.1 INTRODUCTION TO HIGHER-ORDER INTERPOLATIONS

3.1.1 Motivations

The importance of unstructured anisotropic mesh adaptation has been proved in many studies [4, 44, 48]. Indeed, some physical phenomena have strong anisotropic properties. To capture them, we need to generate a mesh which converges simultaneously with the desired solution. Hessian-based methods and in particular multi-scale ones, i.e., relying on L^p error norm, have shown to be a powerful tool for building anisotropic meshes allowing a faster convergence to continuous solutions. In particular, the numerical convergence is close to second order in the linear case, even for coarse meshes. Theoretical investigations tend to show that these favorable properties will also hold at a higher order, as far as a higher-order scheme and a higher-order analysis are employed.

In the context of anisotropic mesh adaptation, there are few results, both theoretical and practical for higher-order interpolations. This can be explained by the number of possible interpolations and their complexity. Thus, it is difficult to be entirely generic. Another complexity is due to the difficulty of generating curved meshes in order to fully use the benefits of very high order numerical schemes. However, adaptation is possible by maintaining the classical framework for anisotropic simplex mesh adaptation [70]. This will be the basis of the present work.

The main motivation comes from the application side with the emergence of natural high order numerical methods such as the discontinuous Galerkin method [10], the residual distribution schemes [1], the CENO2 scheme [23]. Theoretically, these methods allow to increase the order easily. So, having an adaptation method which corresponds to the resolution order seems to be necessary in order to better distribute the degrees of freedom [8].

3.1.2 State of the art

For a general theoretical study, Huang [56] defines generic estimates to control interpolation order. These works are derived from the theory of interpolation in Finite Elements. However, such an approach is quite difficult to implement in practice. Indeed, the metric used to approximate a k-linear form is based on a term to term control of the Hessian of each partial derivative. Thus, for each partial derivative, a metric is derived and the final metric is the intersection of all these metrics. This idea is similar to the one introduced by Hecht [53] for second-order Lagrangian interpolations whose aim is to control the error according to each partial derivative. According to the procedure of metric intersection used, the uniqueness of the solution is not guaranteed. And even worse, in the case of intersections based on the simultaneous reduction, the resulting metric tends to become an isotropic metric, losing all the interest of anisotropic estimates.

The approach proposed by Cao is based on an analytical development of the error in 2D. Cao generalizes the study developed in the linear interpolation case [21] to higher-order case [22]. However, the set of parameterizations introduced are hardly possible in 3D. We can cite the work of Mirebeau whose idea is similar to the one of Cao but with improving results on higher-order interpolations [74]. But once again, the parameterizations developed in 2D haven't been extended to 3D yet. We can also cite the work of Yano and Dermofal from MIT who proposed higher order mesh adaptation applied to a high order discontinuous Galerkin finite element method [91, 92]. Their work is brilliant but realizes only in 2D. So what about the 3D? We can Recently, Hecht and Kuate [54] proposed a new approach to approximate anisotropic metrics from third-order interpolation error in 2D. This approach consists in solving a non linear optimization problem based on the discretization of the isoline 1 of the error function. But, by analyzing this method, we deduced that extending it to 3D and using a simple resolution algorithm will lead to a high complexity. Thus, a reformulation of this problem will be necessary to solve it.

3.1.3 Proposed approach

This chapter addresses the generalization of the results introduced in the linear interpolation case [69] to higher-order interpolations. We start our study by extending the geometric interpretation of mesh adaptation for the linear case as done in [69].
Our approach is based on a generic error model which is a homogeneous polynomial of degree k in any dimension. Then, we seek for a quadratic definite positive form approaching the variations of the initial error model. To do so, we study two methods with respect to geometric principles deduced from the linear case. The first one is based on a min-max optimization problem and the second one uses the diagonalization of symmetric tensors or of homogeneous polynomials presented in **Chapter 2**. We compare these methods to validate our approach and to state the best possible approach to work with higher-order interpolations.

3.2 GEOMETRIC PRINCIPLES FOR METRIC-BASED ADAPTATION

3.2.1 Local error model

The main difficulty in the case of higher-order adaptation is the multiplication of the number of possible interpolations. To overcome this difficulty, it is necessary to find a generic error model that will factorize a large number of interpolations: Lagrangian interpolation, Hermitian interpolation, ... Our approach is based on the space of homogeneous polynomials of degree $k \geqslant 2$.
We focus on the 3D case (the reasoning is the same in the 2D case). In a vicinity of a point $\mathbf{a} = (a_1, a_2, a_3) \in \mathbb{R}^3$, we assume that the error is well represented by a homogeneous polynomial H_e of degree $k \geqslant 2$ in three variables if the interpolation used is of

degree $k-1$. The polynomial H_e has:

$$N_k = \binom{k+n-1}{n-1}$$

coefficients. We can write it under the following form:

$$H_e(\mathbf{a}, x, y, z) = \sum_{i=0}^{k} \sum_{j=0}^{k-i} c_{ij}(x-a_1)^i (y-a_2)^j (z-a_3)^{k-i-j}.$$

For the sake of simplicity and without loss of generality, we assume that the point $\mathbf{a}$ is the origin of our coordinate system, i.e., $\mathbf{a} = (0,0,0)$.
The function H_e allows us to estimate the error along an edge $\mathbf{ax}$ issued from $\mathbf{a}$. If $\mathbf{ax} = (x,y,z)$, then this error is simply given by the function Φ_{H_e}:

$$\Phi_{H_e}(\mathbf{x}) = |H_e(\mathbf{a}; x, y, z)|.$$

This function Φ_{H_e} generalizes the computation of lengths in H_e. If H_e is a definite positive quadratic form, then Φ_{H_e} is the square length of edge $\mathbf{ax}$. Using the homogeneity of H_e, we can deduce the error anywhere from the error defined on the unit ball of any given norm $||.||$. Indeed, for any point $\mathbf{x} = (x,y,z) \neq \mathbf{a}$, as $\mathbf{x_0} = \frac{\mathbf{x}}{||\mathbf{x}||}$ belongs to the unit ball of the norm $||.||$, we have:

$$H_e(\mathbf{a}; \frac{x}{||\mathbf{x}||}, \frac{y}{||\mathbf{x}||}, \frac{z}{||\mathbf{x}||}) = \frac{1}{||\mathbf{x}||^k} H_e(\mathbf{a}; x, y, z) \Longrightarrow \Phi_{H_e}(\mathbf{x}) = ||\mathbf{x}||^k \Phi_{H_e}(\mathbf{x}_0).$$

Example 3.1
We give some examples of errors that can be written as homogeneous polynomials.

- Given a regular function u, we can derive a truncation error estimate using a k^{th} order Taylor development of u. In this case, we get an error that is written as a homogeneous polynomial. Its coefficients c_{ij} are given by:

$$c_{ij} = \frac{1}{k!} \frac{\partial^k u}{\partial x^i \partial y^j \partial z^\ell}(\mathbf{a}), \text{ with } \ell = k - i - j.$$

- For less regular functions, the error can be also written as a homogeneous polynomial [41]. The coefficients are often integrated on an element K.

$$||u - \Pi_h u||_{\mathbf{W}^{m,q}(K)} \leqslant |K|^{\frac{1}{q}-\frac{1}{p}} \sum_{\alpha_1+\alpha_2+\alpha_3=\ell-m} h_1^{\alpha_1} h_2^{\alpha_2} h_3^{\alpha_3} \left| \frac{\partial^{\ell-m} u}{\partial x^{\alpha_1} \partial y^{\alpha_2} \partial z^{\alpha_3}} \right|_{\mathbf{W}^{m,q}(K)},$$

with Π_h an interpolation operator, $|K|$ the volume of K, h_1, h_2 and h_3 the sizes along directions given by the technique of the reference element. These inequalities generally depend on some hypothesis on the powers of Sobolev spaces $\mathbf{W}^{m,q}$,

$\mathbf{W}^{m,p}$ and geometric characteristics of K. We can notice that the right-hand side of this inequality is a homogeneous polynomial of degree $\ell - m$ in three variables $(h_i)_{i=1,2,3}$, with its coefficients given by:

$$c_{ij} = \left| \frac{\partial^k u}{\partial x^i \partial y^i \partial z^\ell} \right|_{\mathbf{W}^{m,q}(K)}, \text{ with } \ell = k - i - j.$$

- The last example comes from the approximation of surfaces [45]. In this case, the approximation error of the surface is written as a homogeneous polynomial of degree 2.

3.2.2 Geometric principles for higher-order interpolation error estimates

In this section, we extend the geometric principles deduced from the study of the linear interpolation error [69] to higher-order interpolation errors in 2D and also in 3D. In the linear case, the error model is a quadratic form. Thus, to generalize the method used in the linear case to the higher-order case, we seek for **approximating locally the variations of $|H_e|$ by a quadratic definite positive form $Q(H_e)$ or $Q(d^{(k)}(H_e))$** (it refers to the quadratic form associated to the third-order derivatives of H_e) taken at power $\frac{k}{2}$,

$$|H_e(\mathbf{x})| \leqslant ({}^t\mathbf{x}\, Q(H_e)(\mathbf{x})\, \mathbf{x})^{\frac{k}{2}}. \tag{3.1}$$

Otherwise, we seek for the optimal local metric $Q(H_e)$ whose unit ball is the maximum area ellipse (resp. maximum volume ellipsoid) included in the isoline 1 (resp. the isosurface 1).

To do so, we first recall the geometric principles for the linear interpolation. And then, we propose a generalization of these principles for the higher-order case.

Geometric principles for the linear case

A fully geometric vision of the linear interpolation error is proposed in [69]. It has been shown that the local optimal metric is the one whose unit ball is included in the isoline 1 of the quadratic form ${}^t\mathbf{x}\, H(u)\, \mathbf{x}$, with $H(u)$ the Hessian of the solution u. To guarantee the uniqueness in the linear case, this optimal metric must satisfy the following geometric principles:

1. **Consistency** : this principle ensures that the quadratic model $Q(H_e)$ is an upper bound of the absolute value of the homogeneous polynomial.

2. **Maximize the volume** : this principle is related to the order of mesh convergence. Indeed, to minimize the error threshold, we have to minimize the number of elements. Locally, it is equivalent to maximize the volume of the unit ball of the metric.

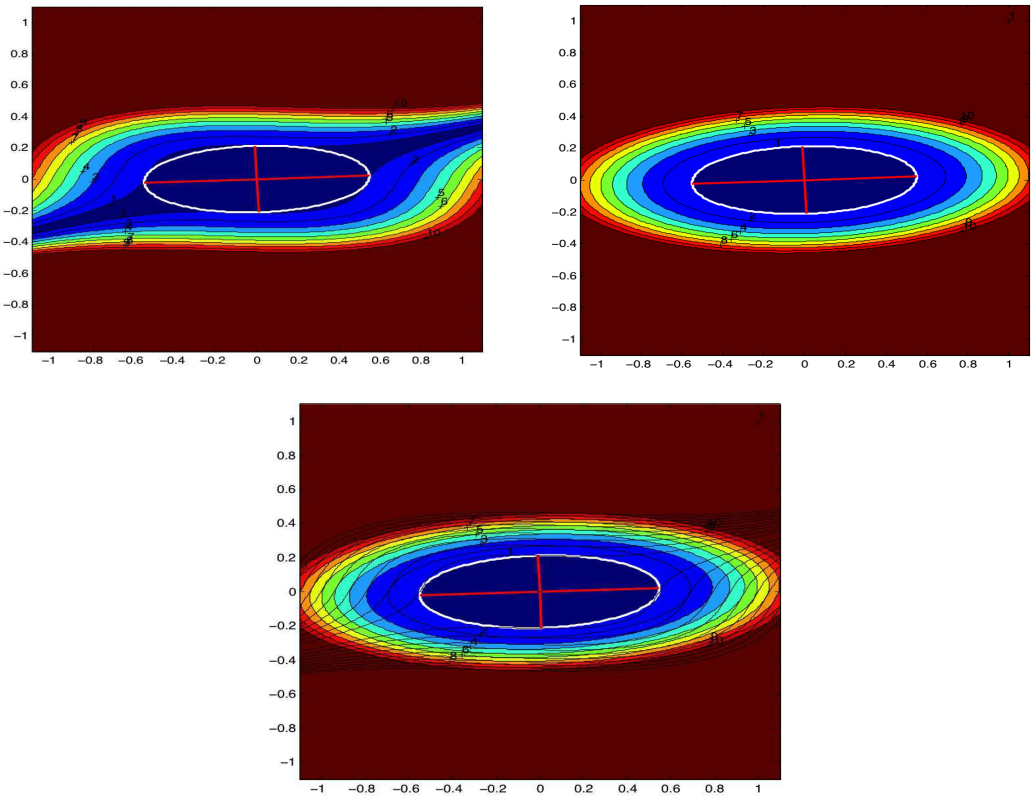

Figure 3.1: (Top, left) Representation of the isolines of $H_e(x, y)$ and the optimal local metric $Q(H_e)$ (white) included in the isoline 1. (Top, right) Representation of the isolines of the approximate error $({}^t\mathbf{x}\,Q(H_e)\,\mathbf{x})^{\frac{3}{2}}$ and the optimal local metric $Q(H_e)$ (white). (Bottom) Superposition of the isolines of $H_e(x, y)$ and $({}^t\mathbf{x}\,Q(H_e)\,\mathbf{x})^{\frac{3}{2}}$ with the optimal local metric

3. **Maximize the smallest size** : this principle avoids getting aligned with infinite branches

From linear to higher-order interpolation

We propose a geometric study of higher-order interpolation errors based on the geometric principles set out in the linear case. For a homogeneous polynomial H_e of degree k, we approach the variations of H_e by the quadratic variations given by a quadratic form Q. To compare these variations of degree 2 and k, we have to normalize them. In this case, we define distance functions:

- Φ_Q is the distance function issued from the metric defined by Q and given by:

$$\Phi_Q(\mathbf{x}) = \sqrt{{}^t\mathbf{x}\,Q\,\mathbf{x}}.$$

- $\Phi_{H_e}(\mathbf{x})$ is the distance function associated with the homogeneous polynomial H_e and given by:

$$\Phi_{H_e}(\mathbf{x}) = |H_e(\mathbf{a};\mathbf{x})|^{\frac{1}{k}}.$$

In the sequel, we state the geometric principles proposed to approach Φ_{H_e} by Φ_Q.

Principle 1. Consistency

Given an error threshold $\epsilon > 0$, a metric $\mathcal{M}$ is a consistent model with respect to the initial error model H_e if its distance function satisfies:

$$\{\mathbf{x} \in \mathcal{V}(\mathbf{a}) |\, \Phi_{\mathcal{M}}(\mathbf{x}) \leqslant \epsilon\} \subset \{\mathbf{x} \in \mathcal{V}(\mathbf{a}) |\, \Phi_{H_e}(\mathbf{x}) \leqslant \epsilon\}. \tag{3.2}$$

Using the homogeneity property of H_e and $\mathcal{M}$, if this condition is satisfied for $\epsilon = 1$ then this condition is satisfied everywhere. We have to notice that when we directly work with the error H_e and not with Φ_{H_e}, then the previous inclusion becomes:

$$\{\mathbf{x} \in \mathcal{V}(\mathbf{a}) |\, {}^t\mathbf{x}\, \mathcal{M}(\mathbf{a})\, \mathbf{x} \leqslant \epsilon^2\} \subset \{\mathbf{x} \in \mathcal{V}(\mathbf{a}) : |H_e(\mathbf{a}; \mathbf{x})| \leqslant \epsilon^k\}.$$

From a geometrical point of view, the inclusion above requires that the unit ball of $\mathcal{M}$ is included in the isoline 1 of H_e. Indeed, the relation (3.2) implies:

$$\forall\, \mathbf{x} \in \mathcal{V}(\mathbf{a}) \text{ such that } {}^t\mathbf{x}\, \mathcal{M}\, \mathbf{x} \leqslant 1 \text{ then } |H_e(\mathbf{a}; \mathbf{x})|^{\frac{1}{k}} \leqslant ({}^t\mathbf{x}\, \mathcal{M}\, \mathbf{x})^{\frac{1}{2}}.$$

This inequality is essential to certify that locally (i.e in the vicinity $\mathcal{V}(\mathbf{a})$ of $\mathbf{a}$), the quadratic error model

$$\forall\, \mathbf{x} \in \mathcal{V}(\mathbf{a}) : \quad \mathbf{x} \mapsto {}^t\mathbf{x}\, \mathcal{M}\, \mathbf{x},$$

is an upper bound of the error model H_e.

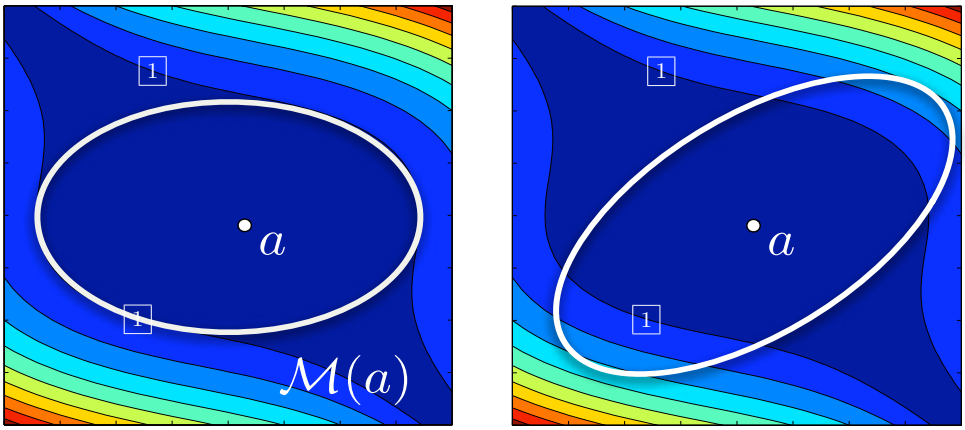

Figure 3.2: Representation of the isolines of a polynomial $H_e(\mathbf{a}; x, y)$ of degree 3. The isoline 1 is depicted by $\boxed{1}$. On the left, we depict the unit ball of a metric that satisfies the first principle and on the right, the unit ball of a metric that doesn't satisfy the first principle.

Figure 3.2 gives a geometric illustration of a consistent and an inconsistent metric for a given polynomial of degree 3. For the unit ball of the right metric, $\exists\, \mathbf{x}$ such that $\mathbf{x}\, \mathcal{M}\, \mathbf{x} = 1$ and $H_e(\mathbf{a}, \mathbf{x}) \geqslant 1$. Thus, the quadratic model is not an upper bound of the

error H_e.

Principle 2. Optimality

The first principle gives an infinity of possible solutions. Indeed, if $\mathcal{M}_0$ is a metric which satisfies the first geometric principle, then the set of metrics $\alpha\,\mathcal{M}_0$ parametrized by $\alpha > 1$ satisfies this principle too. So, we have to propose an optimality criterion such that we minimize the number of elements of the mesh for a given error threshold. Let $\mathcal{M}$ be a metric satisfying (3.2). We know that the area or volume of elements unit with respect to a metric $\mathcal{M}$ is related to the volume of the unit ball of $\mathcal{M}$. It has been proved in **Chapter 2** of [69] that the optimal continuous interpolation error in $\mathbf{L}^p$-norm depends of the complexity $\mathcal{C}(\mathcal{M})$ of the continuous mesh and is given under the following form:

$$E_p(\mathcal{M}) = \frac{Cte}{\mathcal{C}(\mathcal{M})^{\frac{k}{n}}}.$$

The complexity of the mesh is given by $\mathcal{C}(\mathcal{M}) = \int_\Omega \sqrt{\det \mathcal{M}(\mathbf{x})}\,\mathrm{d}\mathbf{x}$, with $\det \mathcal{M}(\mathbf{x})$ the volume of the unit ball $\mathcal{M}$ at point $\mathbf{x}$. To minimize the previous estimate, we can maximize the complexity $\mathcal{C}(\mathcal{M})$, which locally is equivalent to maximize the volume of $\mathcal{M}$. To conclude, the second principle consists in seeking for the maximum volume ellipsoid included in the isoline 1 of H_e. Then, given two metrics $\mathcal{M}_1$ and $\mathcal{M}_2$ which satisfy the first principle, $\mathcal{M}_1$ is a better model than $\mathcal{M}_2$ if:

$$\mathcal{M}_1 \geqslant \mathcal{M}_2 \iff \det(\mathcal{M}_2) \leqslant \det(\mathcal{M}_1). \tag{3.3}$$

Principle 3. Choice of main directions

This constraint is induced by the fact that the error H_e is not a norm, i.e., the set $\{\mathbf{x}|\, H_e(\mathbf{x}) = 0\}$ is not equal to $\{0\}$. In 2D and the linear case, i.e., $k = 2$, we can consider the hyperbolic example $H_e(x, y) = x^2 - y^2$. We seek for approaching H_e by a quadratic definite positive form. For a $r \in \mathbf{R}^*$, we have the following relations:

$$\begin{aligned}
|x^2 - y^2| &\leqslant x^2 + y^2 \\
|x^2 - y^2| &\leqslant \frac{r}{2}(x - y)^2 + \frac{2}{r}(x + y)^2,\ \forall\, r > 0 \\
|x^2 - y^2| &\leqslant \frac{r}{2}(x + y)^2 + \frac{2}{r}(x - y)^2,\ \forall\, r > 0
\end{aligned}$$

These inequalities correspond to a set of metrics $(Q(r))_{r>0}$ that satisfy the **Principles 1** and **2**. The only difference between this set of metrics is the choice of the main directions. In this case, we have to find a new principle such that associated to the previous two principles, we get the best model between all the possible one. Indeed, a part of these metrics is aligned with the directions of high gradient of H_e and the other

part is aligned with a direction of null error.

In order to have a unique solution, we propose to consider as best model the quadratic definite positive form $Q(r)$ whose smallest size is maximum. The "maximum smallest size" principle is as follow:
$\mathcal{M}_1$ is a better model than $\mathcal{M}_2$ if

$$\mathcal{M}_1 \geqslant^l \mathcal{M}_2 \iff (h_{11}, h_{12}, h_{13}) \geqslant^l (h_{21}, h_{22}, h_{23}). \tag{3.4}$$

where $\geqslant^l$ is the lexical order.

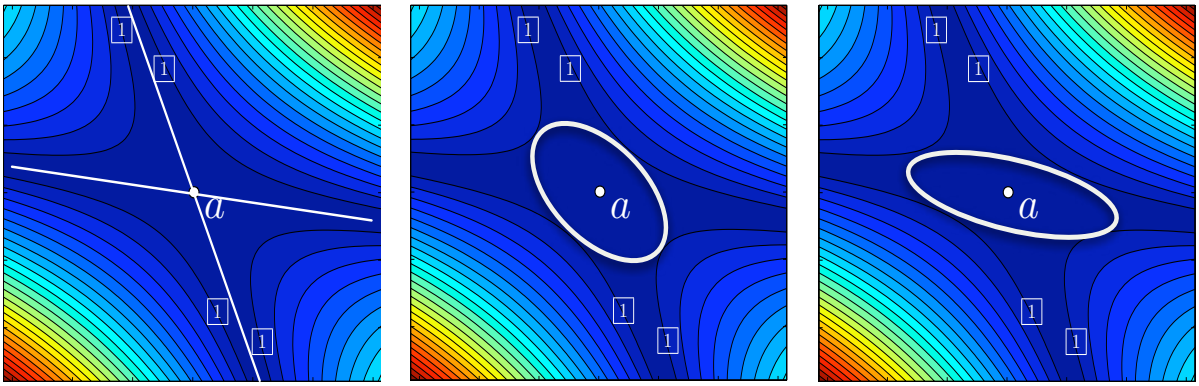

Figure 3.3: (Left) Representation of the isolines of the hyperbolic error H_e defined by its two direction of null error. (Middle, right) Choices of both possible alignments. Both metrics have the same volume and verify the first principle.

However, although this principle is a general and simple extension of Principle 3 proposed for the linear case to higher-order cases, it has to be validated. In the next section, we will show that (3.4) is too restrictive to find the maximum volume ellipse included in the isoline 1 of H_e with respect to (3.2) and (3.3).

Problematic

We seek for approximating the local optimal metric included in the isoline 1 of H_e. To do so, we consider a local optimization problem that consists in solving the following continuous system:

$$\begin{cases} \text{Min } \det(\mathcal{M}) \\ \mathcal{M} > 0 \\ \Phi_{\mathcal{M}}(\mathbf{x}) \geqslant |\Phi_{H_e}(\mathbf{x})|, \quad \forall \mathbf{x} \in \mathbb{R}^n. \end{cases} \tag{3.5}$$

This problem is a nonlinear problem with nonlinear constraints. In the 2D case, this problem is replaced by a simpler optimization problem in $\mathbb{R}^2$ by discretizing the constraints [54].

Consider the metric:

$$\mathcal{M} = \begin{pmatrix} a & \frac{c}{2} \\ \frac{c}{2} & b \end{pmatrix}, \quad a > 0,\, b > 0,\, 4\,ab - c^2 > 0.$$

Consider a discretization of the isoline 1 with n points of position (x_i, y_i), set $X_i = \binom{x_i}{y_i}$. The nonlinear problem (3.5) can be reformulate as follow:
Find three reals a, b and c that minimize $4\,ab - c^2$ under the following constraints:

$$\begin{cases} a > 0,\, b > 0, \\ a\,x_i^2 + b\,y_i^2 + c\,x_i\,y_i \geqslant 1, \quad 0 \leqslant i \leqslant n-1, \\ 4\,ab > c^2, \end{cases}$$

A naive resolution algorithm of this problem will be of $\mathcal{O}(n^3)$ complexity, which can be an expensive part of a mesh adaptation process.
In the 3D case, this worst complexity lies between $\mathcal{O}(n^3)$ and $\mathcal{O}(n^6)$. And contrary to the 2D case, a lot of discrete points needs to be used to discretize the isosurface in comparison with the 2D. Thus, a direct resolution is not possible.

In the sequel, we propose to solve an approximate problem of (3.5) to construct the optimal local metric $Q(H_e)$ in 2D and 3D with a reduced complexity.

3.3 CONSTRUCTION OF OPTIMAL LOCAL METRICS IN 2D

This section addresses the construction of optimal local metrics from higher-order interpolation error for mesh adaptation in 2D.
Based on the error model, we seek for the optimal sizes and directions of the unit ball of the optimal metric $Q(H_e)$ that satisfies the pre-cited geometric principles and Relation (3.1). To achieve this goal, several approaches have been proposed during this thesis. We expose these methods in the sequel.

3.3.1 Naive decomposition: Min-Max optimization problem

Our first approach consists in solving a sequence of optimization problems to find the quadratic definite positive form $Q(H_e)$ of maximum volume and included in the isoline 1 of H_e. This is based on Principles (3.2), (3.3) and (3.4).
First, we seek for the best isotropic model. Geometrically, it means that we seek for the maximum volume sphere included in the isoline 1 of the error function defined by the homogeneous polynomial H_e. Equivalently, this problem consists in solving a unconstrained global optimization problem of lower dimension (i.e., dimension 1 in 2D and dimension 2 in 3D).

In this case, we change the standard cartesian coordinates to polar coordinates. This allows us to parametrize the isoline or isosurface 1. We have the following change

of variables:

$$\left\{ \begin{array}{l} x = \rho\, \cos(\theta) \\ y = \rho\, \sin(\theta) \end{array} \right. \iff \left\{ \begin{array}{l} \rho = \sqrt{x^2 + y^2} \\ \theta = arctan(\frac{y}{x}) \end{array} \right. , \quad x \neq 0.$$

The error is given by:

$$H_e(\theta) = \rho^k\, P(\theta),$$

k is the order or degree of H_e and P a homogeneous polynomial depending on $cos(\theta)$ and $sin(\theta)$. Then, the equation of the isoline 1 is given by:

$$\rho = |P(\theta)|^{-\frac{1}{k}}.$$

Therefore, the unit ball $\mathcal{B}_{\mathcal{M}}$ of a metric in dimension 2 is defined by:

$$\left(\begin{array}{c} x \\ y \end{array} \right) = \left(\begin{array}{c} h_1\, \cos(\theta) \\ h_2\, \sin(\theta) \end{array} \right), \theta \in [0, 2\pi].$$

We give a 2D example to illustrate the sequence of problems solved to find the optimal local metric. We consider the following error model of degree $k = 3$:

$$H_e(x, y) = 5(x^3 + 2\, x^2 y + x\, y^2 - 2\, y^3).$$

In the polar system, we have:

$$P(\theta) = 5(cos(\theta)^3 + 2\, cos(\theta)^2 sin(\theta) + cos(\theta)\, sin(\theta)^2 - 2\, sin(\theta)^3).$$

To find the maximum error direction, we seek for the maximum on the isoline 1. We consider the polynomial $P(\theta)^{\frac{1}{k}}$. However, thanks to the homogeneity property of the polynomial, we simply consider $P(\theta)$ and seek for the value of θ that maximizes P. Numerically, we obtain the maximum error direction $\theta_{max} = 4.8133$. This direction corresponds to an error level given by $\rho_{max} = |P(\theta_{max})|^{-\frac{1}{k}}$.

Then, the maximum volume isotropic metric included in the isoline 1 is given by:

$$\mathcal{M}_{iso} = \mathcal{R}(\theta_{max}) \left(\begin{array}{cc} \lambda_{max} & \\ & \lambda_{max} \end{array} \right) {}^t\mathcal{R}(\theta_{max}),$$

with:

$$\mathcal{R} = \left(\begin{array}{cc} cos(\theta_{max}) & -sin(\theta_{max}) \\ cos(\theta_{max}) & sin(\theta_{max}) \end{array} \right), \quad \lambda_{max} = \frac{1}{\rho_{max}^2}.$$

We derive an anisotropic error estimate from the isotropic metric by "inflating" it in the orthogonal direction perpendicular to θ_{max}. Equivalently, this problem consists in solving a Min-Max optimization problem. Indeed, we consider the following parametrized metric $\mathcal{M}(\alpha)$, $\alpha \leqslant 1$:

$$\mathcal{M}(\alpha) = \mathcal{R}(\theta_{max}) \left(\begin{array}{cc} \lambda_{max} & \\ & \alpha\lambda_{max} \end{array} \right) {}^t\mathcal{R}(\theta_{max}).$$

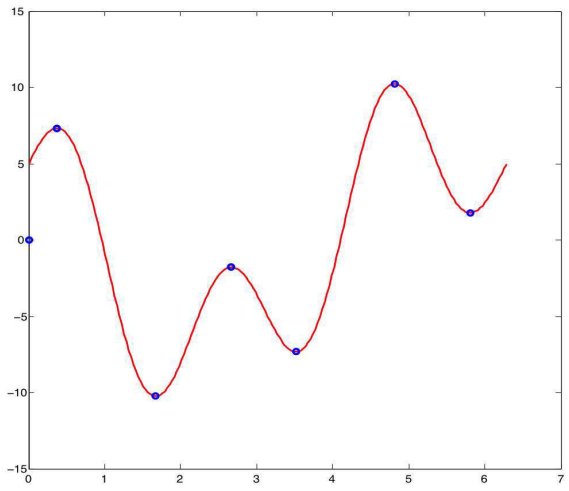

Figure 3.4: Graphical representation of the univariate function P in terms of the variable θ. The blue points are the local and global maxima and minima of P

This Min-Max optimization problem considers the values of the error on the unit ball of $\mathcal{M}$. There exists an optimum value α_{opt} of α such that the maximum value of the error is equal to 1. The orthogonal direction is given by $\theta_{max}^{opt} = \theta_{max} + \frac{\pi}{2}$. This direction corresponds to an error level given by $\rho_{max}^{opt} = |P(\theta_{max}^{opt})|^{-\frac{1}{k}}$. Then, the value of α is given by:

$$\alpha_{opt} = \frac{1}{\lambda_{max}\,(\rho_{max}^{opt})^2}.$$

Newton's method or Newton-Raphson method has been used to find successively better approximations of the roots of the function $P(\theta)$.

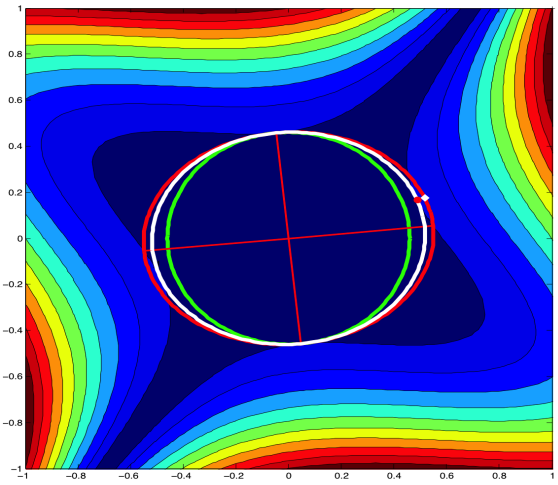

Figure 3.5: Representation of the isolines of the error model $H_e(x, y)$. Representation of the isotropic metric $\mathcal{M}_{iso}$ (green ellipse), the anisotropic metric $\mathcal{M}_{aniso}$ (red ellipse) and Corrected optimal metric $Q(H_e)$ (white ellipse) for the error model H_e.

Figure 3.5 gives an overview of the different steps to get the optimal local metric $Q(H_e)$ included in the isoline 1 of the error model H_e. First, we get the green optimal isotropic metric $\mathcal{M}_{iso}$. Then, we inflate this isotropic metric in the orthogonal direction and this gives the red local anisotropic metric $\mathcal{M}_{aniso}$. We evaluate the error on this metric:
if $\forall \mathbf{x} \in \mathcal{M}_{aniso}$, $H_e(\mathbf{x}) \leqslant 1$, this metric is the desired optimal metric.
But if $\exists \mathbf{x}_0$ (white point on the red ellipse) such that $H_e(\mathbf{x}_0) = \max\limits_{\mathbf{x} \in \mathcal{M}_{aniso}} H_e(\mathbf{x}) > 1$, we project orthogonally this point on the isoline 1 (red point on the white ellipse). Then, we get the optimal local metric centered at $\mathbf{a} = 0$ and passing through the corrected point $\mathbf{x}_0^{cor}$.

Remark 3.1 With this optimization problem, Principles (3.2) and (3.4) are satisfied. However, we will prove in the sequel that Principle (3.3) is not simultaneously satisfied with Principles (3.2) and (3.4). Indeed, we will show that maximizing the smallest size of the optimal metric sought, doesn't lead to the maximum volume anisotropic metric. More precisely, the maximum volume ellipsoid that we are seeking for is not the one that maximize the smallest size or direction. In that case, the optimal directions are not obtained.

3.3.2 Construction based on tensor decompositions

In **Chapter 2**, we exposed two symmetric tensor decomposition methods: the CP3$_{alsls}$ decomposition and Sylvester's decomposition. As we saw in this chapter, the main idea of these methods is to decompose any symmetric tensor of arbitrary degree and dimension as a sum of rank-one tensors. These methods are the extension of Singular Value Decomposition problem or diagonalization for symmetric matrices to higher-order tensors.
In this section, we show how we get the optimal local metric $Q(H_e)$ that satisfies Inequality (3.1) from the tensor decomposition of the error model H_e in 2D.

The first step consists in decomposing the homogeneous polynomial of degree 3 in two variables

$$H_e(x, y) = \sum_{i=0}^{3} \binom{3}{i} c_i \, x^i \, y^{3-i}$$

as a sum of third powers of linear terms

$$H_e(x, y) = \mu_1(\alpha_1 x + y)^3 + \mu_2(\alpha_2 x + y)^3, \tag{3.6}$$

To do so, we use one of the symmetric tensor decomposition: the CP3$_{alsls}$ decomposition or Sylvester's decomposition.

The next step consists in solving an optimization problem to get the optimal local metric $\mathcal{M}_{opt}$ from Decomposition (3.6). To do so, starting from Decomposition (3.6), we use a change of basis ${}^t P \, \mathcal{S} \, P = D$ to find a symmetric definite positive matrix

$\mathcal{S} = {}^tB\,D\,B.$

We have the new basis:

$$P = B^{-1} = \begin{pmatrix} \alpha_1 & 1 \\ \alpha_2 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} u_1 & v_1 \\ u_2 & v_2 \end{pmatrix},$$

$$D = \begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix},$$

with $\xi_1 = |H_e(u_1, u_2)|^{\frac{2}{3}} = \frac{1}{h_1^2}$ and $\xi_2 = |H_e(v_1, v_2)|^{\frac{2}{3}} = \frac{1}{h_2^2}$.

$\mathbf{u} = [u_1, u_2]$ and $\mathbf{v} = [v_1, v_2]$ are respectively the first and the second column vector of B^{-1}, the inverse matrix of B. The values of h_i are obtained by imposing the constraint $t^3\,H_e(\mathbf{u}) \leqslant 1$ and $t^3\,H_e(\mathbf{v}) \leqslant 1$, $\forall\, t > 0$.

Thus, in the new basis B^{-1}, we have:

$$\mathcal{S} = {}^t\begin{pmatrix} \alpha_1 & 1 \\ \alpha_2 & 1 \end{pmatrix}\begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix}\begin{pmatrix} \alpha_1 & 1 \\ \alpha_2 & 1 \end{pmatrix}.$$

For all $\xi_i \in \mathbb{R}^+$, $\alpha_i \in \mathbb{C}$, i=1,2, $\mathcal{S}$ is a **real symmetric matrix**.

-Indeed, if $\xi_i \in \mathbb{R}^+$, $\alpha_i \in \mathbb{R}$, $\forall\, i = 1, 2$, then:

$$\mathcal{S} = {}^t\begin{pmatrix} \alpha_1 & 1 \\ \alpha_2 & 1 \end{pmatrix}\begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix}\begin{pmatrix} \alpha_1 & 1 \\ \alpha_2 & 1 \end{pmatrix},$$

$$= \begin{pmatrix} \xi_1\,\alpha_1^2 + \xi_2\,\alpha_2^2 & \xi_1\,\alpha_1 + \xi_2\,\alpha_2 \\ \xi_1\,\alpha_1 + \xi_2\,\alpha_2 & \xi_1 + \xi_2 \end{pmatrix}.$$

-If $\xi_i \in \mathbb{R}^+$, $\alpha_i \in \mathbb{C}\backslash\mathbb{R}$, $\forall\, i = 1, 2$, as we have ${}^tB = {}^t\bar{B}$ the complex transpose of B then:

$$\mathcal{S} = {}^t\begin{pmatrix} \bar{\alpha_1} & 1 \\ \bar{\alpha_2} & 1 \end{pmatrix}\begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix}\begin{pmatrix} \alpha_1 & 1 \\ \alpha_2 & 1 \end{pmatrix}.$$

We recall that if $c = Re(c) + i\,Im(c)$ is a complex number, then the conjugate of c is $\bar{c} = Re(c) - i\,Im(c)$.

As H_e is a homogeneous polynomial with real coefficients, if the coefficients μ_i, α_i of Decomposition 3.6 are complex, then they are complex conjugates, i.e.,

$$\mu_2 = \bar{\mu_1}, \quad \alpha_2 = \bar{\alpha_1}.$$

Then,

$$\mathbf{v} = \bar{\mathbf{u}} \implies h_1 = h_2, \quad i.e., \quad \xi_1 = \xi_2.$$

In this case, $\mathcal{S}$ is written as:

$$\mathcal{S} = {}^t\begin{pmatrix} \bar{\alpha_1} & 1 \\ \alpha_1 & 1 \end{pmatrix}\begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_1 \end{pmatrix}\begin{pmatrix} \alpha_1 & 1 \\ \bar{\alpha_1} & 1 \end{pmatrix},$$

$$= \begin{pmatrix} \xi_1(\alpha_1^2 + \bar{\alpha_1}^2) & \xi_1(\alpha_1 + \bar{\alpha_1}) \\ \xi_1(\alpha_1 + \bar{\alpha_1}) & 2\,\xi_1 \end{pmatrix},$$

$$\mathcal{S} = \begin{pmatrix} 2\,\xi_1\,(Re(\alpha_1)^2 - Im(\alpha_1)^2) & 2\,\xi_1\,Re(\alpha_1) \\ 2\,\xi_1\,Re(\alpha_1) & 2\,\xi_1 \end{pmatrix},$$

where $Re(\alpha_1)$ and $Im(\alpha_1)$ are respectively the real and the complex part of the complex number α_1.

As $\mathcal{S}$ is a real symmetric matrix, it is diagonalizable with positive eigenvalues $(\lambda_i)_{i=1,2}$ and orthogonal eigenvectors $\mathcal{R}$. $\mathcal{S}$ corresponds to an ellipse whose directions are given by the eigenvectors $(\mathbf{v_i})_{i=1,2}$ of $\mathcal{R}$ and whose sizes are given by $h_i = \lambda_i^{-\frac{1}{2}}$. This ellipse is included in the isoline 1 of the error function $H_e(x, y)$ (according to Sylvester's decompostion).

From $\mathcal{S}$, we seek for the symmetric definite positive matrix $Q(H_e)$ whose unit ball $\mathcal{B}_{Q(H_e)}$ is the maximum area ellipse included in the isoline 1 of $H_e(x, y)$. To do so, we seek for a positive constant $0 < c_m \leqslant 1$ (because we seek for increasing the volume of the ellipse and being included in the isoline 1) such that:

$$Q(H_e) = c_m\,\mathcal{S} = {}^tB\begin{pmatrix} c_m\,\xi_1 & 0 \\ 0 & c_m\,\xi_2 \end{pmatrix} B,$$

and

$$Area(\mathcal{B}_{Q(H_e)}) = A(c_m) = \frac{1}{\sqrt{|\det(Q(H_e))|}},$$

is maximum.

According to the nature of the coefficients α_i (real or complex), we have the following results:

• **Real case** : α_i are real coefficients
If α_i are real coefficients, $\mathbf{c_m} = \mathbf{1}$. Then, the optimal local metric $Q(H_e)$ of maximum area included in the isoline 1 of $H_e(x, y)$ is given by:

$$Q(H_e) = {}^tB\begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix} B, \quad \text{with} \quad B = \begin{pmatrix} \alpha_1 & 1 \\ \alpha_2 & 1 \end{pmatrix}.$$

• **Complex case** : α_i are complex coefficients
If α_i are complex coefficients, $\mathbf{c_m} = \mathbf{2}^{-\frac{\mathbf{1}}{\mathbf{3}}}$. Then, the optimal local metric $Q(H_e)$ of

maximum area included in the isoline 1 of $H_e(x, y)$ is given by:

$$Q(H_e) = 2^{-\frac{1}{3}}\, {}^t\bar{B} \begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_1 \end{pmatrix} B, \quad \text{with} \quad B = \begin{pmatrix} \alpha_1 & 1 \\ \bar{\alpha_1} & 1 \end{pmatrix}.$$

The Matlab function used to compute the optimal metric $Q(H_e)$ is given in **Appendix B.3**.

Remark 3.2 The values of the constant c_m in the real and the complex case have been obtained using an iterative process. Indeed, using a Matlab function that computes the local metric S, a numerical proof shows that a variation of c_m between 0 and 1 with $Q(H_e) = c_m S$ is such that:
• in the real case, if $0 < c_m < 1$, the metric $c_m S$ is included in the isoline 1 of H_e but is not the maximum area ellipse sought. If $c_m = 1$, $c_m S$ is the desired maximum volume ellipse.
• in the complex case, if $0 < c_m < 2^{-\frac{1}{3}}$, the metric $c_m S$ is included in the isoline 1 of H_e but is not the maximum area ellipse sought. However, if $2^{-\frac{1}{3}} < c_m \leqslant 1$, the metric $c_m S$ is not included in the isoline 1 of H_e. The optimal local metric is obtained with $c_m = 2^{-\frac{1}{3}}$.

3.3.3 Two-dimensional examples

In this section, we give 2D examples of construction of optimal local metrics based on the previous methods. In these examples, we compare Sylvester's decomposition method (Sylvester), the CP3$_{alsls}$ decomposition method and the Min-Max optimization method. We study their effectiveness for the construction of optimal local metrics $Q(H_e)$ by comparing the area $A(Q(H_e))$ and the anisotropic ratio $r(Q(H_e)) = \frac{\max(h_i)}{\min(h_i)}$, with $(h_i)_{i=1,2}$ the directional sizes of the unit ball of $Q(H_e)$ obtained.

Example 3.2
We consider the homogeneous polynomial of degree 3 in two variables:

$$H_e(x, y) = 6\, x^3 - 25.782\, x^2 y - 20.94\, xy^2 - 10\, y^3.$$

We have the following results:

	Min-Max	CP3$_{alsls}$	Sylvester
$r(Q(H_e))$	1.2233	1.5950	1.5950
$A(Q(H_e))$	0.1695	0.1778	0.1778

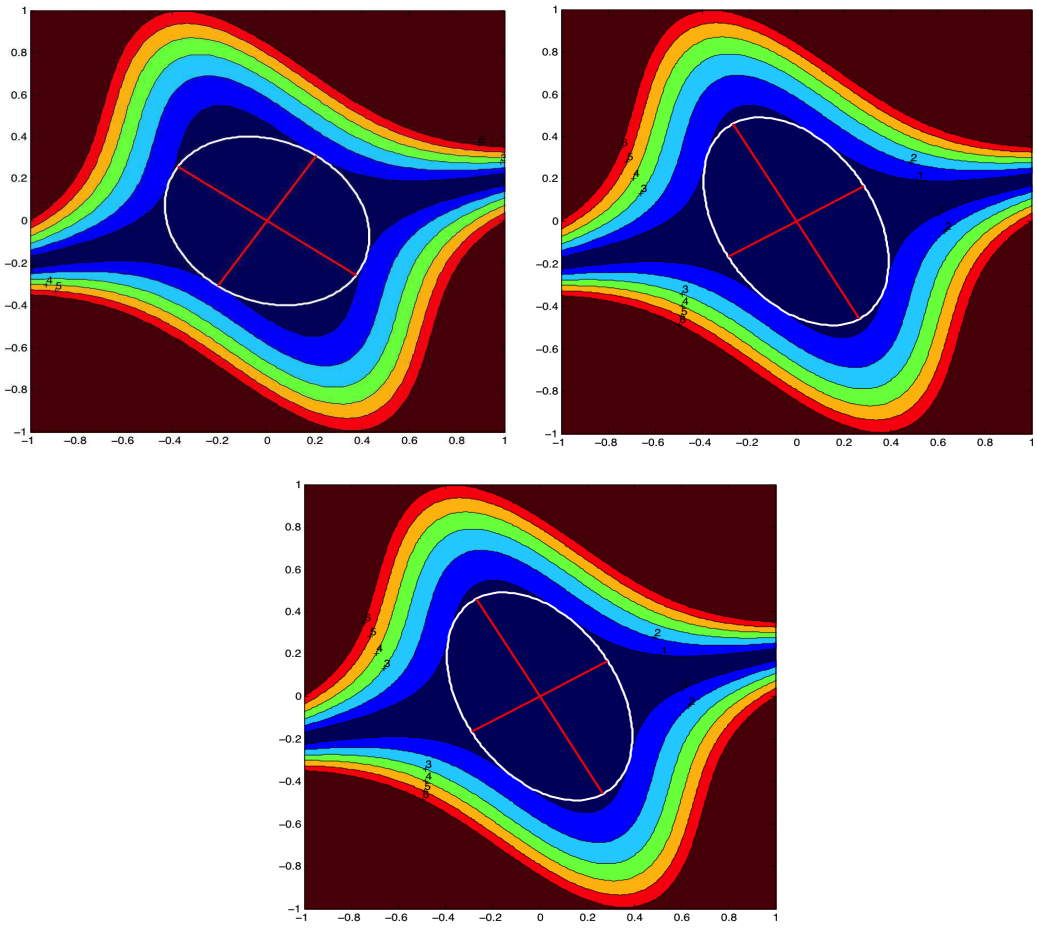

Figure 3.6: Representation of the isolines of $H_e(x,y) = 6\,x^3 - 25.782\,x^2y - 20.94\,xy^2 - 10\,y^3$ and the optimal local ellipse (white) included in the isoline 1 obtained with the Min-Max optimization method (top left), the $\mathrm{CP3}_{alsls}$ decomposition method (top right) and Sylvester's decomposition method (bottom).

Example 3.3
We consider the homogeneous polynomial of degree 3 in two variables:

$$H_e(x, y) = 2\,x^3 + 18\,xy^2.$$

We have the following results:

	Min-Max	$\mathrm{CP3}_{alsls}$	Sylvester
$r(Q(H_e))$	1.0002	1.7321	1.7321
$A(Q(H_e))$	0.2646	0.2887	0.2887

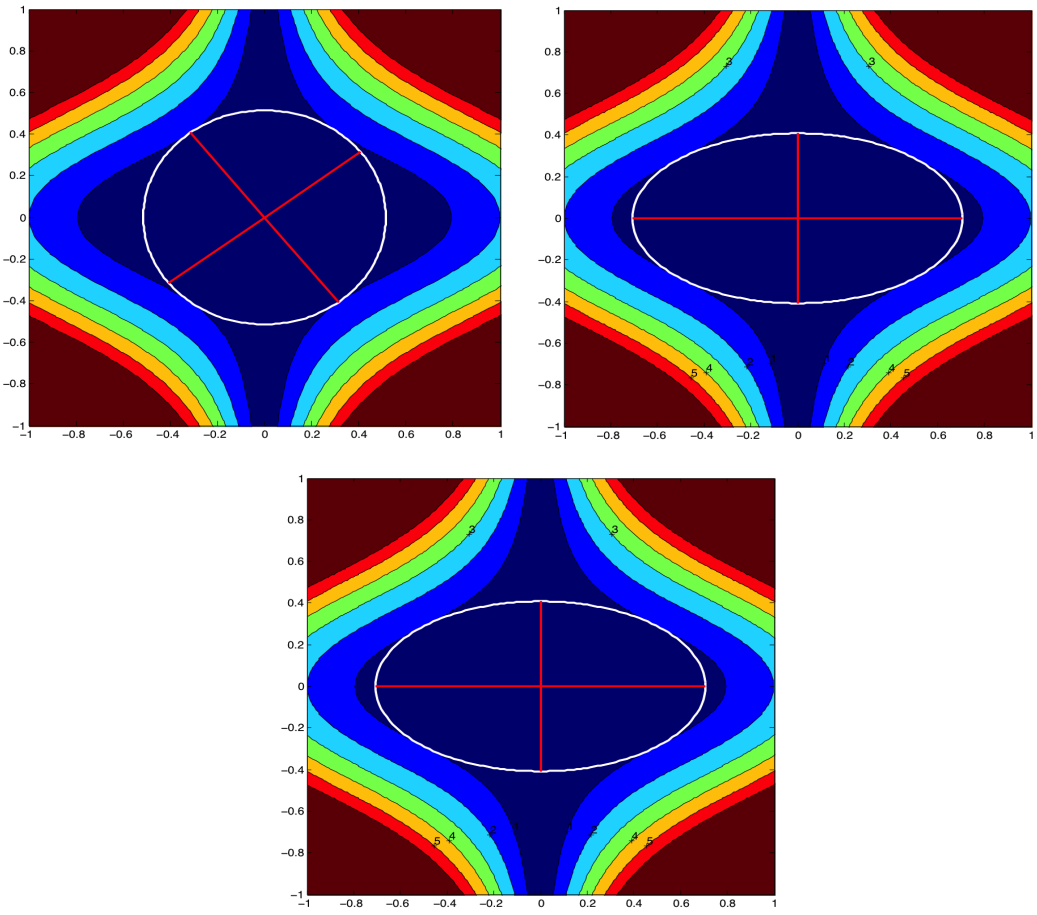

Figure 3.7: Representation of the isolines of $H_e(x, y) = 2\,x^3 + 18\,xy^2$ and the optimal local ellipse (white) included in the isoline 1 obtained with the Min-Max optimization method (top left), the $\mathrm{CP3}_{alsls}$ decomposition method (top right) and Sylvester's decomposition method (bottom).

Example 3.4
We consider the homogeneous polynomial of degree 3 in two variables:

$$H_e(x, y) = x^3 + 1.5\,x^2y + 1.5\,xy^2 + 0.5\,y^3.$$

We have the following results:

	Min-Max	$CP3_{alsls}$	Sylvester
$r(Q(H_e))$	2.0927	2.6180	2.6180
$A(Q(H_e))$	1.5122	1.5874	1.5874

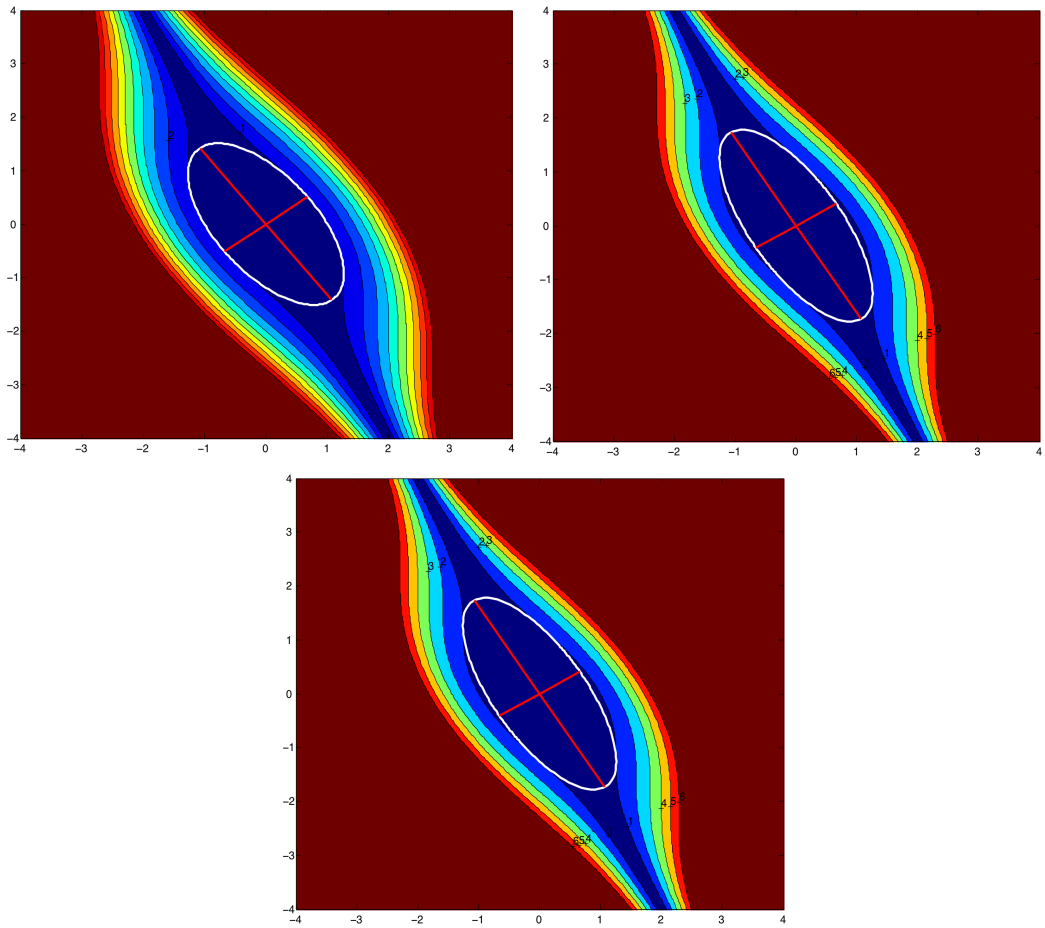

Figure 3.8: Representation of the isolines of $H_e(x, y) = x^3 + 1.5\,x^2y + 1.5\,xy^2 + 0.5\,y^3$ and the optimal local ellipse (white) included in the isoline 1 obtained with the Min-Max optimization method (top left), the $CP3_{alsls}$ decomposition method (top right) and Sylvester's decomposition method (bottom).

Example 3.5
We consider the homogeneous polynomial of degree 3 in two variables:

$$H_e(x, y) = 50\,x^3 - 360\,x^2y - 3000\,xy^2 - y^3.$$

We have the following results:

	Min-Max	$CP3_{alsls}$	Sylvester
$r(Q(H_e))$	1.0686	4.0710	4.0710
$A(Q(H_e))$	0.0092	0.0159	0.0159

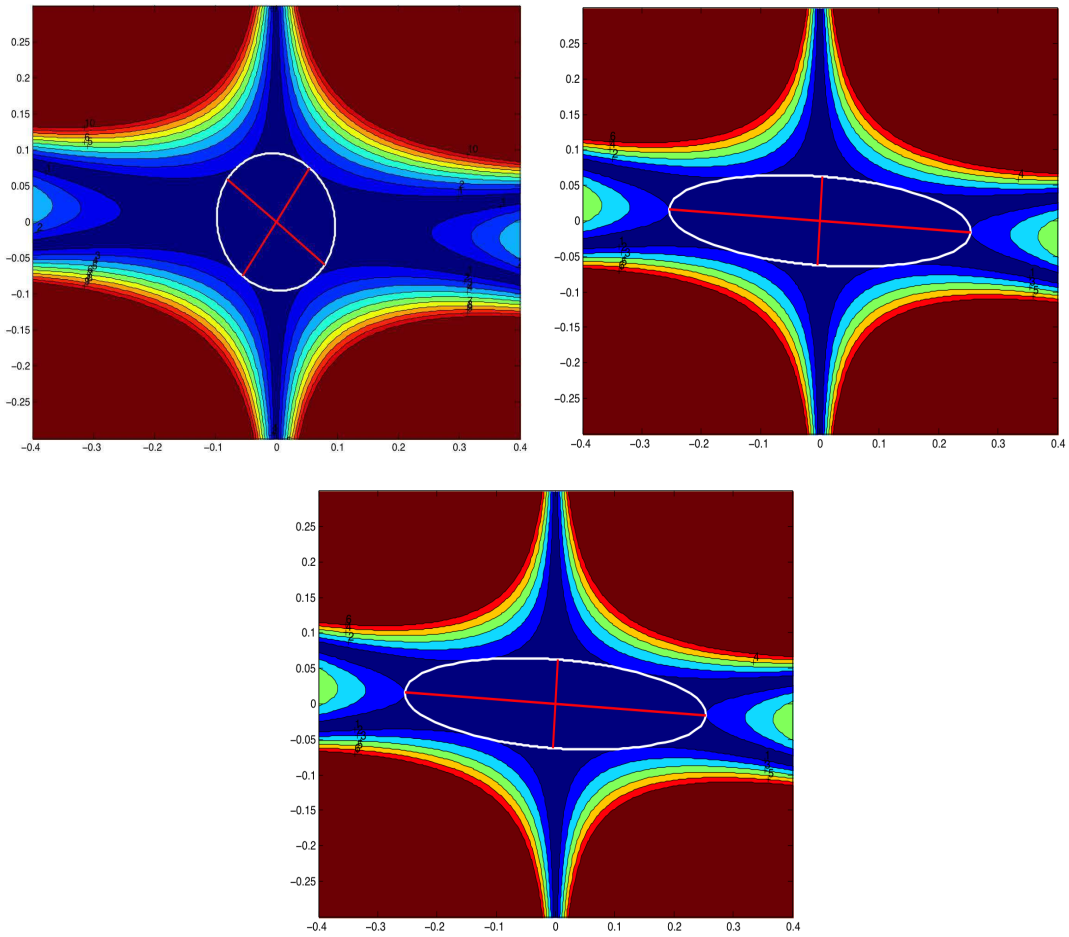

Figure 3.9: Representation of the isolines of $H_e(x, y) = 50\,x^3 - 360\,x^2y - 3000\,xy^2 - y^3$ and the optimal local ellipse (white) included in the isoline 1 obtained with the Min-Max optimization method (top left), the $CP3_{alsls}$ decomposition method (top right) and Sylvester's decomposition method (bottom).

In view of tables and results obtained in Examples 3.2, 3.3, 3.4 and 3.5, we can say that the Min-Max optimization method is far from being an optimum method. Indeed,

as you can see, comparing the anisotropic ratio $r(Q(H_e))$ and the area $A(Q(H_e))$ of the final metric, the optimal directions and sizes of the metric $Q(H_e)$ are found with the $\mathrm{CP3}_{alsls}$ decomposition and Sylvester's decomposition. Example 3.5 perfectly shows the ineffectiveness and non-optimality of the first method to catch the anisotropy of the error model H_e.

To summarize, maximizing the smallest size in order to catch the anisotropic properties of higher-order error models is a bad idea. On the contrary, using a tensor decomposition method seems to be the best solution. These methods that consist in diagonalizing the error model allow to deduce the best directions from this error model and find the optimal local metric. Note that this is the strict analogy of what is done in the linear case. Indeed, in the linear case, the diagonalization of Hessian matrix is at the core of the method while symmetric tensor diagonalization is involved for higher-order interpolation error.

3.4 CONSTRUCTION OF OPTIMAL LOCAL METRICS IN 3D

As we saw in the previous section, the $\mathrm{CP3}_{alsls}$ decomposition and Sylvester's decomposition are good methods to construct optimal local metrics from homogeneous models of order 3 in 2D. Therefore, we can legitimately wonder if using their extension in 3D, these methods will be strong enough to construct optimal local metric from error models.

3.4.1 Construction based on tensor decompositions

This section addresses the construction of optimal local metrics from higher-order interpolation error for mesh adaptation in 3D. Based on the error model, we seek for the optimal sizes and directions of the ellipsoid that satisfies the geometric principles cited in Section 3.2. To do so, we use the extension of the tensor decomposition methods exposed in the previous section to 3D: the $\mathrm{CP3}_{alsls}$ decomposition method and the symmetric tensor decomposition method.

The reasoning is the same as in the 2D case. We consider the error model H_e of degree $k = 3$ in three variables x, y, z:

$$H_e(x, y, z) = \sum_{i=0}^{3} \sum_{j=0}^{3-i} \binom{3}{i, j} a_{ij}\, x^i\, y^j\, z^{3-i-j}.$$

The first step consists in decomposing $H_e(x, y, z)$ as a sum of third powers of linear terms

$$H_e(x, y, z) = \sum_{i=1}^{r} \mu_i (\alpha_i x + \beta_i y + \gamma_i z)^3,$$

using the $\mathrm{CP3}_{alsls}$ decomposition or the 3D symmetric tensor decomposition.

But as we saw in **Table 2.1** in **Chapter 2**, a third-order homogeneous polynomial of three variables admits a generic rank equal to 4. In this case, we are expecting to get a decomposition rank $r = 4$ in general. However, getting such a decomposition that will be our basis to construct the optimal local metric included in the isoline 1 of the homogeneous error model, is neither easy nor stable.

To overcome this issue, we impose the generic rank $r = 3$ so that the dimensions (I, J, K) are equal to r as in the 2D case.

Thus, assume that the decomposition rank of any symmetric tensor $H_e(x, y, z)$ of degree 3 in three variables is equal to 3. In this case, we have:

$$H_e(x, y, z) = \sum_{i=1}^{3} \mu_i (\alpha_i x + \beta_i y + z)^3. \tag{3.7}$$

The next step consists in solving an optimization problem to get the optimal local metric $\mathcal{M}_{opt}$ from the decomposition (3.7). To do so, we are going to expose both decomposition methods separately to show the strong and weak points of each of them and try to choose the one that can be used in 3D to approximate the optimal local metric.

The CP3$_{alsls}$ decomposition in 3D

In the case of an exact symmetric tensor, i.e., the exact decomposition rank of a given third order symmetric tensor is $R = 3$, the CP3$_{alsls}$ decomposition algorithm converges to an exact solution.
On the contrary, if the symmetric tensor is non-exact, no dimension of the input tensor H_e are greater than or equal to the decomposition rank $R = 4$. In this case, all the initializations of the optimization algorithm are random. Then, the CP3$_{alsls}$ decompostion method gives a non-optimal decomposition of H_e and this decomposition is not unique. Thus, it leads to different metrics with different sizes and orientations.

But, since we assume the rank decomposition is equal to 3, a Direct Trilinear Decomposition will be used to give one good initialization and thanks to the minimization process, the optimization algorithm will "try" to find the best decomposition of the symmetric tensor. Then, we associate the corresponding polynomial form (3.7) to the final tensor decomposition. Unfortunately, the convergence of the algorithm is not always guaranteed because of the random initializations on which depends the minimization process.

Once we got the relation (3.7), a change of basis as in the 2D case, leads to an approximate construction of the maximum volume ellipsoid $Q(H_e)$ included in the isoline 1 of H_e. We will show in the next section the theoretical calculations to get the metric $Q(H_e)$ since they will be the same as we use the CP3$_{alsls}$ decomposition or the symmetric tensor decomposition in 3D.

Symmetric tensor decomposition in 3D

We present Sylvester's algorithm applied to symmetric tensors of order 3 in 3D with $R = 3$.

We consider the following general homogeneous polynomial:

$$\begin{aligned} H_e(x,y,z) &= a_{300}\,x^3 + 3\,a_{210}\,x^2y + 3\,a_{120}\,xy^2 + a_{030}\,y^3 + 3\,a_{201}x^2z + 3\,a_{102}\,xz^2 \\ &\quad + a_{003}\,z^3 + 3\,a_{021}\,y^2z + 3\,a_{012}\,yz^2 + 6\,a_{111}\,xyz. \end{aligned}$$

We recall the different steps of the symmetric tensor decomposition algorithm below (cf. **Chapter 2**, **Algorithm 2**).

-Step 1: We compute the coefficients of the dual element H_e^* in the dual basis from the coefficients of the polynomial H_e in the monomial basis.

-Step 2: We form the formal Hankel matrix associated to H_e^*, the rows and columns of which correspond to the coefficients of the polynomial H_e^* in the dual basis.

-Step 3: We extract from the formal Hankel matrix a principal minor Δ_0 of full rank $\Delta_0 \neq 0$. Assume that, in our example, the 3×3 principal minor is of full rank. We have:

$$\Delta_0 = \begin{pmatrix} a_{300} & a_{210} & a_{201} \\ a_{210} & a_{120} & a_{111} \\ a_{201} & a_{111} & a_{102} \end{pmatrix}.$$

-Step 4: We compute $\Delta_1 = \frac{y}{x}\,\Delta_0$ and $\Delta_2 = \frac{z}{x}\,\Delta_0$, the corresponding monomials of the columns of Δ_0.

$$\Delta_1 = \begin{pmatrix} a_{210} & a_{120} & a_{111} \\ a_{120} & a_{030} & a_{021} \\ a_{111} & a_{021} & a_{012} \end{pmatrix},$$

$$\Delta_2 = \begin{pmatrix} a_{201} & a_{111} & a_{102} \\ a_{111} & a_{021} & a_{012} \\ a_{102} & a_{012} & a_{003} \end{pmatrix}.$$

If the multiplication operators $\Delta_1\Delta_0^{-1}$ and $\Delta_2\Delta_0^{-1}$ commute, we go to the next step, i.e we solve the generalized eigenvalue problem $(\Delta_1 - \lambda\Delta_0)\mathbf{v} = 0$. We get the normalized eigenvectors that corresponds to the coefficients of the linear terms of the decomposition. Then, solving a linear system obtained by equating the initial polynomial and its decomposition, we determine the last unknowns of our problem.

However, if the multiplication operators $\Delta_1\Delta_0^{-1}$ and $\Delta_2\Delta_0^{-1}$ don't commute during Step 4, then, the decomposition rank is greater than 3. We go back to Step 3 and extract a new minor from the formal Hankel matrix. Thus, the principal minor Δ_0 is

given by:

$$\Delta_0 = \begin{pmatrix} a_{300} & a_{210} & a_{201} & a_{120} \\ a_{210} & a_{120} & a_{111} & a_{030} \\ a_{201} & a_{111} & a_{102} & a_{021} \\ a_{120} & a_{030} & a_{021} & h_{040} \end{pmatrix},$$

under the constraint $\Delta_0 \neq 0$. h_{040} is an unknown.

In this case, the corresponding monomials of the columns of Δ_0, $\Delta_1 = \frac{y}{x}\,\Delta_0$ and $\Delta_2 = \frac{z}{x}\,\Delta_0$ are:

$$\Delta_1 = \begin{pmatrix} a_{210} & a_{120} & a_{111} & a_{030} \\ a_{120} & a_{030} & a_{021} & h_{040} \\ a_{111} & a_{021} & a_{012} & h_{031} \\ a_{030} & h_{040} & h_{031} & h_{050} \end{pmatrix},$$

$$\Delta_2 = \begin{pmatrix} a_{201} & a_{111} & a_{102} & a_{021} \\ a_{111} & a_{021} & a_{012} & h_{031} \\ a_{102} & a_{012} & a_{003} & h_{022} \\ a_{030} & h_{031} & h_{022} & h_{041} \end{pmatrix}.$$

We have to determine the 5 unknown parameters h_{040}, h_{031}, h_{050}, h_{022} and h_{041} under the constraint:

$$\Delta_1\Delta_0^{-1}\Delta_2\Delta_0^{-1} - \Delta_2\Delta_0^{-1}\Delta_1\Delta_0^{-1} = 0. \tag{3.8}$$

The matrix relation (3.8) involves 16 polynomial equations of degree 2. Four of them are trivial. After discarding them, we finally have to solve a non-linear system of 12 polynomial equations in 5 variables $\{h_{040}, h_{031}, h_{050}, h_{022}, h_{041}\}$. Solving this system of equations is not so easy and involves the solution is not unique.
Indeed, we notice that when we use Maple software to solve this system of equations using random values for the coefficients a_{ijk} of the polynomial H_e, it leads to an infinite number of solutions. Thus, we have an infinite number of possible metrics. But all these metrics didn't have the desired sizes and orientations.
For an exact third-order polynomial, the solution is unique and all the parameters h_{040}, h_{031}, h_{050}, h_{022} and h_{041} are all equal to zero. Thus, from this solution, we get the optimal local metric.

However, since we assume that the decomposition rank is equal to 3, no parameter h_{ijk} has to be calculated, the solution of the generalized eigenvalue problem is unique even if it is generally quasi-optimal. Thus, the algorithm generally leads to a quasi-optimal tensor decomposition since the last term of the decomposition has been omitted (generic rank = 4, thus four terms in general).
In the sequel, we will show how the rank = 4 case is treated to approximate local optimal metrics.

The Matlab code used to decompose any third-order symmetric tensor of three variables is given in **Appendix B.2**.

Once we get the decomposition (3.7), the next step will consist in solving an optimization problem to get the optimal local metric $\mathcal{M}_{opt}$. To do so, we use a change of basis ${}^tP\,\mathcal{S}\,P = D$ to find a symmetric definite positive matrix $\mathcal{S} = {}^tQ\,D\,Q$.

We have the new basis:

$$P = Q^{-1} = \begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \alpha_2 & \beta_2 & 1 \\ \alpha_3 & \beta_3 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \\ u_3 & v_3 & w_3 \end{pmatrix},$$

$$D = \begin{pmatrix} \xi_1 & 0 & 0 \\ 0 & \xi_2 & 0 \\ 0 & 0 & \xi_3 \end{pmatrix},$$

with $\xi_1 = |H_e(u_1, u_2, u_3)|^{\frac{2}{3}} = \frac{1}{h_1^2}$, $\xi_2 = |H_e(v_1, v_2, v_3)|^{\frac{2}{3}} = \frac{1}{h_2^2}$ and $\xi_3 = |H_e(w_1, w_2, w_3)|^{\frac{2}{3}} = \frac{1}{h_3^2}$.

$\mathbf{u} = [u_1, u_2, u_3]$, $\mathbf{v} = [v_1, v_2, v_3]$ and $\mathbf{w} = [w_1, w_2, w_3]$ are respectively the first, the second and the third column vector of Q^{-1}, the inverse matrix of Q. The values of $(h_i)_{i=1,2,3}$ are obtained by imposing the constraint $t^3\,H_e(\mathbf{u}) \leqslant 1$, $t^3\,H_e(\mathbf{v}) \leqslant 1$ and $t^3\,H_e(\mathbf{w}) \leqslant 1$, $\forall\, t > 0$.

Thus, in the new basis Q^{-1}, we have:

$$\mathcal{S} = {}^t\begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \alpha_2 & \beta_2 & 1 \\ \alpha_3 & \beta_3 & 1 \end{pmatrix} \begin{pmatrix} \xi_1 & 0 & 0 \\ 0 & \xi_2 & 0 \\ 0 & 0 & \xi_3 \end{pmatrix} \begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \alpha_2 & \beta_2 & 1 \\ \alpha_3 & \beta_3 & 1 \end{pmatrix}.$$

For all $(\xi_i)_{i=1,2,3} \in \mathbb{R}^+$, $(\alpha_i)_{i=1,2,3}$, $(\beta_i)_{i=1,2,3} \in \mathbb{C}$, $\mathcal{S}$ is a **real symmetric matrix**.

- Indeed, if $\xi_i \in \mathbb{R}^+$, α_i, $\beta_i \in \mathbb{R}$, $\forall\, i = 1, 2, 3$, then:

$$\mathcal{S} = {}^t\begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \alpha_2 & \beta_2 & 1 \\ \alpha_3 & \beta_3 & 1 \end{pmatrix} \begin{pmatrix} \xi_1 & 0 & 0 \\ 0 & \xi_2 & 0 \\ 0 & 0 & \xi_3 \end{pmatrix} \begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \alpha_2 & \beta_2 & 1 \\ \alpha_3 & \beta_3 & 1 \end{pmatrix},$$

$$\mathcal{S} = \begin{pmatrix} \mathcal{S}(1,1) & \mathcal{S}(1,2) & \mathcal{S}(1,3) \\ \mathcal{S}(2,2) & \mathcal{S}(2,2) & \mathcal{S}(2,3) \\ \mathcal{S}(3,1) & \mathcal{S}(3,2) & \mathcal{S}(3,3) \end{pmatrix},$$

with

$$\mathcal{S}(1,1) = \alpha_1^2\,\xi_1 + \alpha_2^2\,\xi_2 + \alpha_3^2\,\xi_3, \quad \mathcal{S}(2,2) = \beta_1^2\,\xi_1 + \beta_2^2\,\xi_2 + \beta_3^2\,\xi_3,$$

$$\mathcal{S}(1,2) = \mathcal{S}(2,1) = \alpha_1\,\xi_1\,\beta_1 + \alpha_2\,\xi_2\,\beta_2 + \alpha_3\,\xi_3\,\beta_3,$$

$$\mathcal{S}(1,3) = \mathcal{S}(3,1) = \alpha_1\,\xi_1 + \alpha_2\,\xi_2 + \alpha_3\,\xi_3,$$

$$\mathcal{S}(2,3) = \mathcal{S}(3,2) = \beta_1\,\xi_1 + \beta_2\,\xi_2 + \beta_3\,\xi_3,$$

$$\mathcal{S}(3,3) = \xi_1 + \xi_2 + \xi_3.$$

-If $\xi_i \in \mathbb{R}^+$ and $\exists\,(\alpha_i, \beta_i) \in \mathbb{R}^2$, $(\alpha_{j\neq i}, \beta_{j\neq i}), (\alpha_{k\neq i}, \beta_{k\neq i}) \in \mathbb{C}^2\backslash\mathbb{R}^2$: $\alpha_k = \bar{\alpha}_j$, $\beta_k = \bar{\beta}_j$, $\{i,j,k\} \in \{1,2,3\}$.

We assume that α_1, $\beta_1 \in \mathbb{R}$ and α_2, β_2, α_3, $\beta_3 \in \mathbb{C}$ with $\alpha_3 = \bar{\alpha}_2$ and $\beta_3 = \bar{\beta}_2$.
In that case, $\mathcal{S} = {}^t\bar{Q}\,D\,Q$ becomes:

$$\mathcal{S} = {}^t\begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \bar{\alpha}_2 & \bar{\beta}_2 & 1 \\ \bar{\alpha}_3 & \bar{\beta}_3 & 1 \end{pmatrix} \begin{pmatrix} \xi_1 & 0 & 0 \\ 0 & \xi_2 & 0 \\ 0 & 0 & \xi_3 \end{pmatrix} \begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \alpha_2 & \beta_2 & 1 \\ \alpha_3 & \beta_3 & 1 \end{pmatrix},$$

$$= {}^t\begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \bar{\alpha}_2 & \bar{\beta}_2 & 1 \\ \alpha_2 & \beta_2 & 1 \end{pmatrix} \begin{pmatrix} \xi_1 & 0 & 0 \\ 0 & \xi_2 & 0 \\ 0 & 0 & \xi_3 \end{pmatrix} \begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \alpha_2 & \beta_2 & 1 \\ \bar{\alpha}_2 & \bar{\beta}_2 & 1 \end{pmatrix},$$

Since $Q = \begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \alpha_2 & \beta_2 & 1 \\ \bar{\alpha}_2 & \bar{\beta}_2 & 1 \end{pmatrix}$, then, $Q^{-1} = \begin{pmatrix} u_1 & v_1 & \bar{v}_1 \\ u_2 & v_2 & \bar{v}_2 \\ u_3 & v_3 & \bar{v}_3 \end{pmatrix}$.

Thereby, $\mathbf{w} = \bar{\mathbf{v}}$. So, $\xi_2 = |H_e(v_1, v_2, v_3)|^{\frac{2}{3}} = |H_e(w_1, w_2, w_3)|^{\frac{2}{3}} = \xi_3$.
Therefore, we have:

$$\mathcal{S} = {}^t\begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \bar{\alpha}_2 & \bar{\beta}_2 & 1 \\ \alpha_2 & \beta_2 & 1 \end{pmatrix} \begin{pmatrix} \xi_1 & 0 & 0 \\ 0 & \xi_2 & 0 \\ 0 & 0 & \xi_2 \end{pmatrix} \begin{pmatrix} \alpha_1 & \beta_1 & 1 \\ \alpha_2 & \beta_2 & 1 \\ \bar{\alpha}_2 & \bar{\beta}_2 & 1 \end{pmatrix},$$

$$\mathcal{S} = \begin{pmatrix} \mathcal{S}(1,1) & \mathcal{S}(1,2) & \mathcal{S}(1,3) \\ \mathcal{S}(2,2) & \mathcal{S}(2,2) & \mathcal{S}(2,3) \\ \mathcal{S}(3,1) & \mathcal{S}(3,2) & \mathcal{S}(3,3) \end{pmatrix},$$

with

$$\mathcal{S}(1,1) = \alpha_1^2\,\xi_1 + 2\,\xi_2\,|\alpha_2|^2, \quad \mathcal{S}(2,2) = \beta_1^2\,\xi_1 + 2\,\xi_2\,|\beta_2|^2,$$

$$\mathcal{S}(1,2) = \mathcal{S}(2,1) = \alpha_1\,\xi_1\,\beta_1 + \bar{\alpha}_2\,\xi_2\,\beta_2 + \bar{\beta}_2\,\xi_2\,\alpha_2 = \alpha_1\,\xi_1\,\beta_1 + 2\,Re(\bar{\alpha}_2\,\beta_2)\,\xi_2,$$

$$\mathcal{S}(1,3) = \mathcal{S}(3,1) = \alpha_1\,\xi_1 + \bar{\alpha}_2\,\xi_2 + \alpha_2\,\xi_2 = \alpha_1\,\xi_1 + 2\,Re(\alpha_2)\,\xi_2,$$

$$\mathcal{S}(2,3) = \mathcal{S}(3,2) = \beta_1\,\xi_1 + \bar{\beta}_2\,\xi_2 + \beta_2\,\xi_2 = \beta_1\,\xi_1 + 2\,Re(\beta_2)\,\xi_2,$$

$$\mathcal{S}(3,3) = \xi_1 + 2\,\xi_2.$$

As $\mathcal{S}$ is a real symmetric matrix, it is diagonalizable with positive eigenvalues $(\lambda_i)_{i=1,2,3}$ and orthogonal eigenvectors $\mathcal{R}$. $\mathcal{S}$ corresponds to an ellipsoid whose directions are given by the eigenvectors $(v_i)_{i=1,2,3}$ of $\mathcal{R}$ and whose sizes are given by $h_i = \lambda_i^{-\frac{1}{2}}$, $i = 1, 2, 3$. This ellipsoid is included in the isosurface 1 of the error function $H_e(x, y, z)$.

From $\mathcal{S}$, we seek for the symmetric definite positive matrix $Q(H_e)$ whose unit ball $\mathcal{B}_{Q(H_e)}$ is the maximum area ellipsoid included in the isosurface 1 of $H_e(x, y, z)$. To do so, we seek for a positive constant $c_m > 0$ such that:

$$Q(H_e) = c_m\,\mathcal{S} = {}^tQ \begin{pmatrix} c_m\,\xi_1 & 0 & 0 \\ 0 & c_m\,\xi_2 & 0 \\ 0 & 0 & c_m\,\xi_3 \end{pmatrix} Q,$$

and

$$Area(\mathcal{B}_{Q(H_e)}) = A(c_m) = \frac{1}{\sqrt{|\det(Q(H_e))|}},$$

is maximum.

Concerning the 3D case, since we take into account three linear terms in the decomposition of the error model H_e instead of the four terms needed, we decided to assume that for the real case as for the complex case, the constant $0 < c_m \leqslant 1$ we are seeking for, is equal to 1. With this assumption, the "a priori" optimal local metric sought is given

- in the **real case** by:

$$Q(H_e) = c_m\,{}^tQ \begin{pmatrix} \xi_1 & 0 & 0 \\ 0 & \xi_2 & 0 \\ 0 & 0 & \xi_3 \end{pmatrix} Q,$$

- in the **complex case** by:

$$Q(H_e) = c_m\,{}^t\bar{Q} \begin{pmatrix} \xi_1 & 0 & 0 \\ 0 & \xi_2 & 0 \\ 0 & 0 & \xi_3 \end{pmatrix} Q.$$

REMARK 3.3 In the 3D case, the decomposition of the model error has been truncated. Indeed, the last linear term has been omitted so that we get three terms and avoid the infinite number of solutions that we generally obtained in the high order case. Then, contrary to the 2D case for third-order polynomials, we can notice that the coefficient c_m will not be easy to find because of the metric S that is not necessary locally optimal and has not necessary the good directions and sizes sought.

With this estimate of the optimal local metric, we can now present 3D examples of approximation of optimal local metric of maximum volume based on the CP3_{alsls} decomposition and the 3D symmetric tensor decomposition. With these examples, we will show that the solution we proposed to reach our goal is good enough and by comparing both methods, we will prove that 3D symmetric tensor decomposition seems to be better than the CP3_{alsls} for the construction of the desired optimal metrics.

3.4.2 Three-dimensional examples

In this section, 3D examples of construction of optimal local metrics from third-order homogeneous polynomials are proposed. These approximations have been obtained using as basis the CP3_{alsls} decomposition or the 3D symmetric tensor decomposition. We compare the area $A(Q(H_e))$ and the anisotropic ratios $r_{ij}(Q(H_e)) = \frac{\max(h_i,h_j)}{\min(h_i,h_j)}$, with $(h_i)_{i=1,2,3}$ the directional sizes of the ellipsoid $Q(H_e)$. Many comments will be done to show the weak and the strong points of these two methods. This will help to know if both methods can be used or if one of them is preferable to the other one.

Example 3.6
We consider the homogeneous polynomial of degree 3 in three variables:

$$H_e(x,y,z) = 4x^3 + 9x^2y + 39xy^2 + 9y^3 - 3x^2z + 93xz^2 - 19z^3 - 39y^2z - 111yz^2 - 78xyz.$$

We have the following results:

$\mathrm{CP3}_{alsls}$			
$A(Q(H_e))$	$r_{12}(Q(H_e))$	$r_{13}(Q(H_e))$	$r_{23}(Q(H_e))$
0.0417	1.7296	3.6621	2.1172

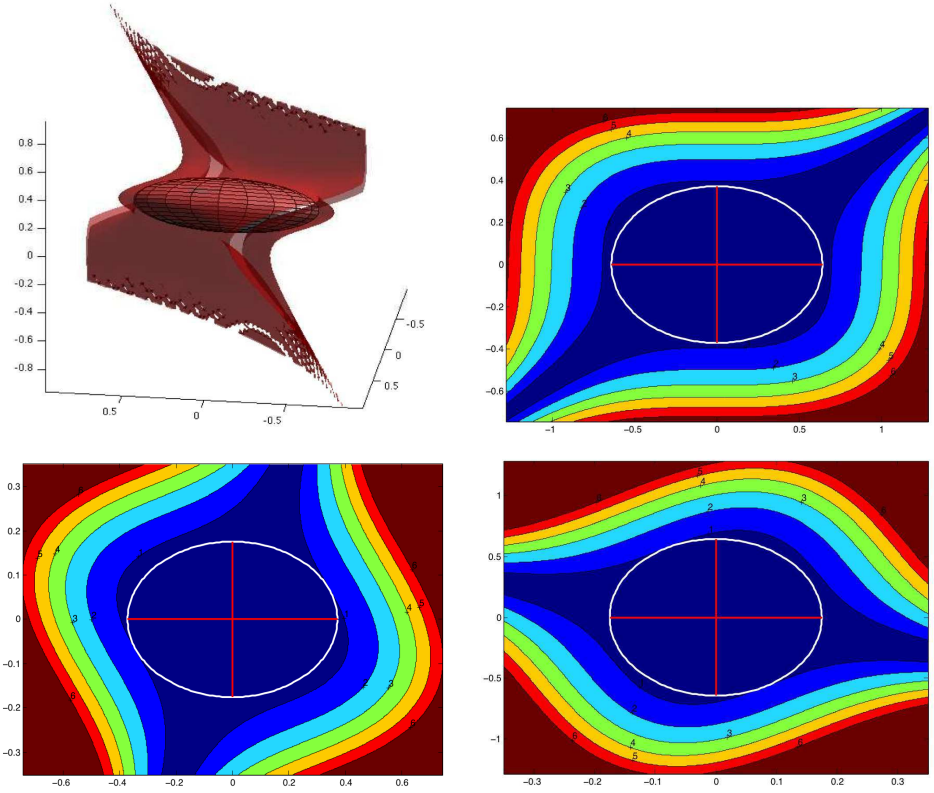

Figure 3.10: Error model $H_e(x,y,z)$. (Top left) Representation of its isosurface 1 (red color) and the corresponding maximum volume ellipsoid included. (Top right, bottom) Cross-sections along the three symmetric planes $(\mathbf{v}_1,\mathbf{v}_2)$, $(\mathbf{v}_2,\mathbf{v}_3)$, $(\mathbf{v}_3,\mathbf{v}_1)$ of the optimal metric.

Symmetric tensor decomposition			
$A(Q(H_e))$	$r_{12}(Q(H_e))$	$r_{13}(Q(H_e))$	$r_{23}(Q(H_e))$
0.0417	1.7296	3.6621	2.1172

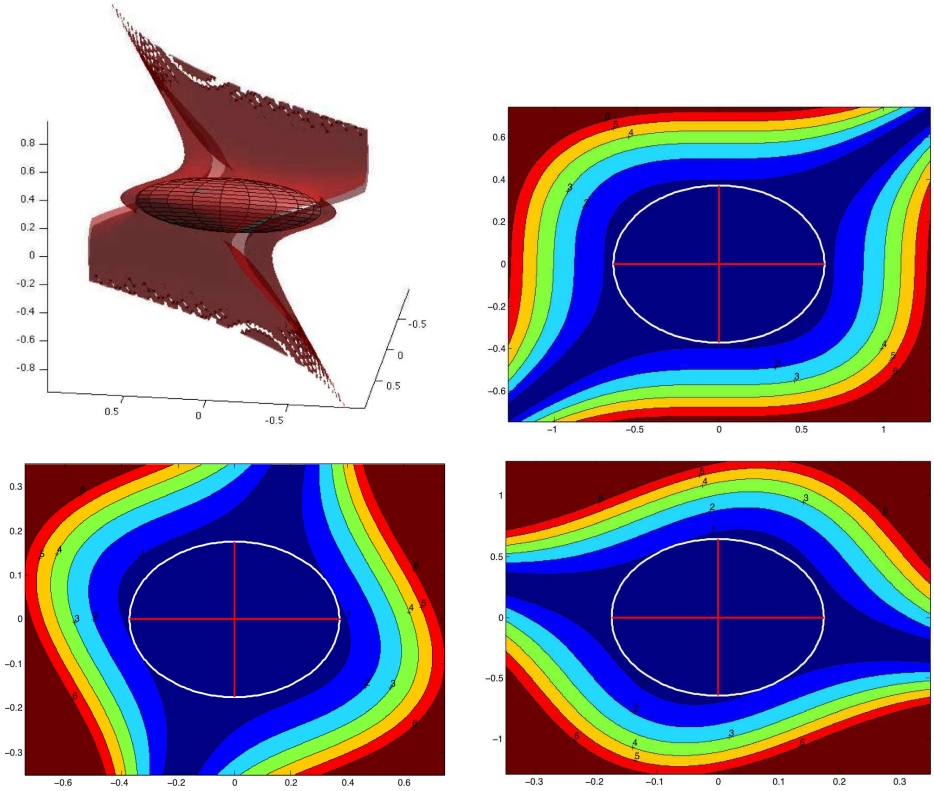

Figure 3.11: Error model $H_e(x, y, z)$. (Top left) Representation of its isosurface 1 (red color) and the corresponding maximum volume ellipsoid included. (Top right, bottom) Cross-sections along the three symmetric planes $(\mathbf{v}_1, \mathbf{v}_2)$, $(\mathbf{v}_2, \mathbf{v}_3)$, $(\mathbf{v}_3, \mathbf{v}_1)$ of the optimal metric.

The function $H_e(x, y, z)$ of Example 3.6 is a homogeneous polynomial whose decomposition involves real terms and the decomposition rank is equal to 3. In that case, the solution obtained from the CP3_{alsls} decomposition algorithm or the 3D symmetric tensor decomposition algorithm is exact and unique. We show the inclusion of the maximum volume ellipsoid in the isosurface 1. We get the same results with both methods and the algorithm needs one call ($it = 1$) of the exact line search function to find a good initialization.

EXAMPLE 3.7
We consider the homogeneous polynomial of degree 3 in three variables:

$$\begin{aligned} H_e(x,y,z) &= 2.7452\,x^3 - 1.1871\,x^2y - 0.0460\,xy^2 + 2.3475\,y^3 + 1.2465\,x^2z \\ &+ 2.3065\,xz^2 + 3.0474\,z^3 - 5.5903\,y^2z - 11.2928\,yz^2 - 12.8406\,xyz. \end{aligned}$$

We have the following results:

$\mathrm{CP3}_{alsls}$			
$A(Q(H_e))$	$r_{12}(Q(H_e))$	$r_{13}(Q(H_e))$	$r_{23}(Q(H_e))$
0.2568	2.5124	1.7831	1.4090

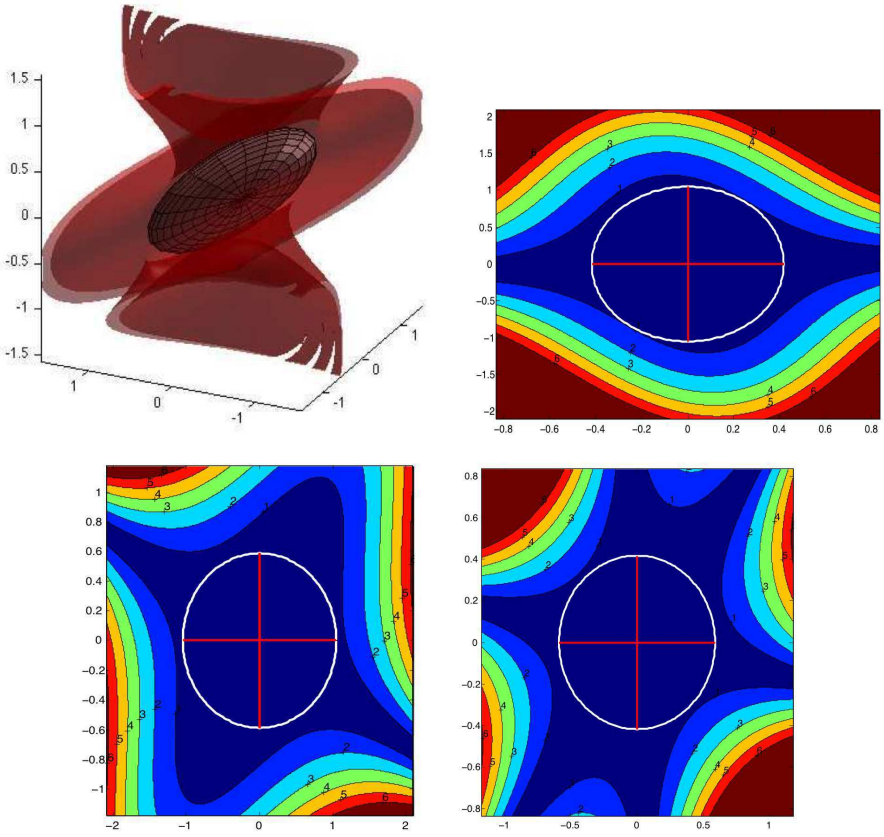

Figure 3.12: (Top left) Representation of its isosurface 1 (red color) and the corresponding maximum volume ellipsoid included. (Top right, bottom) Cross-sections along the three eigenvector-based planes $(\mathbf{v}_1, \mathbf{v}_2)$, $(\mathbf{v}_2, \mathbf{v}_3)$, $(\mathbf{v}_3, \mathbf{v}_1)$ of the optimal metric.

Symmetric tensor decomposition			
$A(Q(H_e))$	$r_{12}(Q(H_e))$	$r_{13}(Q(H_e))$	$r_{23}(Q(H_e))$
0.2559	1.9170	2.9454	1.5364

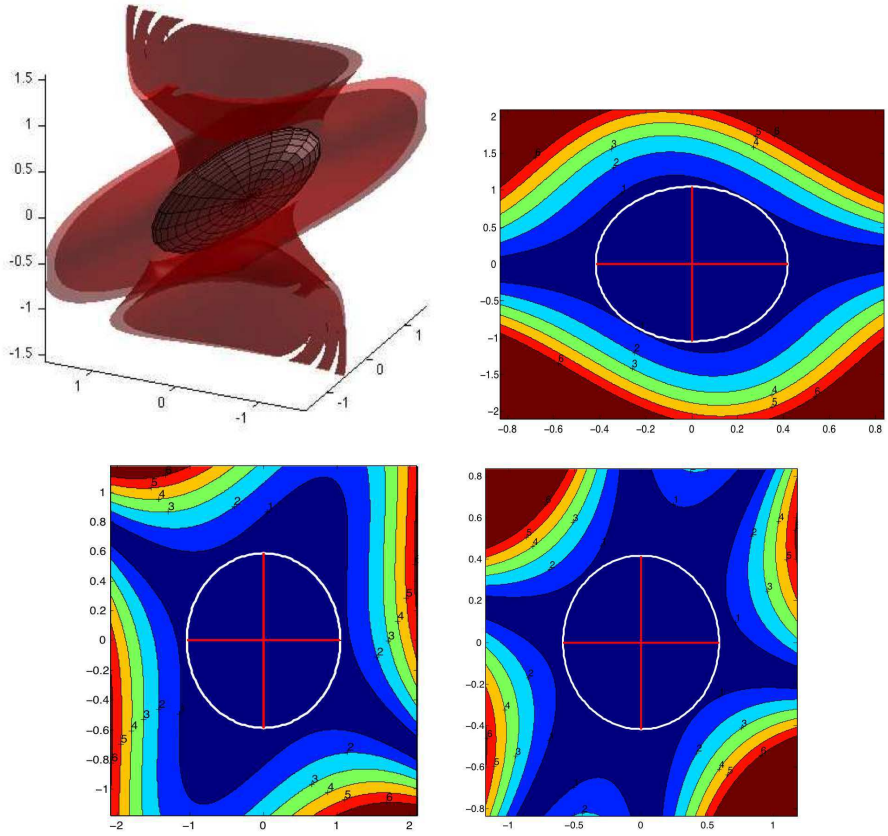

Figure 3.13: (Top left) Representation of its isosurface 1 (red color) and the corresponding maximum volume ellipsoid included. (Top right, bottom) Cross-sections along the three eigenvector-based planes $(\mathbf{v}_1, \mathbf{v}_2)$, $(\mathbf{v}_2, \mathbf{v}_3)$, $(\mathbf{v}_3, \mathbf{v}_1)$ of the optimal metric.

The function $H_e(x, y, z)$ of Example 3.7 is a homogeneous polynomial whose decomposition involves complex terms and the decomposition rank is equal to 4. Therefore, using the proposed idea to approximate the maximum volume ellipsoid included in the isosurface 1 of the function H_e, we get the results above. The CP3_{alsls} decomposition algorithm gives a better solution than the 3D symmetric tensor decomposition (good directions and sizes of the metric, better volume) but it needs a high number of iterations ($it = 66$) to find a good initialization.

EXAMPLE 3.8
We consider the homogeneous polynomial of degree 3 in three variables:

$$\begin{aligned} H_e(x,y,z) &= 5.9536\,x^3 + 0.0830\,x^2y - 1.8183\,xy^2 + 2.9305\,y^3 + 1.1994\,x^2z \\ &- 1.3740\,xz^2 + 5.8176\,z^3 + 0.5618\,y^2z - 1.0777\,yz^2 + 0.3101\,xyz. \end{aligned}$$

We have the following results:

$\mathrm{CP3}_{alsls}$			
$A(Q(H_e))$	$r_{12}(Q(H_e))$	$r_{13}(Q(H_e))$	$r_{23}(Q(H_e))$
0.2126	1.3028	1.3613	1.0448

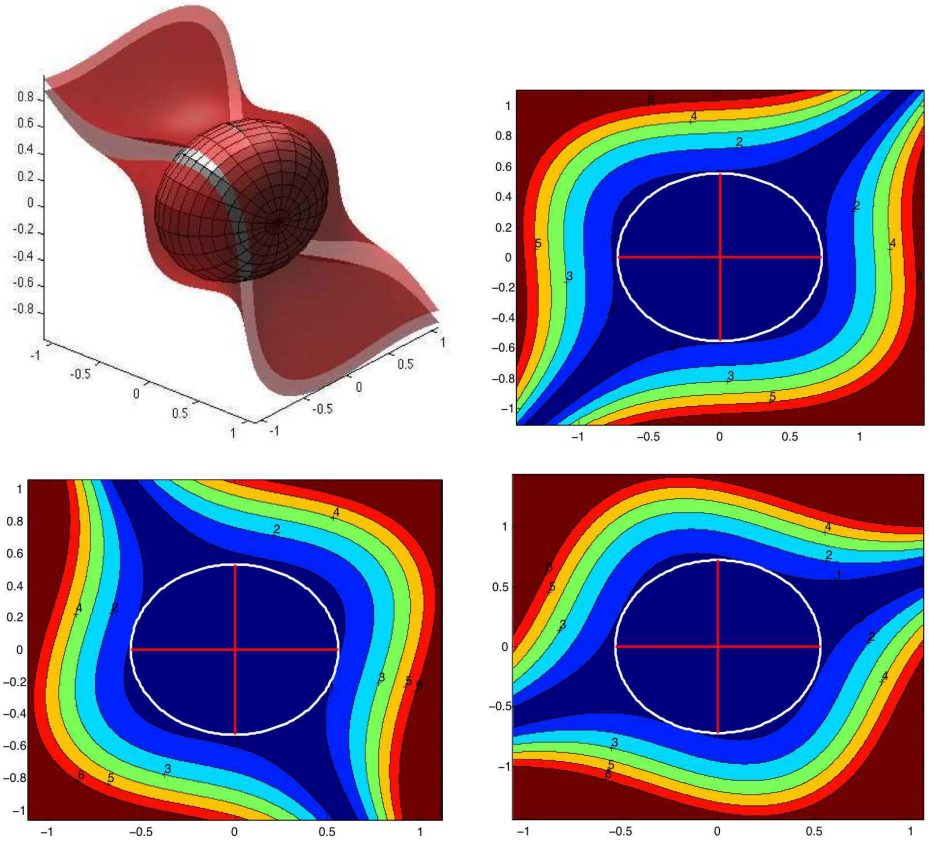

Figure 3.14: (Top left) Representation of its isosurface 1 (red color) and the corresponding maximum volume ellipsoid included. (Top right, bottom) Cross-sections along the three eigenvector-based planes $(\mathbf{v}_1, \mathbf{v}_2)$, $(\mathbf{v}_2, \mathbf{v}_3)$, $(\mathbf{v}_3, \mathbf{v}_1)$ of the optimal metric.

Symmetric tensor decomposition			
$A(Q(H_e))$	$r_{12}(Q(H_e))$	$r_{13}(Q(H_e))$	$r_{23}(Q(H_e))$
0.2127	1.3039	1.3521	1.0369

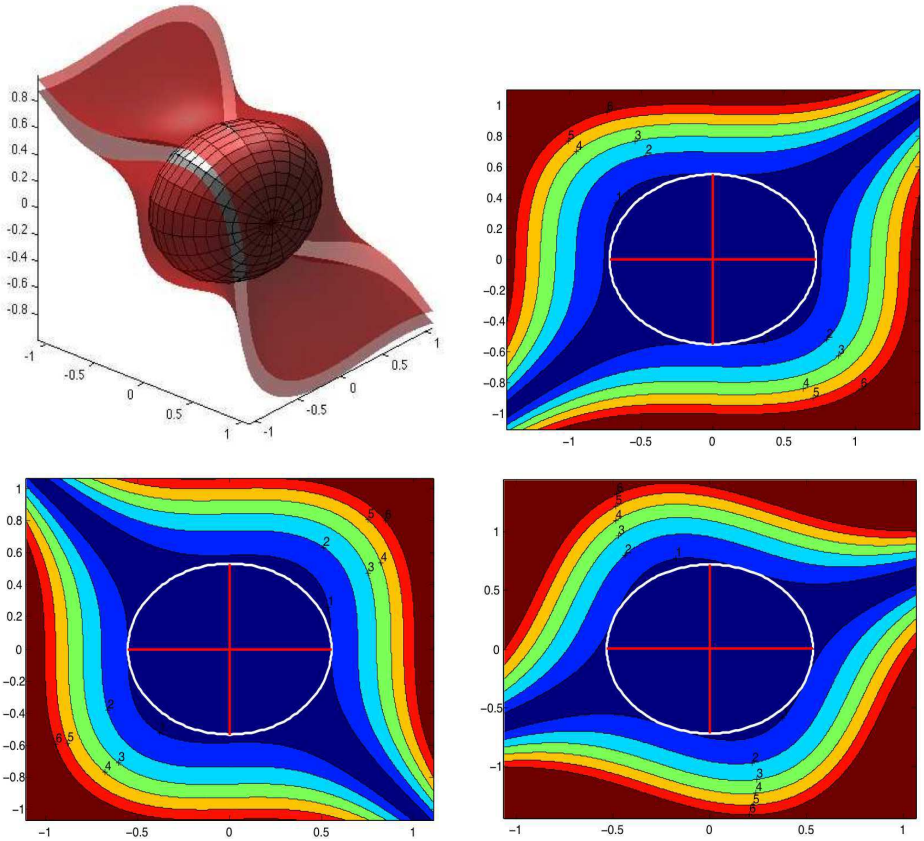

Figure 3.15: (Top left) Representation of its isosurface 1 (red color) and the corresponding maximum volume ellipsoid included. (Top right, bottom) Cross-sections along the three eigenvector-based planes $(\mathbf{v}_1, \mathbf{v}_2)$, $(\mathbf{v}_2, \mathbf{v}_3)$, $(\mathbf{v}_3, \mathbf{v}_1)$ of the optimal metric.

The function $H_e(x, y, z)$ of Example 3.8 is a homogeneous polynomial whose decomposition involves reals coefficients and the decomposition rank is equal to 4. Contrary to the previous example, the 3D symmetric tensor decomposition is better than the CP3_{alsls} decomposition. It leads to the best maximum volume ellipsoid included in the isosurface 1.

EXAMPLE 3.9
We consider the homogeneous polynomial of degree 3 in three variables:

$$\begin{aligned} H_e(x,y,z) &= 2.5197\,x^3 + 0.8041\,x^2y + 4.0012\,xy^2 + 2.8799\,y^3 - 17.9290\,x^2z \\ &- 3.1526\,xz^2 + 2.8379\,z^3 + 2.0450\,y^2z - 10.7623\,yz^2 - 6.9838\,xyz. \end{aligned}$$

We have the following results:

$\mathrm{CP3}_{alsls}$			
$A(Q(H_e))$	$r_{12}(Q(H_e))$	$r_{13}(Q(H_e))$	$r_{23}(Q(H_e))$
0.1850	2.1591	2.3419	1.0846

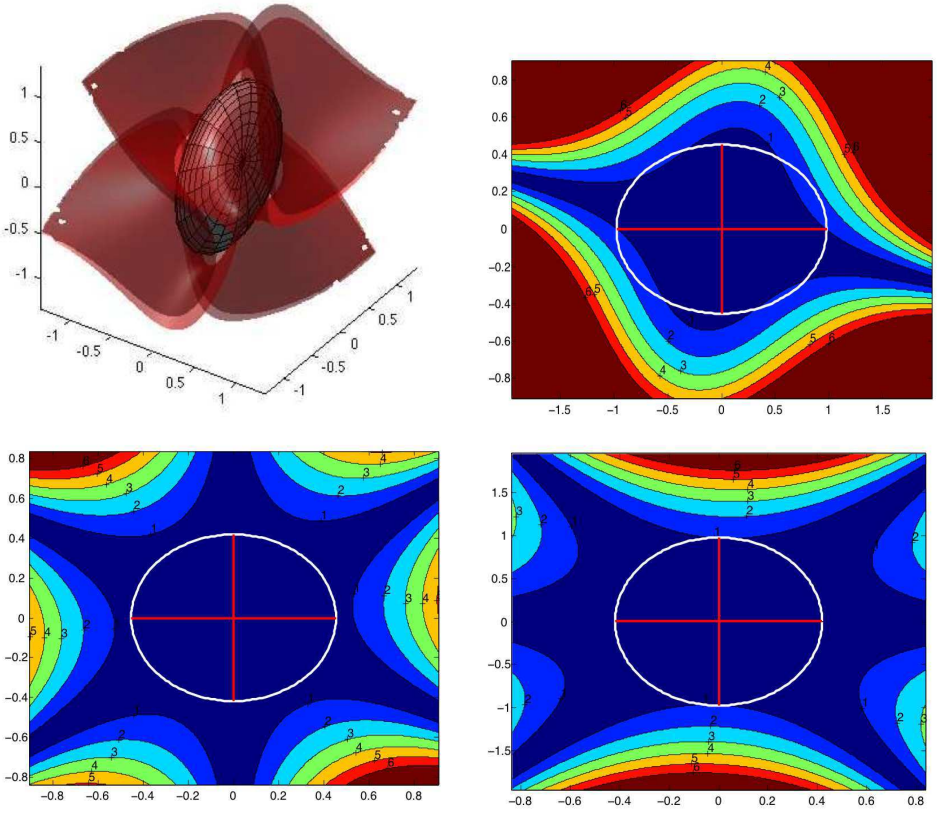

Figure 3.16: (Top left) Representation of its isosurface 1 (red color) and the corresponding maximum volume ellipsoid included. (Top right, bottom) Cross-sections along the three eigenvector-based planes $(\mathbf{v}_1, \mathbf{v}_2)$, $(\mathbf{v}_2, \mathbf{v}_3)$, $(\mathbf{v}_3, \mathbf{v}_1)$ of the optimal metric.

Symmetric tensor decomposition			
$A(Q(H_e))$	$r_{12}(Q(H_e))$	$r_{13}(Q(H_e))$	$r_{23}(Q(H_e))$
0.1513	1.8747	2.5574	1.3641

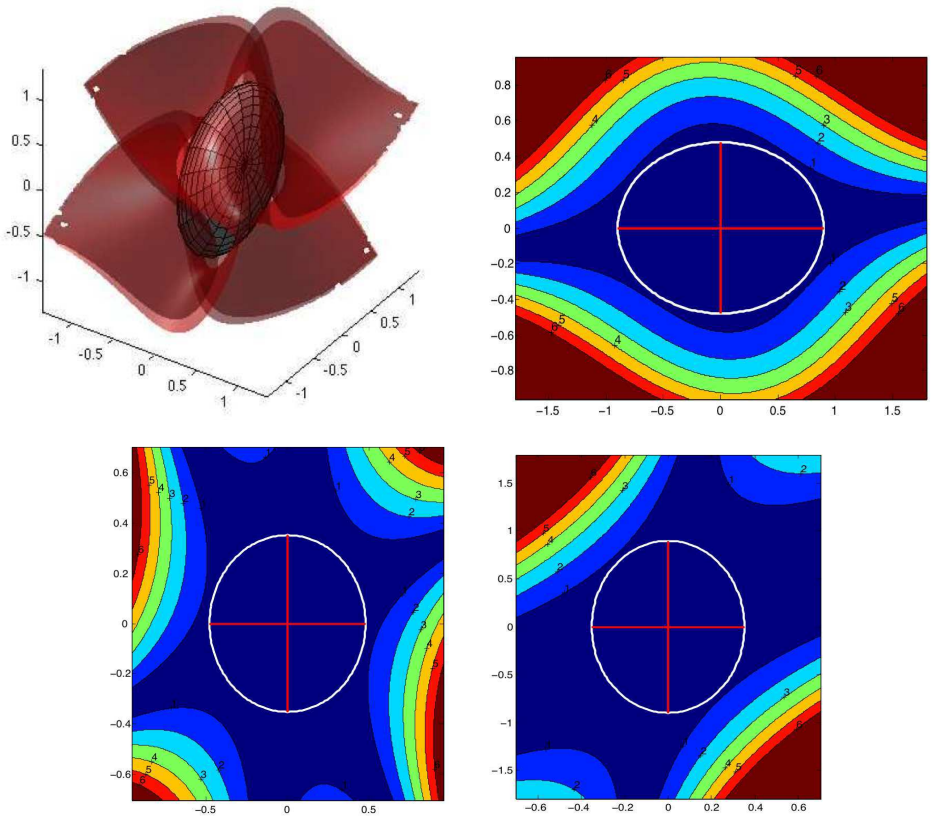

Figure 3.17: (Top left) Representation of its isosurface 1 (red color) and the corresponding maximum volume ellipsoid included. (Top right, bottom) Cross-sections along the three eigenvector-based planes $(\mathbf{v}_1, \mathbf{v}_2)$, $(\mathbf{v}_2, \mathbf{v}_3)$, $(\mathbf{v}_3, \mathbf{v}_1)$ of the optimal metric.

The function $H_e(x, y, z)$ of Example 3.9 is a homogeneous polynomial whose decomposition involves complex coefficients and the decomposition rank is equal to 4. As we can see in **Figure 3.16**, the final metric obtained from the CP3$_{alsls}$ decomposition algorithm is not verifying the first criterion i.e., the ellipse is not included in the isoline 1 of H_e. Its directions are not good, then the metric is out of the isosurface 1. The algorithm needs 32 iterations of the exact line search to find an initialization but it didn't converge to a good decomposition. On the contrary, the 3D symmetric tensor decomposition is better and leads to a better approximation of the desired optimal metric.

EXAMPLE 3.10
We consider the homogeneous polynomial of degree 3 in three variables:

$$\begin{aligned} H_e(x,y,z) &= 2\,x^3 + 0.015\,x^2y + 18\,xy^2 + y^3 + 1.08\,x^2z + 2.7\,xz^2 + z^3 + 3\,y^2z \\ &+ 30\,yz^2 + 18\,xyz. \end{aligned}$$

We have the following results:

$\mathrm{CP3}_{alsls}$			
$A(Q(H_e))$	$r_{12}(Q(H_e))$	$r_{13}(Q(H_e))$	$r_{23}(Q(H_e))$
0.0683	8.7394	2.4857	3.5157

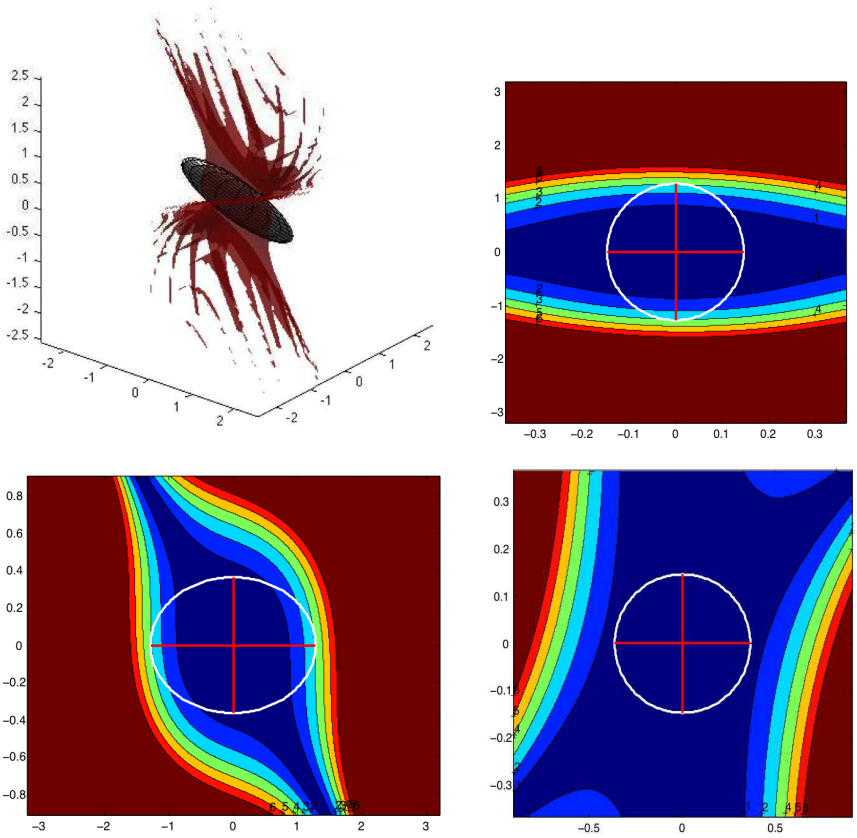

Figure 3.18: (Top left) Representation of its isosurface 1 (red color) and the corresponding maximum volume ellipsoid included. (Top right, bottom) Cross-sections along the three eigenvector-based planes $(\mathbf{v}_1, \mathbf{v}_2)$, $(\mathbf{v}_2, \mathbf{v}_3)$, $(\mathbf{v}_3, \mathbf{v}_1)$ of the optimal metric.

Symmetric tensor decomposition			
$A(Q(H_e))$	$r_{12}(Q(H_e))$	$r_{13}(Q(H_e))$	$r_{23}(Q(H_e))$
0.1201	2.8434	2.2834	1.2452

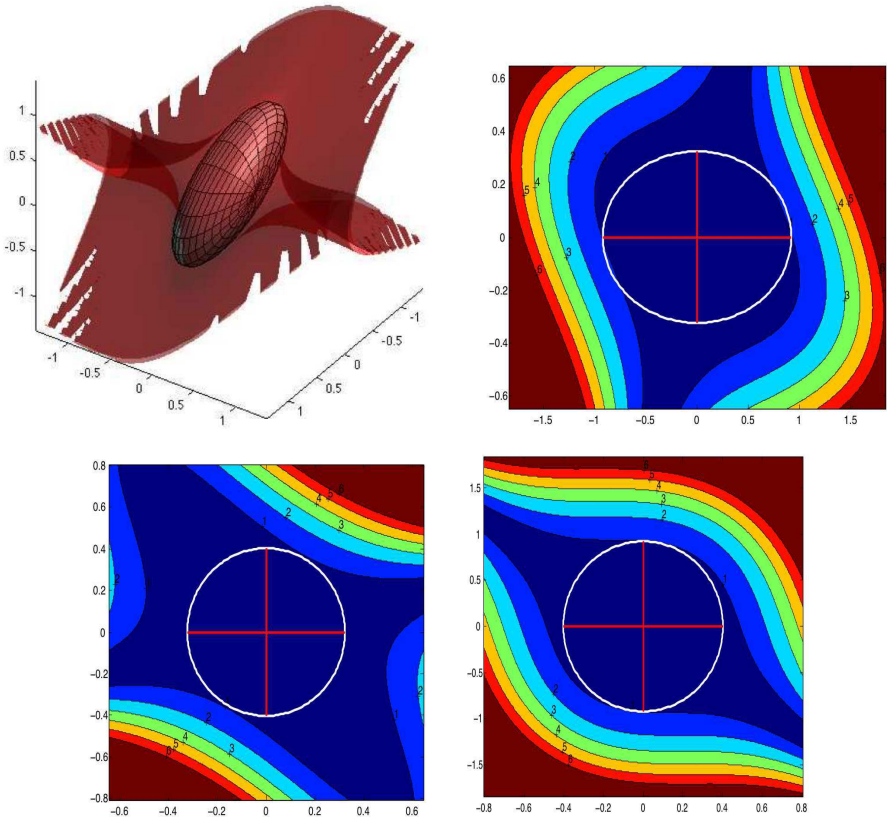

Figure 3.19: (Top left) Representation of its isosurface 1 (red color) and the corresponding maximum volume ellipsoid included. (Top right, bottom) Cross-sections along the three eigenvector-based planes $(\mathbf{v}_1, \mathbf{v}_2)$, $(\mathbf{v}_2, \mathbf{v}_3)$, $(\mathbf{v}_3, \mathbf{v}_1)$ of the optimal metric.

Example 3.10 is a good example to show which of two methods seems to be better than the other to lead a good approximation of the desired optimal local metric. $H_e(x, y, z)$ of Example 3.10 is a homogeneous polynomial whose decomposition involves complex coefficients and the decomposition rank is equal to 4. When we use the CP3_{alsls} decomposition, at the end of the given number of iterations, the algorithm is not able to give a good initialization of the loading matrices. Then, as you can see it in **Figure 3.18**, the algorithm didn't converge to a correct solution. The resulting numeric ellipsoid violates the isoline criterion. On the contrary, as we can see in **Figure 3.19**, the 3D symmetric tensor decomposition gives a better solution even if the metric is not totally included in the isosurface 1 of the function.

3.5 CONCLUSION

In this chapter, we exposed three methods to find the optimal local metric of maximum volume included in the isoline 1 (resp. the isosurface 1) of an interpolation error model belonging to the set of homogeneous polynomial of order 3 in two variables (resp. three variables). We deduce the following remarks:

- The Min-Max optimization problem is not a good way to approximate the desired optimal metric from the error model. Indeed, maximizing the smallest size of the ellipsoid doesn't mean maximizing its volume.

- Tensor decomposition seems to be a good basis to construct optimal local metrics included in isoline or isosurface 1. However, the study of two methods: the $CP3_{alsls}$ decomposition and Sylvester's decomposition and its extension to higher dimensions shows that one of them is more stable than the other one and quasi-optimal.
  Indeed, the $CP3_{alsls}$ decomposition algorithm needs good initializations of the loading matrices to converge to good solutions. But this iterative process can be very long (high number of iterations) and the use of random initializations may lead to a non-convergence of the algorithm.
  On the contrary, Sylvester's decomposition and the 3D symmetric tensor decomposition have shown to be good methods to reach our goal in 2D and 3D respectively. However, even if in the 3D case particularly, the symmetric decomposition method presents some insufficiencies that lead to the construction of local metrics not enough optimal, this can be corrected.

Thereby, in view of the analysis made in this chapter, 2D and 3D analytical examples of mesh adaptation that are presented in the next chapter will be based on the binary decomposition algorithm and the 3D symmetric tensor decomposition respectively.

4

Higher-order mesh adaptation

Contents

4.1 INTRODUCTION

In its more general form, the problem of mesh adaptation consists in finding the mesh $\mathcal{H}$ of a domain Ω that minimizes a given error for a given function u. Mesh adaptation to control the linear interpolation error $u - \Pi_h u$ in $\mathbf{L}^p$-norm has been studied in many "pioneering" and recent works. This problem is stated in an *a priori* way:

$$\text{Find } \mathcal{H}_{opt} \text{ having } N \text{ nodes such that } E(\mathcal{H}_{opt}) = \min_{\mathcal{H}} ||u - \Pi_h u||_{\mathbf{L}^p(\Omega_h)}. \qquad (\mathrm{P}^1)$$

(P^1) is a global combinatorial problem that is purely intractable practically. Indeed, this would require the simultaneous optimization of both the mesh topology and the vertices location, a problem which cannot be considered. Moreover, the problem is ill-posed as several optimal meshes can be found for a single function u. Consequently, simpler problems are considered to approximate the solution. A common simplification is to perform a local analysis of the error instead of considering the global problem. A first set of methods consists in deriving a local bound of the optimal element shape [21, 37]. A second set consists in deriving a local bound of the interpolation error. This bound is then transformed into a metric-based estimate [41, 56, 79]. Direct minimization of the error can also be considered by using directly the interpolation error as a cost function in the mesh generator [61]. All these strategies have in common the resolution of a local problem as they act in the vicinity of an element. Consequently, such error minimizations are equivalent to a steepest descent algorithm that converges only to a local minimum with poor convergence properties. This drawback arises because a minimization on a discrete mesh is directly considered.
The resolution of (P^1) in a continuous setting has been proposed in [69]. Thus, (P^1) is recast as a continuous optimization problem where the discrete interpolation error is replaced by the continuous one:

$$\text{Find } \mathbf{M}_{opt} \text{ having a complexity } N \text{ such that } E_p(\mathbf{M}_{opt}) = \min_{\mathcal{M}} ||u - \pi_{\mathcal{M}} u||_{\mathbf{L}^p(\Omega)}.$$

Contrary to discrete-based studies, the continuous formulation succeeds in solving globally the optimal interpolation error problem by using powerful mathematical tools such as calculus of variations.

In this chapter, we study the extension of this continuous concept to higher-order case. Thus, we consider the problem of finding the optimal mesh that globally minimizes the $\mathbf{L}^p$-norm of the higher-order interpolation error of a continuous function. This problem is stated in the following a priori way:

$$\text{Find } \mathcal{H}_{opt} \text{ having } N \text{ nodes such that } E_p(\mathcal{H}_{opt}) = \min_{\mathcal{H}} ||u - \Pi_h^k u||_{\mathbf{L}^p(\Omega_h)}. \qquad (\mathrm{P}^k)$$

where Π_h^k is the k^{th}-order discrete interpolate of u, i.e., it is piecewise k^{th}-order representation of u on a mesh. It is equal to u on each node of the mesh and of order k inside each element.

As in the linear case, (P^k) is also intractable practically. Thanks to the success of the continuous mesh framework in the linear case, we propose to address the resolution of P^k in a continuous setting. In this case, (P^k) is recast as a continuous optimization problem where the discrete error $e_h = |u - \Pi_h^k u|$ is replaced by the continuous one $e_{\mathcal{M}}$. The mesh $\mathcal{H}$ is replaced by its continuous representation $\mathbf{M}$ as introduced in **Chapter 1**.

The higher-order error model $e_{\mathcal{M}}$ depends on the higher-order derivative matrix $d^{(k)}(u)$ of the numerical solution u on each node of the mesh, contrary to the linear case where the interpolation error depends on the local Hessian matrix. Thus, the resolution of the global continuous problem for a higher-order error model $e_{\mathcal{M}}$ is based on the local optimization problem we solved in **Chapter 2**.

In the quadratic case, the well-posed global optimization problem of finding the optimal continuous mesh minimizing the third-order continuous interpolation error $e_{\mathcal{M}}$ in $\mathbf{L}^p$-norm is:

$$\text{Find } \mathbf{M}_{opt} = \min_{\mathcal{M}} E_p(\mathcal{M}) \quad = \quad \left(\int_\Omega |e_{\mathcal{M}}(\mathbf{x})|^p \, \mathrm{d}\mathbf{x} \right)^{\frac{1}{p}}, \tag{4.1}$$

with

$$e_{\mathcal{M}} = d^{(3)}(u)(\mathcal{M}^{-\frac{1}{2}}\mathbf{x}).$$

Thanks to the quadratic model of the third-order derivatives of u, $Q(d^{(3)}(u))$, the previous equation simplifies to:

$$E_p(\mathcal{M}) = \left(\int_\Omega \mathrm{trace}\left(\mathcal{M}^{-\frac{1}{2}} \, Q(d^{(3)}(u))(\mathbf{x}) \, \mathcal{M}^{-\frac{1}{2}} \right)^{\frac{3p}{2}} \, \mathrm{d}\mathbf{x} \right)^{\frac{1}{p}},$$

under the constraint $\mathcal{C}(\mathcal{M}) = \displaystyle\int_\Omega d(\mathbf{x}) \, \mathrm{d}\mathbf{x}$.

We start this chapter with the resolution of the third-order global optimization problem (4.1). Then, we present a classical technique used to recover the third-order derivatives of u on each node of the mesh. We end by analytical examples based on multi-scale third-order interpolation error.

4.2 MULTI-SCALE MESH ADAPTATION

In this section, we address the resolution of the global optimization problem (4.1). We prove the uniqueness of the optimal solution and estimate the order of convergence. This section summarizes and extends the results of multi-scale mesh adaptation obtained in [69].

4.2.1 Global generic optimization problem

We first address the resolution of a generic metric-based error problem. We will specifically consider the quadratic interpolate case later in the sequel.

For a given function u defined in $\mathbb{R}^n$, we consider the following generic local error model on a continuous mesh $\mathcal{M}$:

$$e_{\mathcal{M}} = c_n \left(\sum_{i=1}^{n} h_i^{\beta} \gamma_i \right)^{\alpha}, \tag{4.2}$$

where $\alpha > 0$ and $\beta > 0$ are parameters, $\gamma_i > 0$ depends on the error estimate, thus on the quadratic form Q and on the orientations of $\mathcal{M}$.

The problem (4.1) is reformulated as follow:

$$\text{Find} \quad \mathbf{M}_{opt} = \min_{\mathcal{M}} E_p(\mathcal{M}) = \left(\int_{\Omega} e_{\mathcal{M}}^p \right)^{\frac{1}{p}} = \left(\int_{\Omega} \left(\sum_{i=1}^{n} h_i^{\beta} \gamma_i \right)^{\alpha p} \right)^{\frac{1}{p}}, \tag{4.3}$$

under the constraint

$$\mathcal{C}(\mathcal{M}) = \int_{\Omega} d = N.$$

The constraint on the complexity is added to avoid the trivial solution where all $(h_i)_{i=1,...,n}$ are zero which provides a null error. We use a calculus of variations to globally solve (4.3).

4.2.2 Global optimality principle

Optimization problem (4.3) is solved for the subset of continuous meshes having the same fixed $(\gamma_i)_{i=1,...,n}$, i.e, we seek for the optimal sizes $(h_i)_{i=1,...,n}$, solutions of (4.3), the $(\gamma_i)_{i=1,...,n}$ being fixed. The resolution is based on a change of variables which involves the density d and the anisotropic quotients $(r_i)_{i=1,...,n}$ of the metric. The change of variables is then given by:

$$h_i = d^{-\frac{1}{n}} r_i \text{ for } i = 1, ..., n-1 \text{ and } h_n = d^{-\frac{1}{n}} P^{-1},$$

where

$$ri = h_i \left(\prod_{j=1}^{n} h_j \right)^{-\frac{1}{n}} \text{ and } P = \left(\prod_{i=1}^{n-1} r_i \right).$$

With this new set of unknowns, the function $e_{\mathcal{M}}$ locally writes:

$$e_{\mathcal{M}} = d^{-\frac{\alpha\beta}{n}} \left(\sum_{i=1}^{n-1} r_i^{\beta} \gamma_i + P^{-\beta} \gamma_n \right)^{\alpha}.$$

Thus, we have to solve:

$$\min_{(r_i)_i, d} \int_{\Omega} d^{\frac{-\alpha\beta p}{n}} \left(\sum_{i=1}^{n-1} r_i^{\beta} \gamma_i + P^{-\beta} \gamma_n \right)^{\alpha p},$$

under the linear constraint:

$$\int_\Omega d = N. \tag{4.4}$$

One main consequence of considering d as an unknown is to have now a linear constraint, so that Problem (4.3) becomes convex. This change of variables also leads to an uncoupled problem: the optimal anisotropic quotients $(r_i)_{i=1,\dots,n}$ are first exhibited and the optimal density is derived in a second step.

The classical Euler-Lagrange necessary condition states that the variation of E_p at point $\mathcal{M}$ in the direction $\delta\mathcal{M}$ is proportional to the variation of the constraint $\mathcal{C}$ in the neighborhood of a critical point. As we use a formal approach, the variation of E_p is approximated by:

$$\delta E_p(\mathcal{M};\delta\mathcal{M}) = \lim_{\epsilon\to 0}\frac{1}{\epsilon}\left(\int_\Omega e^p_{\mathcal{M}+\epsilon\,\delta\mathcal{M}} - \int_\Omega e^p_{\mathcal{M}}\right) \approx \int_\Omega \frac{\partial e^p_{\mathcal{M}}}{\partial\mathcal{M}}\delta\mathcal{M}.$$

As we have an equality constraint, the variation of $\mathcal{C}$ is null, so that the necessary Euler-Lagrange condition simplifies to $\delta E_p(\mathcal{M};\delta\mathcal{M}) = 0$ and $\delta\mathcal{C}(\mathcal{M};\delta\mathcal{M}) = 0$ for all $\delta\mathcal{M}$. For the variation $\delta\mathcal{M} = ((\delta r_i)_{i=1,\dots,n-1}, \delta d)$, it comes:

$$\forall\delta r_i,\ \forall\delta d \text{ with } \int\delta d = 0, \text{ we have } \sum_{i=1}^{n-1}\delta E_p(\mathcal{M};\delta r_i) + \delta E_p(\mathcal{M};\delta d) = 0.$$

If (Γ) stands for $\left(\sum_{i=1}^{n-1} r_i^\beta\gamma_i + P^{-\beta}\gamma_n\right)$, the previous equality leads to:

$$\delta E_p(\mathcal{M};\delta r_i) = \int_\Omega \alpha\beta p\, d^{-\frac{\alpha\beta p}{n}}\,(\Gamma)^{\alpha p-1}\left(r_i^{\beta-1}\gamma_i - \frac{P^{-\beta}}{r_i}\gamma_n\right)\delta r_i = 0, \tag{4.5}$$

with the legal choice of $\delta d = 0$. As for all i, functions d, r_i, γ_i, P, α, β and p are supposed to be strictly positive, a particular condition to ensure (4.5) is given by:

$$r_i^{\beta-1}\gamma_i - \frac{P^{-\beta}}{r_i}\gamma_n = 0, \text{ for } i\in[1,n-1].$$

Thus, the anisotropic quotients depend on P and are given by:

$$r_i = \left(\frac{\gamma_n}{\gamma_i}\right)^{\frac{1}{\beta}} P^{-1}, \text{ for } i\in[1,n-1]. \tag{4.6}$$

Multiplying the $n-1$ previous equalities and using the definition of P, we get:

$$\begin{aligned}\left(\prod_{i=1}^{n-1} r_i\right) &= \left(\frac{\gamma_n}{\gamma_i}\right)^{\frac{n-1}{\beta}} P^{-n+1},\\ \text{i.e.}\quad P &= \gamma_n^{\frac{n-1}{\beta}}\left(\prod_{i=1}^{n-1}\gamma_i\right)^{-\frac{1}{\beta}} P^{-n+1}.\end{aligned}$$

Then,

$$P = \gamma_n^{\frac{1}{\beta}} \left(\prod_{i=1}^{n-1} \gamma_i \right)^{-\frac{1}{n\beta}}.$$

Replacing the new expression of P in Equation (4.6), we get the final expression of the anisotropic ratios:

$$r_i = \gamma_i^{-\frac{1}{\beta}} \left(\prod_{j=1}^{n} \gamma_j \right)^{\frac{1}{n\beta}}, \text{ for } i \in [1, n-1]. \tag{4.7}$$

For the legal choices $\delta r_i = 0$ for $i = 1, ..., n-1$, the necessary condition leads to:

$$\delta E_p(\mathcal{M}; \delta d) = \int_\Omega -\frac{\alpha\beta p}{n} d^{-\frac{\alpha\beta p+n}{n}} \left(\sum_{i=1}^{n-1} r_i^\beta \gamma_i + P^{-\beta} \gamma_n \right)^{\alpha p} \delta d = 0, \text{ with } \int_\Omega \delta d = 0. \tag{4.8}$$

A condition to ensure (4.8) is given by:

$$d^{-\frac{\alpha\beta p+n}{n}} \left(\sum_{i=1}^{n-1} r_i^\beta \gamma_i + P^{-\beta} \gamma_n \right)^{\alpha p} = K, \tag{4.9}$$

with K a real constant. From (4.7), we deduce that:

$$\begin{aligned}
P &= \left(\prod_{i=1}^{n-1} r_i \right), \\
&= \prod_{i=1}^{n-1} \left(\gamma_i^{-\frac{1}{\beta}} \left(\prod_{j=1}^{n} \gamma_j \right)^{\frac{1}{n\beta}} \right), \\
&= \left(\prod_{i=1}^{n-1} \gamma_i \right)^{-\frac{1}{\beta}} \left(\prod_{j=1}^{n} \gamma_j \right)^{\frac{n-1}{n\beta}}, \\
&= \gamma_n^{\frac{1}{\beta}} \left(\prod_{j=1}^{n} \gamma_j \right)^{-\frac{1}{\beta}-\frac{n-1}{n\beta}}, \\
&= \gamma_n^{\frac{1}{\beta}} \left(\prod_{j=1}^{n} \gamma_j \right)^{-\frac{1}{n\beta}}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\sum_{i=1}^{n-1} r_i^{\beta}\gamma_i + P^{-\beta}\gamma_n &= \sum_{i=1}^{n-1}\left(\gamma_i^{-\frac{1}{\beta}}\left(\prod_{j=1}^{n}\gamma_j\right)^{\frac{1}{n\beta}}\right)^{\beta}\gamma_i + P^{-\beta}\gamma_n,\\
&= \sum_{i=1}^{n-1}\left(\gamma_i^{-1}\left(\prod_{j=1}^{n}\gamma_j\right)^{\frac{1}{n}}\right)\gamma_i + \left(\gamma_n^{\frac{1}{\beta}}\left(\prod_{j=1}^{n}\gamma_j\right)^{-\frac{1}{n\beta}}\right)^{-\beta}\gamma_n,\\
&= (n-1)\left(\prod_{j=1}^{n}\gamma_j\right)^{\frac{1}{n}} + \left(\prod_{j=1}^{n}\gamma_j\right)^{\frac{1}{n}},\\
&= n\left(\prod_{j=1}^{n}\gamma_j\right)^{\frac{1}{n}}.
\end{aligned}$$

Therefore, Equation (4.9) is rewritten:

$$d^{-\frac{\alpha\beta p+n}{n}}\left(n\left(\prod_{j=1}^{n}\gamma_j\right)^{\frac{1}{n}}\right)^{\alpha p} = K.$$

Thus,

$$d^{-\frac{\alpha\beta p+n}{n}} = Kn^{-\alpha p}\left(\prod_{j=1}^{n}\gamma_j\right)^{-\frac{\alpha p}{n}}.$$

Then,

$$d = \tilde{K}\left(\prod_{j=1}^{n}\gamma_j\right)^{\frac{\alpha p}{\alpha\beta p+n}}, \text{ with } \tilde{K} \text{ a constant.} \tag{4.10}$$

Using the constraint on the complexity defined by (4.4), we get:

$$\begin{aligned}
\int_{\Omega} d &= \tilde{K}\int_{\Omega}\left(\prod_{j=1}^{n}\gamma_j\right)^{\frac{\alpha p}{\alpha\beta p+n}},\\
\Longrightarrow \quad \tilde{K} &= N\left(\int_{\Omega}\left(\prod_{j=1}^{n}\gamma_j\right)^{\frac{\alpha p}{\alpha\beta p+n}}\right)^{-1}.
\end{aligned}$$

Thus, the final expression of the optimal density is:

$$d^{opt} = N\left(\int_{\Omega}\left(\prod_{i=1}^{n}\gamma_j\right)^{\frac{\alpha p}{\alpha\beta p+n}}\right)^{-1}\left(\prod_{j=1}^{n}\gamma_j\right)^{\frac{\alpha p}{\alpha\beta p+n}}.$$

Finally, the optimal eigenvalues of the optimal continuous mesh $\mathcal{M}_{opt}$ solution of Problem (4.3) with the $(\gamma_i)_{i=1,\dots,n}$ fixed, are given by:

$$\lambda_i^{opt} = (h_i^{opt})^{-2} = N^{\frac{2}{n}}\left(\int_{\Omega}\left(\prod_{i=1}^{n}\gamma_j\right)^{\frac{\alpha p}{\alpha\beta p+n}}\right)^{-\frac{2}{n}}\left(\prod_{j=1}^{n}\gamma_j\right)^{-\frac{2}{\beta(\alpha\beta p+n)}}\gamma_i^{\frac{2}{\beta}}. \tag{4.11}$$

4.2.3 Uniqueness and properties of the optimal metric

Order of convergence

Using solution (4.11), we rewrite the error model (4.2):

$$\begin{aligned} e_{\mathcal{M}_{opt}} &= \left(\sum_{i=1}^{n} h_i^{\beta}\,\gamma_i\right)^{\alpha}, \\ e_{\mathcal{M}_{opt}} &= n^{\alpha} N^{-\frac{\alpha\beta}{n}} \left(\int_{\Omega} \left(\prod_{i=1}^{n} \gamma_j\right)^{\frac{\alpha p}{\alpha\beta p+n}}\right)^{\frac{\alpha\beta}{n}} \left(\prod_{j=1}^{n} \gamma_j\right)^{\frac{\alpha}{\alpha\beta p+n}}. \end{aligned}$$

The $\mathbf{L}^p$-norm of the optimal error is deduced:

$$||e_{\mathcal{M}_{opt}}(\mathbf{x})||_{\mathbf{L}^p} = \left(\int_{\Omega} e_{\mathcal{M}}^p\right)^{\frac{1}{p}} = n^{\alpha} N^{-\frac{\alpha\beta}{n}} \left(\int_{\Omega} \left(\prod_{i=1}^{n} \gamma_j\right)^{\frac{\alpha p}{\alpha\beta p+n}}\right)^{\frac{\alpha\beta p+n}{np}}.$$

Consequently, from the previous optimal error, we get the asymptotic order of convergence:

$$||e_{\mathcal{M}_{opt}}(\mathbf{x})||_{\mathbf{L}^p} \leqslant \frac{Cst}{N^{\frac{\alpha\beta}{n}}}. \tag{4.12}$$

From this inequality, we deduce that the order of convergence is $\alpha\beta$.

Uniqueness

We now prove that the optimal continuous mesh defined by (4.11) is the unique solution of Problem (4.3) verifying $E_p(\mathcal{M}_{opt})^p \leqslant E_p(\mathcal{M})^p$, for all $\mathcal{M}$ having the same fixed $(\gamma_i)_{i=1,\ldots,n}$. To do so, we consider a continuous mesh $\mathcal{M}$ of complexity N defined by its $(n-1)$ anisotropic ratios r_i, with $i \in [1, n-1]$ and its density d. To take into account the constraint on the density, the density is rewritten as: $d = N\left(\int_{\Omega} f\right)^{-1} f$. The functions r_i, with $i \in [1, n-1]$, f and d are strictly positive. From (4.3), the error committed with $\mathcal{M}$ is:

$$E_p(\mathcal{M})^p = N^{-\frac{\alpha p}{n}} \left(\int_{\Omega} f\right)^{-\frac{\alpha\beta p}{n}} \int_{\Omega} f^{-\frac{\alpha\beta p}{n}} \left(\sum_{i=1}^{n-1} r_i^{\beta}\gamma_i + \gamma_n \prod_{i=1}^{n-1} r_i^{-\beta}\right)^{\alpha p}.$$

The error committed with the optimal solution $\mathcal{M}_{opt}$ is:

$$E_p(\mathcal{M}_{opt})^p = n^{\alpha p} N^{-\frac{\alpha\beta p}{n}} \left(\int_{\Omega} \left(\prod_{i=1}^{n} \gamma_i\right)^{\frac{\alpha p}{\alpha\beta p+n}}\right)^{-\frac{\alpha\beta p+n}{n}}.$$

To prove $E_p(\mathcal{M}_{opt})^p \leqslant E_p(\mathcal{M})^p$, we use the generalized arithmetic-geometric inequality which comes from the concavity of the logarithm function ln:

$$\begin{aligned}
\ln\left(\frac{1}{n}\sum_{i=1}^{n} r_i^{\beta}\gamma_i\right) &\geqslant \frac{1}{n}\sum_{i=1}^{n}\ln\left(r_i^{\beta}\gamma_i\right), \\
&\geqslant \sum_{i=1}^{n}\ln\left(r_i^{\frac{\beta}{n}}\gamma_i^{\frac{1}{n}}\right), \\
&\geqslant \ln\left(\prod_{i=1}^{n} r_i^{\frac{\beta}{n}}\prod_{i=1}^{n}\gamma_i^{\frac{1}{n}}\right) = \ln\left(\prod_{i=1}^{n}\gamma_i^{\frac{1}{n}}\right),
\end{aligned}$$

as $\prod_{i=1}^{n} r_i = 1$. Substituting the value of r_n in the previous inequality, it comes:

$$\sum_{i=1}^{n-1} r_i^{\beta}\gamma_i + \gamma_n\prod_{i=1}^{n-1} r_i^{-\beta} \geqslant n\ln\left(\prod_{i=1}^{n}\gamma_i^{\frac{1}{n}}\right).$$

Finally, if we denote

$$g = \left(\prod_{i=1}^{n}\gamma_i\right)^{\frac{\alpha p}{\alpha\beta p+n}},$$

we have

$$\left\{\begin{array}{rcl}
E_p(\mathcal{M}_{opt})^{\frac{n}{\alpha\beta p+n}} &=& n^{\frac{\alpha\beta pn}{\alpha\beta p+n}} N^{-\frac{\alpha\beta p}{\alpha\beta p+n}}\displaystyle\int_\Omega g, \\
E_p(\mathcal{M})^{\frac{n}{\alpha\beta p+n}} &\geqslant& n^{\frac{\alpha\beta pn}{\alpha\beta p+n}} N^{-\frac{\alpha\beta p}{\alpha\beta p+n}}\left(\displaystyle\int_\Omega f\right)^{\frac{\alpha\beta p}{\alpha\beta p+n}}\left(\displaystyle\int_\Omega f^{-\frac{\alpha\beta p}{n}} g^{\frac{\alpha\beta p+n}{n}}\right)^{\frac{n}{\alpha\beta p+n}}.
\end{array}\right.$$

Using the Hölder inequality, it comes:

$$\left(\int_\Omega f^{\frac{\alpha\beta p}{\alpha\beta p+n}}\left(\frac{g}{f^{\frac{\alpha\beta p}{\alpha\beta p+n}}}\right)\right) \leqslant \left(\int_\Omega f\right)^{\frac{\alpha\beta p}{\alpha\beta p+n}}\left(\int_\Omega f^{-\frac{\alpha\beta p}{n}} g^{\frac{\alpha\beta p+n}{n}}\right)^{\frac{n}{\alpha\beta p+n}}, \tag{4.13}$$

as

$$\left\{\begin{array}{c}
1+\frac{n}{\alpha\beta p} \geqslant 1, \\
1+\frac{\alpha\beta p}{n} \geqslant 1, \\
\dfrac{1}{1+\frac{n}{\alpha\beta p}} + \dfrac{1}{1+\frac{\alpha\beta p}{n}} = 1.
\end{array}\right.$$

Moreover, Relation (4.13) implies $E_p(\mathcal{M}_{opt}) \leqslant E_p(\mathcal{M})$, for all $\mathcal{M}$ having the same fixed $(\gamma_i)_{i=1,..,n}$. As Problem (4.3) is strictly convex, the optimal solution is $\mathcal{M}_{opt}$ is unique.

4.3 THE QUADRATIC INTERPOLATION CASE

In the quadratic interpolation case, the local error model on a continuous mesh $\mathbf{M}$ defined by its sizes $(h_i)_{i=1,n}$ and its eigenvectors $(\mathbf{v}_i)_{i=1,n}$ is given by $\alpha = \frac{3}{2}$ and $\beta = 2$.

In the previous section, we solved a local optimization problem based on tensor decomposition to find the optimal local metric $\mathcal{M}_{opt}^{loc}(u) = Q(d^{(3)}(u))$ included in the isoline or the isosurface 1 of the error function u in 2D and 3D. Let us assume the dimension of the space is n.

The local error model on $\mathcal{M}$ is given by:

$$e_{\mathcal{M}}(\mathbf{x}) = d^{(3)}(u)(\mathcal{M}^{-\frac{1}{2}}\mathbf{x}) = \text{trace}\left(\mathcal{M}^{-\frac{1}{2}}\, Q(d^{(3)}(u))(\mathbf{x})\, \mathcal{M}^{-\frac{1}{2}}\right)^{\frac{3}{2}}.$$

Consequently, in the quadratic case, the optimal global metric $\mathbf{M}_{opt} = (\mathcal{M}_{opt}(\mathbf{x}))_{\mathbf{x}\in\mathbb{R}^n}$ is the solution of the following variational calculus problem:

$$\mathbf{M}_{opt} = \min_{\mathcal{M}} E_p(\mathcal{M}), \tag{4.14}$$

with

$$\begin{aligned} E_p(\mathcal{M}) &= \left(\int_\Omega |e_{\mathcal{M}}(\mathbf{x})|^p \,\mathrm{d}\mathbf{x}\right)^{\frac{1}{p}}, \\ &= \left(\int_\Omega \left(\left|d^{(3)}(u)(\mathcal{M}^{-\frac{1}{2}}\mathbf{x})\right|\right)^p \,\mathrm{d}\mathbf{x}\right)^{\frac{1}{p}}, \\ &= \left(\int_\Omega \text{trace}\left(\mathcal{M}^{-\frac{1}{2}}\, \mathcal{M}_{opt}^{loc}(u)(\mathbf{x})\, \mathcal{M}^{-\frac{1}{2}}\right)^{\frac{3p}{2}} \,\mathrm{d}\mathbf{x}\right)^{\frac{1}{p}}, \end{aligned}$$

under the constraint $\mathcal{C}(\mathcal{M}) = N = \displaystyle\int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x}$.

Using the general resolution process of the previous section, we deduce the solution of Problem (4.14):

$$\mathcal{M}_{opt}(\mathbf{x}) = N\,\left(\int_\Omega \det(\mathcal{M}_{opt}^{loc}(u)(\mathbf{x}))^{\frac{3p}{2(3p+n)}}\right)^{-1} \det(\mathcal{M}_{opt}^{loc}(u)(\mathbf{x}))^{-\frac{1}{3p+n}}\, \mathcal{M}_{opt}^{loc}(\mathbf{x}). \tag{4.15}$$

The optimal value of the density:

$$d_{opt}(\mathbf{x}) = N\,\left(\int_\Omega \det(\mathcal{M}_{opt}^{loc}(u)(\mathbf{x}))^{\frac{3p}{2(3p+n)}}\right)^{-1} \det(\mathcal{M}_{opt}^{loc}(u)(\mathbf{x}))^{\frac{3p}{2(3p+n)}},$$

and the optimal value of the error is:

$$E_p(\mathcal{M}_{opt})^p = n^{\frac{3}{2}}\, N^{-\frac{3}{n}}\,\left(\int_\Omega \det(\mathcal{M}_{opt}^{loc}(u)(\mathbf{x}))^{\frac{3p}{2(3p+n)}}\right)^{\frac{3p+n}{np}}.$$

4.3.1 Optimal sizes and orientations

For a complexity N, the optimal local metric $\mathcal{M}_{opt}^{loc}(u) = Q(d^{(3)}(u))$ admits a spectral decomposition given by:

$$\begin{aligned} \mathcal{M}_{opt}^{loc} &= \mathcal{R}_{opt}^{loc}\ \Lambda_{opt}^{loc}\ {}^t\mathcal{R}_{opt}^{loc}, \\ &= \mathcal{R}_{opt}^{loc}\ diag(\lambda_i^{loc})\ {}^t\mathcal{R}_{opt}^{loc}. \end{aligned}$$

The orientation of $\mathcal{M}_{opt}^{loc}$ is given by the basis $\mathcal{R}_{opt}^{loc} = (\mathbf{v}_{opt,i}^{loc})_{i=1,...,n}$. The directional sizes are given by $h_i^{loc} = \frac{1}{\sqrt{\lambda_i^{loc}}}$, $\forall\, i = 1, ..., n$.

Thus, the multi-scale continuous mesh $\mathcal{M}_{\mathbf{L}^p}$ is given by the following decomposition:

$$\begin{aligned} \mathcal{M}_{\mathbf{L}^p} &= \mathcal{R}_{opt}^{loc}\ \Lambda\ {}^t\mathcal{R}_{opt}^{loc}, \\ &= D_{\mathbf{L}^p}\ \det\left(\mathcal{M}_{opt}^{loc}(u)\right)^{-\frac{1}{3p+n}}\ \mathcal{R}_{opt}^{loc}\ \Lambda_{opt}^{loc}\ {}^t\mathcal{R}_{opt}^{loc}, \end{aligned}$$

with $D_{\mathbf{L}^p}$ a global normalization coefficient given by:

$$D_{\mathbf{L}^p} = N\ \left(\ \int_\Omega \det\left(\mathcal{M}_{opt}^{loc}(u)(\mathbf{x})\right)^{\frac{3p}{2(3p+n)}}\ \right)^{-1}.$$

The multi-scale optimal metric in terms of orientations and elongations is given by:

$$\mathcal{M}_{\mathbf{L}^p} = D_{\mathbf{L}^p}\ \det\left(\left|Q(d^{(3)}(u))\right|\right)^{-\frac{1}{3p+n}}\ \left|d^{(3)}(u)\right|,$$

with $D_{\mathbf{L}^p}$ a global normalization coefficient given by:

$$D_{\mathbf{L}^p} = N\ \left(\ \int_\Omega \det\left(\left|Q(d^{(3)}(u))\right|\right)^{\frac{3p}{2(3p+n)}}\ \right)^{-1}.$$

4.3.2 Mesh convergence

The order of convergence for a sequel of continuous meshes $(\mathcal{M}_{\mathbf{L}^p}^N)_N$ verifies:

$$||e_{\mathcal{M}_{\mathbf{L}^p}^N}||_{\mathbf{L}^p(\Omega)} \leqslant \frac{C}{N^{\frac{3}{n}}}. \tag{4.16}$$

Relation (4.16) points out a global third-order of mesh convergence for the mesh adaptation process. Indeed, in 3D, as $N = \mathcal{O}(h^{-3})$, relation (4.16) simplifies to:

$$||e_{\mathcal{M}_{\mathbf{L}^p}^N}||_{\mathbf{L}^p(\Omega)} \leqslant C\, h^3.$$

4.4 APPLICATION

This section presents a double $\mathbf{L}^2$-projection method, a classical recovery technique to find the third-order derivatives a_i of u_h on each point of in 2D and 3D. But, we first recall how estimates are applied on discrete solution u_h.

4.4.1 Application to solution given by numerical approximation

Let $\bar{V}_h^k$ be the space of piecewise polynomials of degree k and V_h^k be the space of continuous piecewise polynomials of degree k associated with a given mesh $\mathcal{H}$ of domain Ω_h. Let $u_h \in V_h^k$. We denote by Π_h^k the k-order interpolate of the numerical solution u_h and by R_h a reconstruction operator applied to the numerical approximation u_h. We assume that the reconstruction $R_h u_h$ is better than u_h for a given norm $\|.\|$ in the sense that:

$$\|u - R_h u_h\| \leq \alpha \|u - u_h\| \quad \text{where} \;\; 0 \leq \alpha < 1\,.$$

From the triangle inequality we deduce:

$$\|u - u_h\| \;\leq\; \frac{1}{1-\alpha}\|R_h u_h - u_h\|\,.$$

If the reconstruction operator R_h has the property:

$$\Pi_h^k R_h \phi_h = \phi_h\,, \quad \forall \phi_h \in V_h^k\,, \tag{4.17}$$

we can then bound the approximation error of the solution by the interpolation error of the reconstructed function $R_h u_h$:

$$\|u - u_h\| \leq \frac{1}{1-\alpha}\|R_h u_h - \Pi_h^k R_h u_h\|\,. \tag{4.18}$$

We can exhibit the following upper bound of the approximation error:

$$\|u - u_h\| \leq \frac{6\,N^{-\frac{2}{3}}}{1-\alpha}\left(\int_\Omega \det\left(|H_{R_h u_h}|\right)^{\frac{p}{2p+3}}\right)^{\frac{2p+3}{3p}}\,.$$

In the general case, it is important to note that $\mathcal{M}_{\mathbf{L}^p}$ applied to $R_h u_h$ does not allow us to generate an optimal adapted mesh to control the approximation error $\|u - u_h\|$. The approximation error is only controlled when all previous assumptions are verified.

4.4.2 Third-order derivatives recovery technique

Let K be a element of a mesh $\mathcal{H}$ and $[\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n]$ its list of vertices (n=2,3). We consider the continuous local interpolation error $e_{\mathcal{M}}$ associated with the discrete local interpolation error $e_h = |u - \Pi_h^2 u|$. If $\mathbf{x} \in K$, the quadratic interpolate $\Pi_h^2 u$ of the numerical solution u_h on K is given by:

$$\forall \mathbf{x} \in \Omega, \quad \Pi_h^2 u(\mathbf{x}) = \sum_{i=1}^{n_i} u_h(\mathbf{p}_i)\varphi_i(\mathbf{x}),$$

where $n_i = \frac{(n+2)!}{2!\,n!}$ is the number of nodes of the element K and $\varphi_i(.)$ is the i^{th} $\mathbb{P}^2$ Lagrange shape function defined by:

$$\begin{cases} \varphi_i(\mathbf{x}) = \psi_i(\mathbf{x})(2\psi_i(\mathbf{x}) - 1), & 1 \leqslant i \leqslant n \,(\text{vertices}) \\ \varphi_i(\mathbf{x}) = 4\,\psi_{[i]}(\mathbf{x})\,\psi_{[i+1]}(\mathbf{x}), & n+1 \leqslant i \leqslant n_i \,(\text{midpoints of edges}) \end{cases}$$

and ψ_i is the i^{th} $\mathbb{P}^1$ Lagrange shape function defined by :

$$\psi_i(\mathbf{p}_j) = \delta_{ij}, \quad \mathbf{p}_j \in \mathcal{H}.$$

As we saw in the previous chapter, the continuous error $e_{\mathcal{M}}$ is modeled by a homogeneous polynomial of degree 3 in n variables on each vertex of the mesh:

$$\begin{cases} P_e(\mathbf{x}) = \sum\limits_{i=0}^{3} \binom{3}{i} a_i \, x^i \, y^{3-i}, & \text{in 2D} \\ P_e(\mathbf{x}) = \sum\limits_{i=0}^{3} \sum\limits_{j=0}^{3-i} \binom{3}{i,j} a_{ij} \, x^i \, y^j \, z^{3-i-j}, & \text{in 3D} \end{cases}$$

We seek for recovering the coefficients a_i (respectively a_{ij}) by the pre-cited technique. To do so, we consider the piecewise constant Hessian $Hu_h(\mathbf{x})$ given by:

$$Hu_h(\mathbf{x}) = \sum_{i=1}^{n_i} u_h(\mathbf{p}_i) \, H\varphi_i(\mathbf{x}).$$

As Hu_h is not defined at mesh vertices, the nodal Hessians are recovered from the piecewise constant Hessian representation $Hu_h(\mathbf{x})$ using the $\mathbf{L}^2$-projection operator based on the Clément interpolation operator [28].
The stencil $S_i = \{K_j\}_j$ of shape function φ_i is the topological ball of a vertex $\mathbf{p}_i$. We introduce the following approximation spaces:

$$V_h^0 = \left\{ \; v \in \mathbf{L}^2(\Omega) \, | \, v_{|K} \in \mathbb{P}^0, \; \forall K \in \mathcal{H} \; \right\},$$

$$V_h^1 = \left\{ \; v \in \mathbf{C}^0(\Omega) \, | \, v_{|K} \in \mathbb{P}^1, \; \forall K \in \mathcal{H} \; \right\}.$$

The Clément interpolation operator $\Pi_c : V_h^0 \to V_h^1$ is defined by:

$$\forall \, v \in V_h^0, \quad \Pi_c v = \sum_{i=0}^{n} \Pi_0 v(\mathbf{p}_i) \, \varphi_i,$$

where $\Pi_0 v \in V_h^0$ is defined by:

$$\text{for } v \in \mathbf{L}^2, \text{ for } S_i \subset \mathcal{H}, \begin{cases} \Pi_0 v_{|S_i} \in \mathbb{P}^0 \\ \int_{S_i} (\Pi_0 v - v) w = 0, \; \forall w \in \mathbb{P}^0. \end{cases}$$

For each $\mathbf{p}_i$, we thus have the following Hessian reconstruction:

$$H_R u_h(\mathbf{p}_i) = \frac{\sum_{K_j \in S_i} |K_j| \, Hu_{h_{|K_j}}}{|S_i|},$$

where $Hu_{h_{|K_j}}$ is the constant Hessian on element $K_j \in S_i$. $|S_i|$ and $|K_j|$ denote the volume of stencil S_i and element K_j, respectively.

The recovery procedure provides us with Hessian nodal values and thus we get a piecewise linear representation of the Hessian on $\mathcal{H}$.

To recover the third-order derivatives $a_i = [D_R^{(3)}]_i$ (respectively $a_{ij} = [D_R^{(3)}]_{ij}$) from u_h, we apply a gradient reconstruction procedure to each component of the gradient of the recovered Hessian $H_R u_h$:

$$D_R^{(3)} u_h(\mathbf{p}_i) = \frac{\sum_{K_j \in S_i} |K_j| \, \nabla (H_R u_h)_{|K_j}}{|S_i|}. \tag{4.19}$$

4.5 ANALYTICAL EXAMPLES

In this section, we propose to validate our approach in 2D and 3D analytical examples where a mesh adaptation based on $\mathbf{L}^1$-norm is applied. To do so, we compare an adaptation based on the third-order optimal metric (with third-order derivative recovery) with an adaptation only based on the Hessian of the solution u. In other words, we compare a $\mathbb{P}^1$-driven adaptation with a $\mathbb{P}^2$-driven adaptation.
For both strategies, the interpolation error level is computed by means of 5^{th} order Gauss interpolation to estimate $\|u - \Pi_h^2 u\|_{\mathbf{L}^1(\Omega)}$ and $\|u - \Pi_h^2 u\|_{\mathbf{L}^2(\Omega)}$. For both case, we use a $\mathbb{P}^2$-Lagrange element (triangle or tetrahedron) to represent the function.
In 2D, we use **Yams** [43] to adapt the mesh and we use **Feflo.a** [72] in 3D. All the strategies have been implemented in **Metrix** [5]. We refer interested authors to **appendix B** for all the relative functions in Matlab.

4.5.1 Mesh adaptation algorithm

We present the mesh adaptation algorithm for analytic functions.

Algorithm 3: Mesh Adaptation Loop for Analytic Functions

Initial mesh $\mathcal{H}_0$ and targeted complexity N

For $i = 1, n_{adap}$

1. *$\{f_{i-1}, H_{f_{i-1}}\}$ = Evaluate f and its Hessian H_f on mesh $\mathcal{H}^{i-1}$;*
2. *$(\mathcal{M}_{\mathbf{L}^p, i-1})$ = Compute metric $\mathcal{M}_{\mathbf{L}^p}$ according to Relation (4.15);*
3. *$\mathcal{H}_i$ = Generate a new mesh from pair $(\mathcal{H}_{i-1}, \mathcal{M}_{\mathbf{L}^p, i-1})$;*

End For

4.5.2 Two-dimensional examples

Example 4.1
We consider the function $f_1(x, y)$ defined on $[-1, 1] \times [-1, 1]$:

$$f_1(x, y) = x^3 + y^3.$$

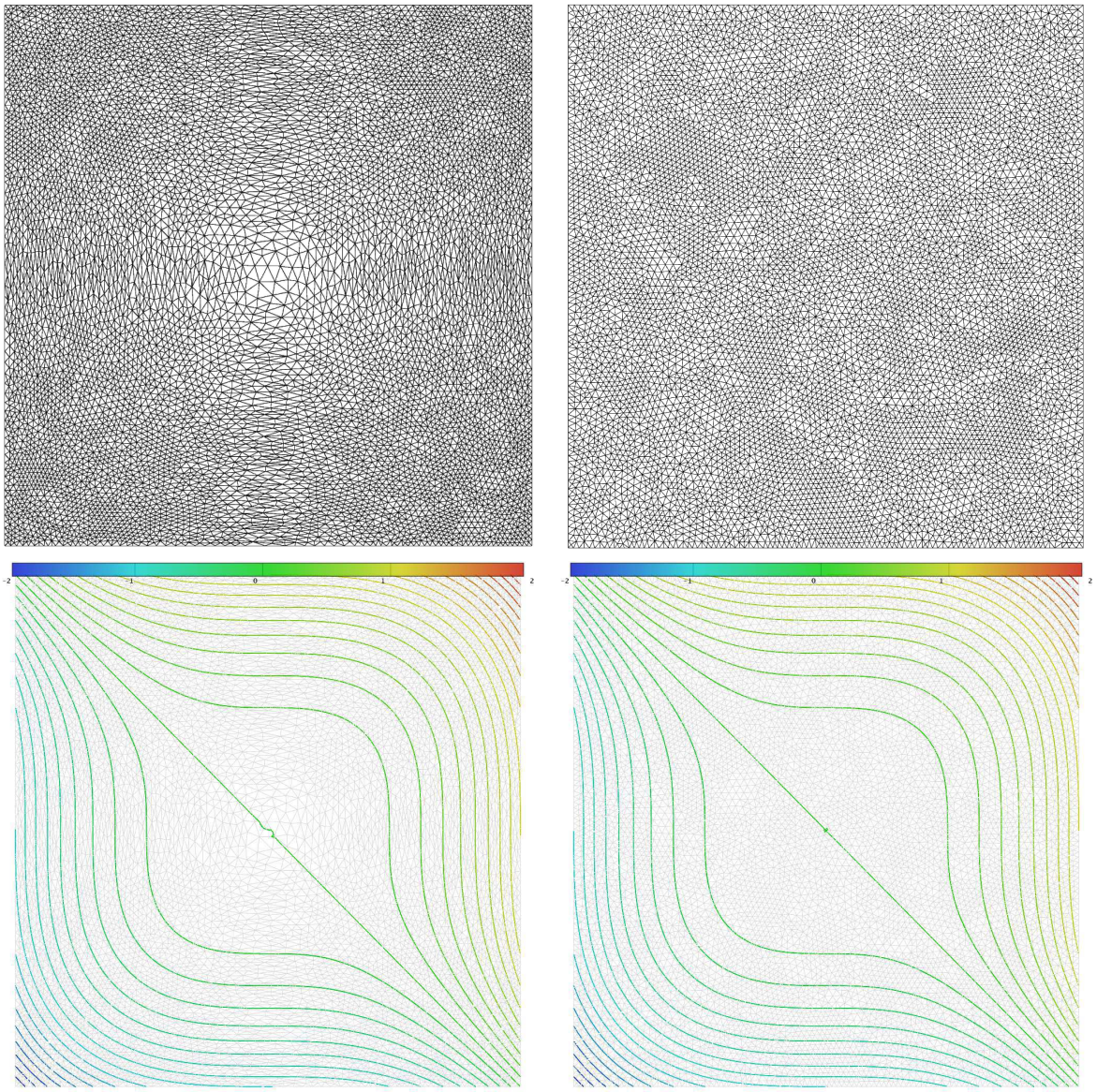

Figure 4.1: $\mathbb{P}^1$-driven adapted mesh (top left) and $\mathbb{P}^2$-driven adapted mesh (top right) to f_1. Each mesh contains around 32 000 degrees of freedom. Adapted meshes and iso-values of f_1 for a $\mathbb{P}^1$-driven adaptation (bottom left) and a $\mathbb{P}^2$-driven adaptation (bottom right).

Comments. The f_1 function is an isotropic function which behaves in the same manner in the x and y directions, cf. **Figure 4.1**. For a fixed complexity $N = 6400$, we see that

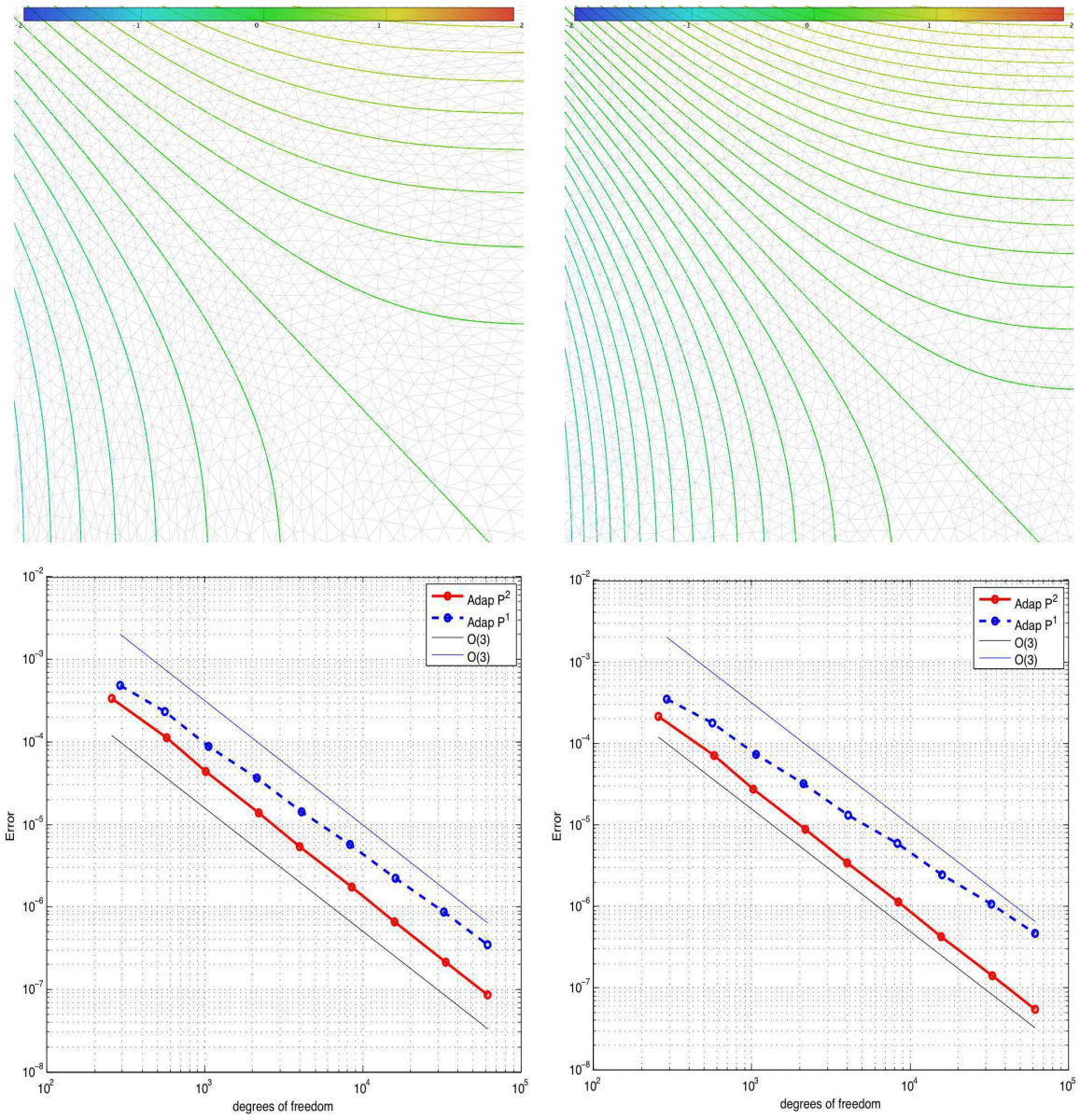

Figure 4.2: Closer view of the adapted meshes and the iso-values of f_1 for a $\mathbb{P}^1$-driven adaptation (top left) and a $\mathbb{P}^2$-driven adaptation (top right). Convergence curves: $\mathbf{L}^1$-norm of the error versus the number of degrees of freedom (bottom left) and $\mathbf{L}^2$-norm of the error versus the number of degrees of freedom (bottom right). Both meshes contains around 32 000 degrees of freedom.

the $\mathbb{P}^2$-driven adaptation leads to a quasi-uniform mesh as $d^{(3)}(f_1)$ is constant for all point of the computational domain. The representation of the function is $\mathbb{P}^2$-exact for a $\mathbb{P}^2$-driven adaptation unlike the $\mathbb{P}^1$-driven adaptation. The spatial convergence curves, depicted in **Figure 4.2** on the bottom, show an asymptotic third-order of convergence for $\mathbf{L}^1$ and $\mathbf{L}^2$-norms for a sequence of $\mathbb{P}^2$-driven adapted mesh. On the contrary, a sequence of $\mathbb{P}^1$-driven adapted mesh leads to a loss of this third-order of convergence.

Example 4.2
We consider the Gaussian function $f_2(x, y)$ defined on $[-1, 1] \times [-1, 1]$:

$$f_2(x, y) = e^{-10(x^2+y^2)}.$$

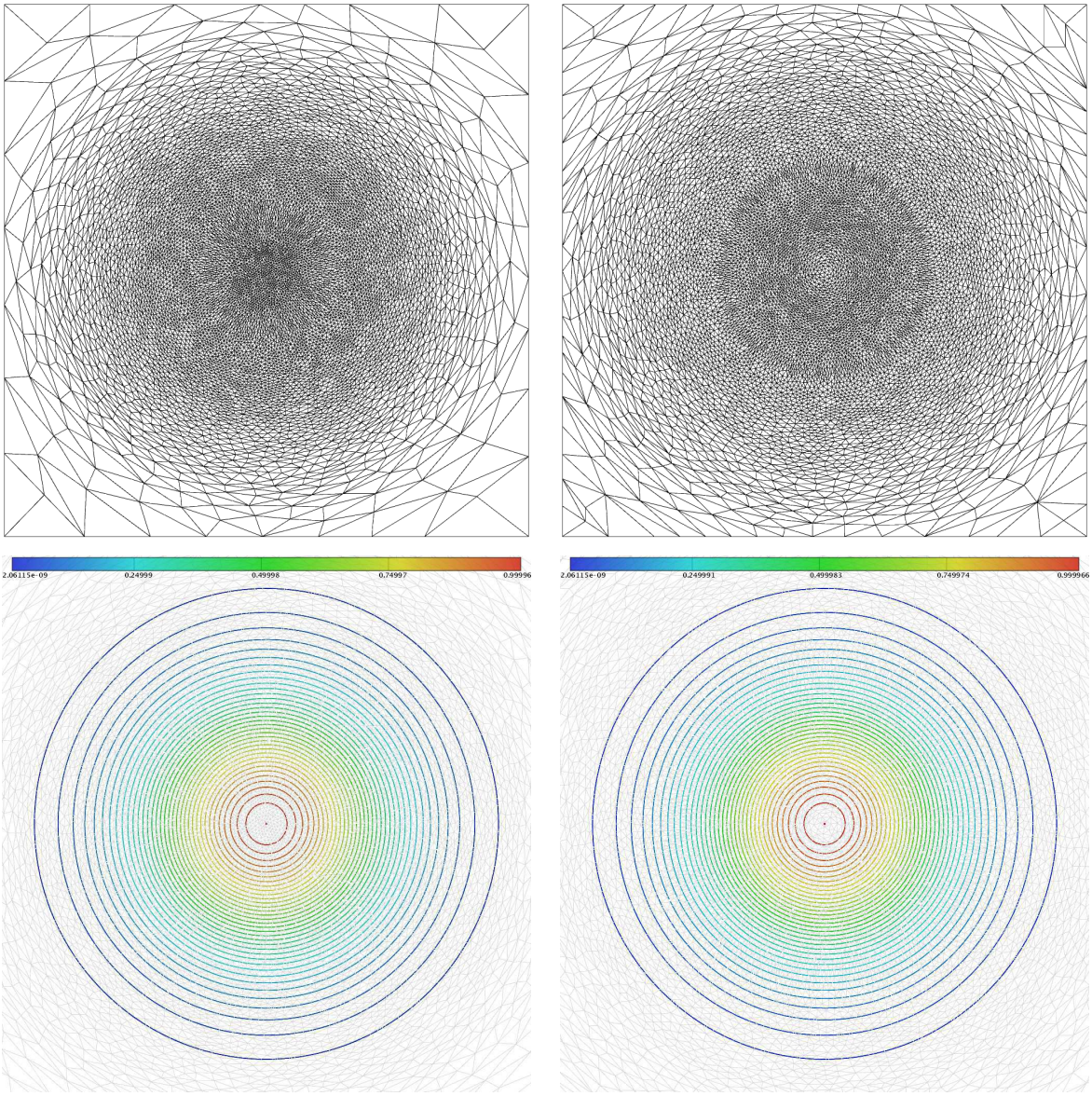

Figure 4.3: $\mathbb{P}^1$-driven adapted mesh (top left) and $\mathbb{P}^2$-driven adapted mesh (top right) to f_2. Each mesh contains around 32 000 degrees of freedom. Adapted meshes and iso-values of f_2 for a $\mathbb{P}^1$-driven adaptation (bottom left) and a $\mathbb{P}^2$-driven adaptation (bottom right).

Comments. The f_2 function is an isotropic function which behaves in the same manner in the x and y directions like the f_1 function, cf. **Figure 4.3**. For a fixed complexity $N = 6400$, the adaptation driven by an optimal third-order metric shows a

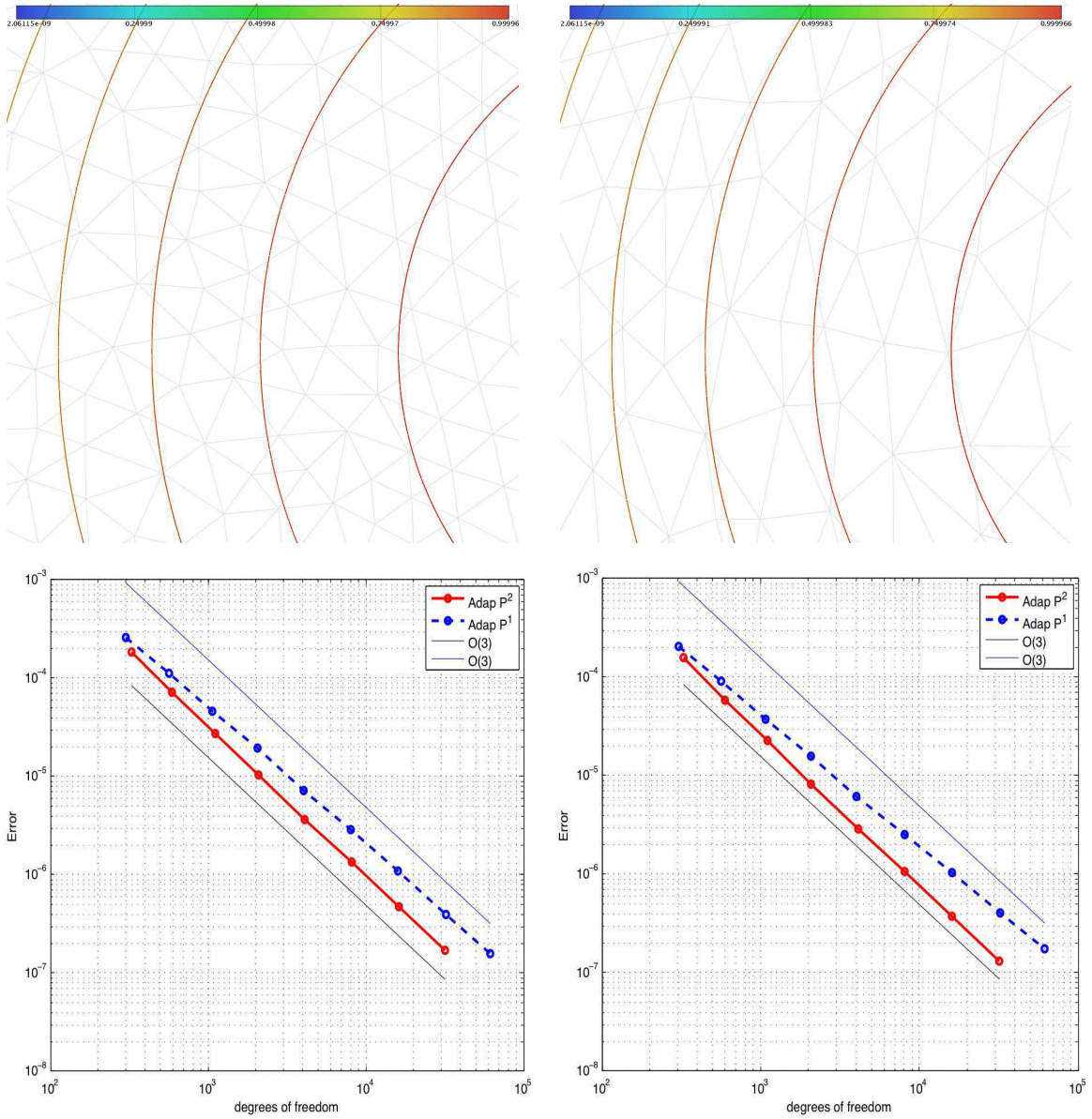

Figure 4.4: Closer view of the adapted meshes and the iso-values of f_2 for a $\mathbb{P}^1$-driven adaptation (top left) and a $\mathbb{P}^2$-driven adaptation (top right). Convergence curves: $\mathbf{L}^1$-norm of the error versus the number of degrees of freedom (bottom left) and $\mathbf{L}^2$-norm of the error versus the number of degrees of freedom (bottom right). Both meshes contains around 32 000 degrees of freedom.

non-uniform mesh with $d^{(3)}(f_2)$ not constant on all the computational domain. Similarly, the $\mathbb{P}^1$-driven adaptation is not uniform as it is the case on the Hessian of the Gaussian function. However, the spatial convergence curves, depicted in **Figure 4.4** on the bottom, show an asymptotic third-order of convergence for $\mathbf{L}^1$ and $\mathbf{L}^2$-norms for a sequence of $\mathbb{P}^2$-driven adapted mesh. On the contrary, a sequence of $\mathbb{P}^1$-driven adapted mesh based on the Hessian of f_2 leads to a loss of the asymptotic third-order

of convergence.

Example 4.3
We consider the smooth function $f_3(x, y)$ defined on $[-1, 1] \times [-1, 1]$:

$$f_3(x,y) = \begin{cases} 0.01 & \sin(50\,xy) \quad \text{if} \quad xy \leqslant -\frac{\pi}{50} \\ & \sin(50\,xy) \quad \text{if} \quad -\frac{\pi}{50} < xy \leqslant \frac{2\pi}{50} \\ 0.01 & \sin(50\,xy) \quad \text{if} \quad xy > \frac{2\pi}{50} \end{cases}$$

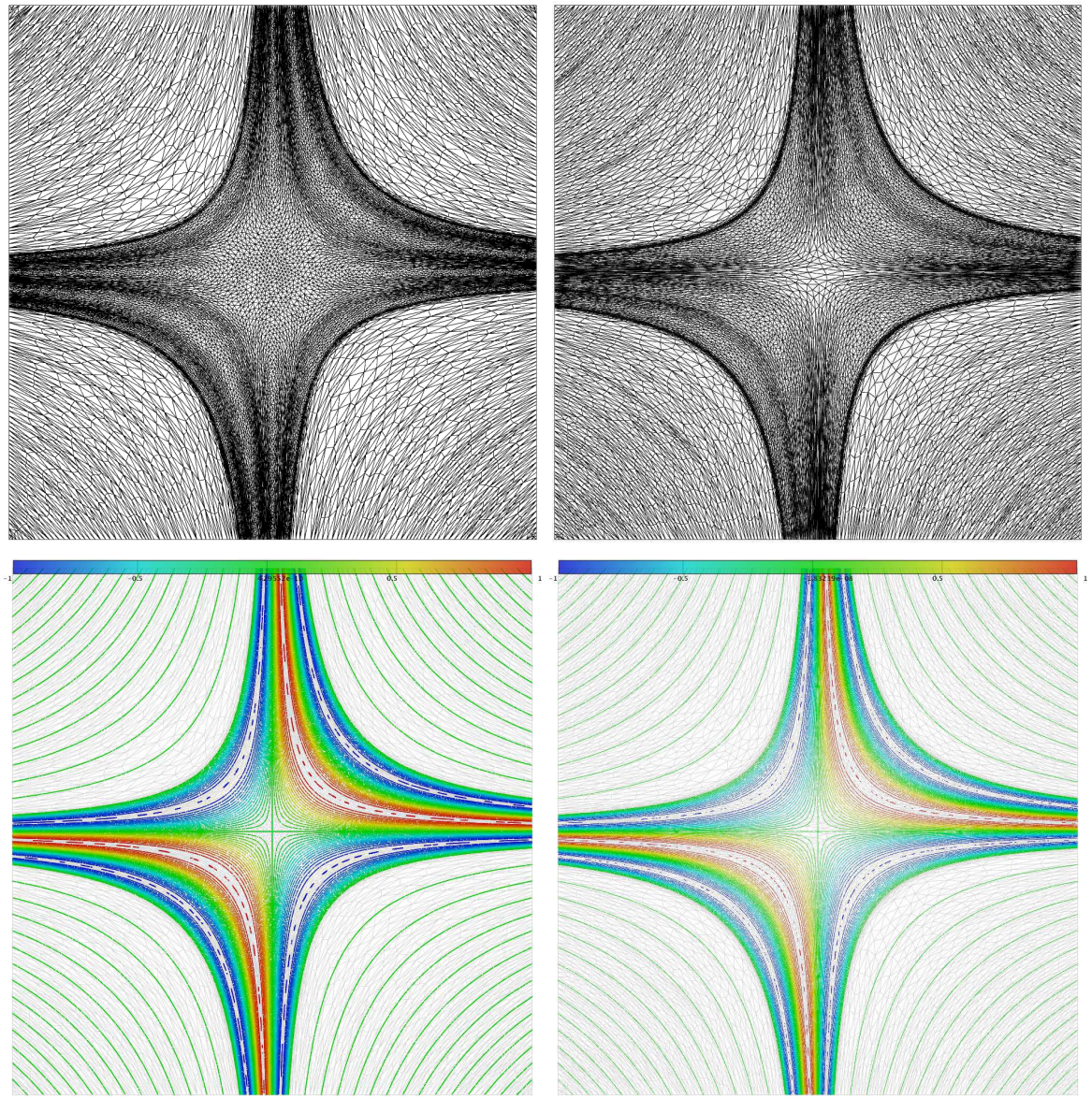

Figure 4.5: $\mathbb{P}^1$-driven adapted mesh (top left) and $\mathbb{P}^2$-driven adapted mesh (top right) to f_3. Each mesh contains around 42 000 degrees of freedom. Adapted meshes and iso-values of f_3 for a $\mathbb{P}^1$-driven adaptation (bottom left) and a $\mathbb{P}^2$-driven adaptation (bottom right).

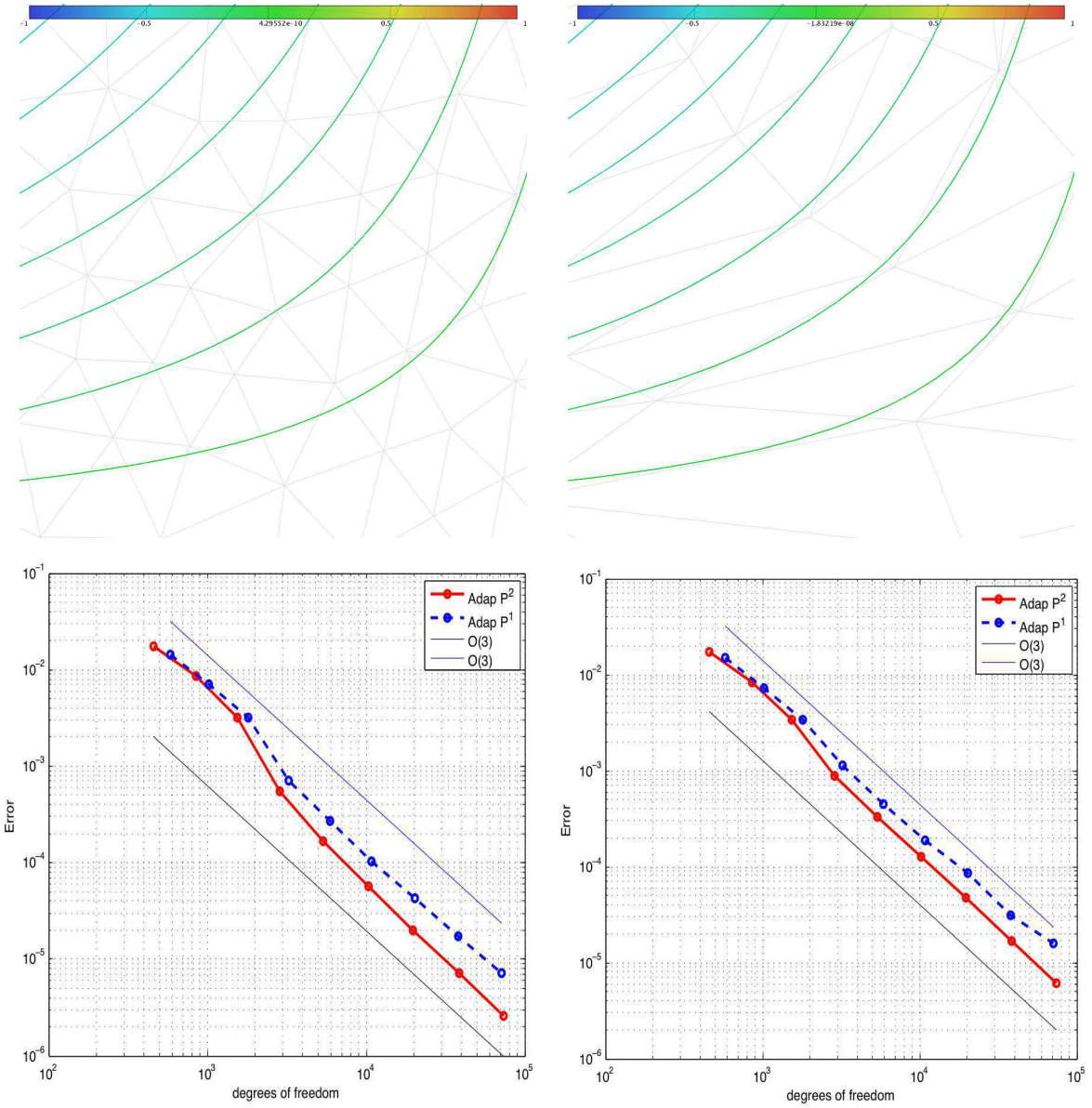

Figure 4.6: Closer view of the adapted meshes and the iso-values of f_3 for a $\mathbb{P}^1$-driven adaptation (top left) and a $\mathbb{P}^2$-driven adaptation (top right). Convergence curves: $\mathbf{L}^1$-norm of the error versus the number of degrees of freedom (bottom left) and $\mathbf{L}^2$-norm of the error versus the number of degrees of freedom (bottom right). Both meshes contains around 42 000 degrees of freedom.

Comments. The f_3 function is an anisotropic function which is composed of small and large scale variations with an amplitude of 0.01 and 1, cf. **Figure 4.5**. As f_3 is a smooth function and as we use a $\mathbb{P}^2$-Lagrange approximation, we have to find an asymptotic order of convergence of order three even for a uniform mesh. But that will be achieved once the small fluctuations will be captured. Looking at the convergence curves, depicted in **Figure 4.6** on the bottom, we have an asymptotic third-order of

convergence for $\mathbf{L}^1$ and $\mathbf{L}^2$-norms for a sequence of $\mathbb{P}^2$-driven adapted mesh once the small variations of the function have been captured. With the sequence of $\mathbb{P}^1$-driven adapted mesh, the asymptotic convergence order is reached but has been achieved later.

4.5.3 Three-dimensional example

Example 4.4
We consider the function $f(x, y, z)$ defined on $[-5.5, -4.5] \times [-0.5, 0.5] \times [-0.5, 0.5]$:

$$f_4(x, y, z) = cos(\pi\, xyz).$$

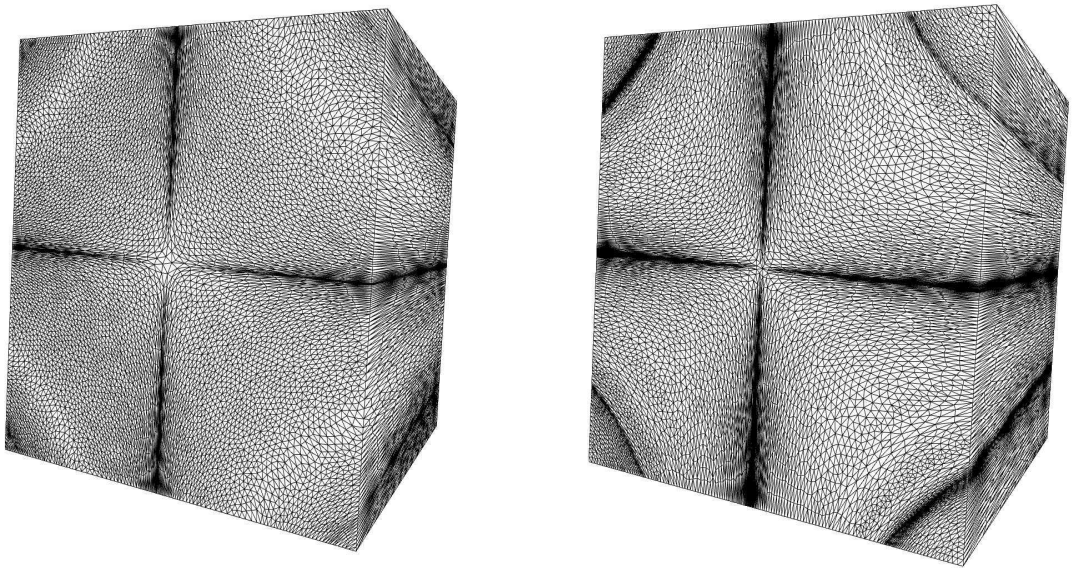

Figure 4.7: $\mathbb{P}^1$-driven adapted mesh (left) and $\mathbb{P}^2$-driven adapted mesh (right) to f_4. Each mesh contains around 344 000 degrees of freedom.

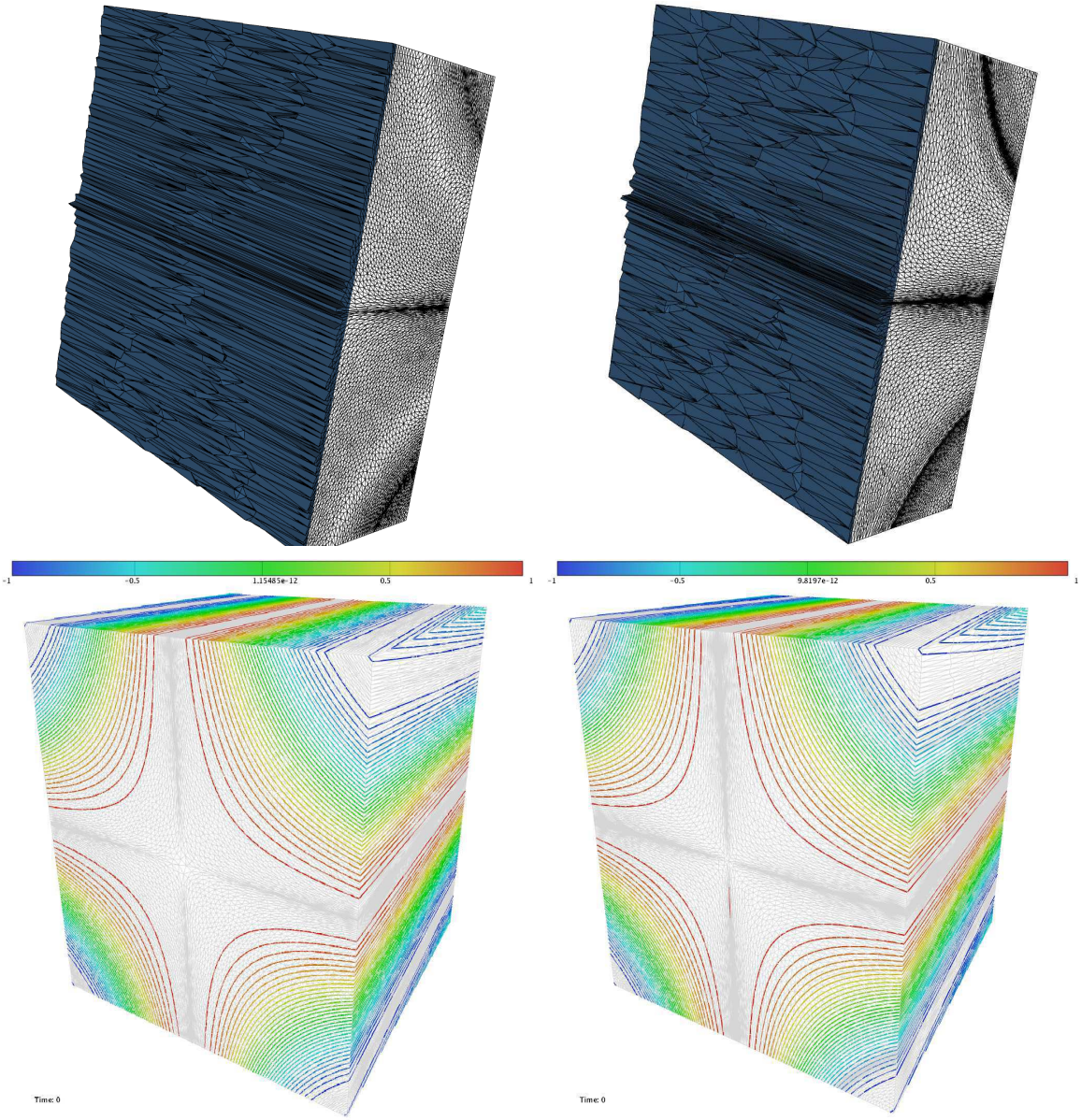

Figure 4.8: Sections showing the anisotropy of $\mathbb{P}^1$-driven adapted mesh (top left) and $\mathbb{P}^2$-driven adapted mesh (top right) to f_4. Each mesh contains around 344 000 degrees of freedom. Iso-values of f_4 for a $\mathbb{P}^1$-driven adaptation (bottom left) and a $\mathbb{P}^2$-driven adaptation (bottom right).

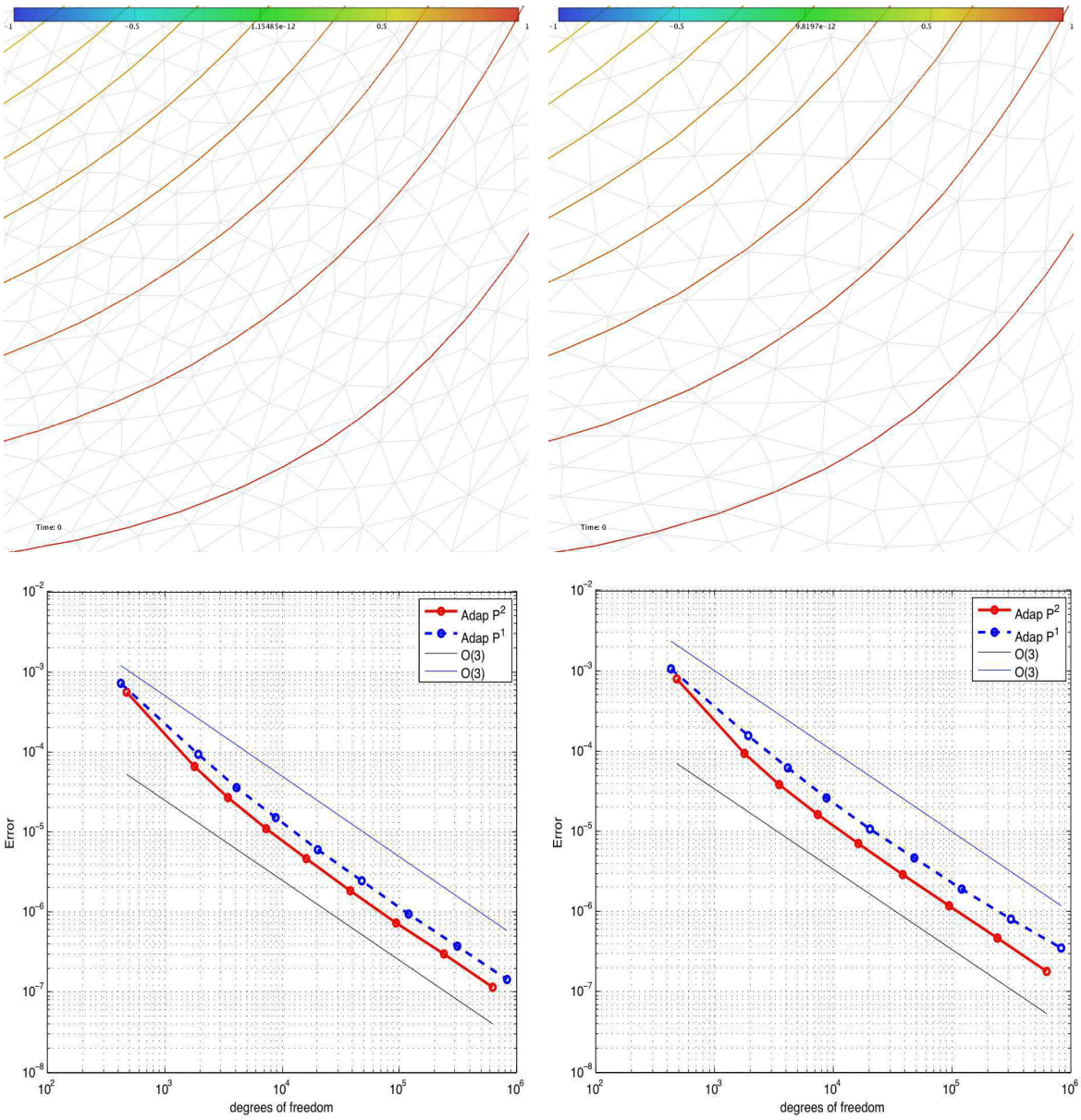

Figure 4.9: Closer view of the adapted meshes and the iso-values of f_4 for a $\mathbb{P}^1$-driven adaptation (top left) and a $\mathbb{P}^2$-driven adaptation (top right). Both meshes contains around 344 000 degrees of freedom. Convergence curves: $\mathbf{L}^1$-norm of the error versus the number of degrees of freedom (bottom left) and $\mathbf{L}^2$-norm of the error versus the number of degrees of freedom (bottom right)

Ratios et quotients d'anisotropie : $\mathbb{P}^1$-driven adaptation

Number of vertices	Number of tetrahedra	Min ratio	Max ratio	Average ratio
48062	231649	1.7175	1408.4613	37.7755

1	< rat <	2	2	0.00 %
2	< rat <	3	89	0.04 %
3	< rat <	4	616	0.27 %
4	< rat <	5	1720	0.74 %
5	< rat <	10	27753	11.98 %
10	< rat <	50	151256	65.30 %
50	< rat <	1e+02	38373	16.57 %
1e+02	< rat <	1e+03	11835	5.11 %
1e+03	< rat <	1e+04	5	0.00 %

Min quotient	Max quotient	Average quotient
2.7393	519791.3262	323.6046

1	< quo <	2	0	0.00 %
2	< quo <	3	5	0.00 %
3	< quo <	4	94	0.04 %
4	< quo <	5	295	0.13 %
5	< quo <	10	8795	3.80 %
10	< quo <	50	99834	43.10 %
50	< quo <	1e+02	50354	21.74 %
1e+02	< quo <	1e+03	63974	27.62 %
1e+03	< quo <	1e+04	7210	3.11 %
1e+04	< quo <	1e+05	1056	0.46 %
1e+05	< quo <	1e+06	32	0.01 %

Ratios et quotients d'anisotropie: $\mathbb{P}^2$-driven adaptation

Number of vertices	Number of tetrahedra	Min ratio	Max ratio	Average ratio
38375	179963	2.7329	4758.6071	66.3541

1	$<$ rat $<$	2	0	0.00 %
2	$<$ rat $<$	3	1	0.00 %
3	$<$ rat $<$	4	25	0.01 %
4	$<$ rat $<$	5	109	0.06 %
5	$<$ rat $<$	10	5438	3.02 %
10	$<$ rat $<$	50	116732	64.86 %
50	$<$ rat $<$	1e+02	30184	16.77 %
1e+02	$<$ rat $<$	1e+03	26984	14.99 %
1e+03	$<$ rat $<$	1e+04	490	0.27 %

Min quotient	Max quotient	Average quotient
4.0572	14124773.7689	4205.8381

1	$<$ quo $<$	2	0	0.00 %
2	$<$ quo $<$	3	0	0.00 %
3	$<$ quo $<$	4	0	0.00 %
4	$<$ quo $<$	5	4	0.00 %
5	$<$ quo $<$	10	304	0.17 %
10	$<$ quo $<$	50	51634	28.69 %
50	$<$ quo $<$	1e+02	39600	22.00 %
1e+02	$<$ quo $<$	1e+03	63131	35.08 %
1e+03	$<$ quo $<$	1e+04	21295	11.83 %
1e+04	$<$ quo $<$	1e+05	3105	1.73 %
1e+05	$<$ quo $<$	1e+06	758	0.42 %
1e+06	$<$ quo		132	0.07 %

Comments. The domain of the f_4 function is shifted in the x direction in order to have strong third-order derivatives. This example shows that we reach an optimal third-order of convergence with the third-order metric. Stronger anisotropy is generated with the third-order metric as reported in the previous table and **Figure 4.8**.

4.6 CONCLUSION

In this chapter, we have proposed an optimal multi-scale mesh adaptation in $\mathbf{L}^p$-norm for quadratic interpolation error. We have compared two mesh adaptations based respectively on $\mathbb{P}^1$ and $\mathbb{P}^2$-interpolations on $\mathbb{P}^2$-Lagrange representation of analytic functions. The optimal continuous mesh has been obtained from optimal local metrics, solutions of the local optimization problem solved in **Chapter 2** and based on tensor decompositions. Theoretical order of convergence has been observed on 2D and 3D numerical examples. In each case, the optimal third-order metric has a lower error bound than considering second-order derivatives and early asymptotic convergence.
Thereby, to summarize:

- Multi-scale mesh adaptation in 2D and 3D have been successfully extended to quadratic interpolation using symmetric tensor decomposition algorithms to approximate optimal local metrics.
- $\mathbb{P}^2$-driven mesh adaptation based on third order derivative-based model, realized on analytic functions, shows an asymptotic convergence order close to the third order that tends to be lost using $\mathbb{P}^1$-driven mesh adaptation based on the Hessian of the solution. The variations of analytic functions on $\mathbb{P}^2$-Lagrange meshes are also captured with quadratic interpolation.
- Thus, increasing the interpolation order of the solution on a given mesh requires increasing the order of the numerical methods to approximate the optimal global and local metrics for higher-order mesh adaptation.

Conclusion and perpectives

The goal of this thesis was to extend anisotropic mesh adaptation to higher-order interpolations based on theoretical and numerical results obtained in the linear case. To reach this goal, different steps have been followed:

• **Symmetric tensor decompositions**
Our approach consisted in modeling the higher-order interpolation error by an homogeneous polynomial or a symmetric tensor of degree $k \geqslant 3$ on each node of the computational domain. Then, we proceeded to a diagonalization of this initial error model using symmetric tensor decomposition algorithms. Two tensor decomposition algorithms have been studied: the $\mathbf{CP3}_{alsls}$ algorithm and Sylvester's algorithms. However, we noted that Sylvester's algorithms were the best method among both to get a diagonalization of the homogeneous error model. Indeed, it has been shown that the $\mathrm{CP3}_{alsls}$ method was not stable and didn't always converge to the desired solution in three dimensions. On the contrary, Sylvester's method, although it showed some insufficiencies in the decomposition process, is less expensive taking into account CPU time. It can also be corrected to approach the desired solution.

• **Construction of optimal third-order metrics**
Using the diagonalization of the error model obtained with tensor decomposition algorithms, we solved a local optimization problem to approximate the optimal local metric. This quadratic definite positive form approaches at best the variations of the initial error on each node on the computational domain. This idea is close to the one used in the linear case. These optimal local metrics are such that the criteria of: - consistency, - optimality and - choice of main directions are verified.

• **Extension of multi-scale metric based mesh adaptation to higher-order interpolations**
The solution of the local optimization problem allowed to solve the global optimization problem of finding the optimal metric field or continuous mesh minimizing the higher-order continuous interpolation error in $\mathbf{L}^p$-norm. It has been proved on various two-dimensional analytical examples that multi-scale mesh adaptation based on third-order interpolation error builds a sequence of anisotropic meshes whose numerical convergence is close to third-order. Thereby, the extension of Hessian-based mesh adaptation have been successfully extended to higher-order interpolations in 2D. However, concerning the 3D case, analytical example has proved that multi-scale mesh

adaptation could successfully be realized on simple analytical functions using symmetric tensor decomposition. But, when we use more complex analytical functions, the approximation of local metrics from symmetric tensor decomposition is not correct. Thus, adapted meshes are fake and then, the numerical convergence is far from the desired third-order.

4.7 ADVANTAGES AND DISADVANTAGES OF OUR APPROACH

Sylvester method seemed to be a good idea to approximate local optimal metrics and to get the final metric field. Indeed, in the case of mesh adaption based on linear interpolation, the diagonalization of Hessian matrix was at the core of the method. Therefore, it was quite natural to use a diagonalization process for the high order case. This decomposition method allowed to find the best directions of the local optimal metrics in 2D.
However, the extension of Sylvester method to 3D has been partially successful to generate adapted meshes and reached the desired convergence order because of the non uniqueness of the solution of the algorithm. Indeed, it has been proved that the decomposition rank of a binary polynomial of degree 2 is equal to 2, i.e, we have two linear terms, thus two possible directions in 2D. On the contrary, for a generic homogeneous polynomial of degree 3 (and more) in three variables, the decomposition rank is equal to 4 (and more) and the solution is not unique, cf. **Table 2.1**. In this case, it is very difficult to find precisely the directions of the desired local optimal metrics in 3D from the homogeneous error model, because of the lack of selection criteria that will help to choose the best decomposition among the proposed solutions.

4.8 PERSPECTIVES

To summarize, the goal of this thesis was to extend the "Hessian-based" classical approach to very high order interpolations in two and three dimensions. The extension to very high order interpolations or to other kind (like Uniform B-spline, NURBS, ...) requires taking into account the Jacobian of the element for the reconstruction of local derivatives and the normalization of metrics. However, the algorithms presented in this thesis for the quadratic approximation of a symmetric tensor remains valid.

Future work on this subject will mostly consist in:

-Validating our approach using higher-order numerical schemes. Indeed, two- and three-dimensional examples proposed in **Chapter 4** are analytical examples. Numerical applications will be realized by coupling our approach to high order computational fluid dynamics (CFD) codes like the CENO2 scheme [24] or the residual distribution schemes [1].

-Extending the proposed approach to higher-order mesh adaptation on curved

isoparametric elements. Indeed, all numerical tests realized in this thesis has been done on meshes with straight elements (triangles in 2D or tetrahedra in 3D). The main problem that we may be encountered in this case concerns the generation of curvilinear meshes rather than the error estimate.

A

Algebraic tools

In this appendix, we recall the algebraic tools we will need to describe and analyze **Algorithms 1 and 2**, cf. **Chapter 2**. These elements are presented in a simple manner without going into details. However, for more details about these algebraic tools, see [16, 17].

A.1 NOTATIONS

Let $\mathbb{K}$ be an algebraically closed field (e.g. $\mathbb{K} = \mathbb{C}$ the field of complex numbers). If $\alpha = (\alpha_1, ..., \alpha_n)$ is a vector in $\mathbb{N}^n$, then $|\alpha| = \sum_{i=1}^{n} \alpha_i$. We denote by $\mathbf{x}^\alpha$ the monomial $x_1^{\alpha_1} ... x_n^{\alpha_n}$.
Let R be the ring of polynomials $\mathbb{K}[x_1, ..., x_n] = \mathbb{K}[\underline{\mathbf{x}}]$, while R_k will denote the ring the polynomials of (total) degree at most k. The set $\{\mathbf{x}\}^\alpha_{|\alpha| \leqslant k} = \{x_1^{\alpha_1} ... x_n^{\alpha_n}\}_{\alpha_1 + ... + \alpha_n \leqslant k}$ represents the elements of the monomial basis of the vector space R_k and contains $\binom{n+k}{k}$ elements.
We denote by S the ring of polynomials $\mathbb{K}[x_0, ..., x_n] = \mathbb{K}[\mathbf{x}]$ and S_k the vector space of homogeneous polynomial in $n+1$ variables $x_0, ..., x_n$. This is also the symmetric k^{th} power $S^k(E)$ where $E = \left\langle x_0, ..., x_n \right\rangle$, a vector space.

A.2 DEHOMOGENIZATION

The *dehomogenization* of a polynomial $f \in S_k$ with respect to the variable x_0 is denoted $f^a := f(1, x_1, ..., x_n) \in R$.

A.3 DUALITY

For a $\mathbb{K}$-vector space E, its dual $E^* = Hom_{\mathbb{K}}(E, \mathbb{K})$ is the set of $\mathbb{K}$-linear forms from E to $\mathbb{K}$.
A basis of the dual space R_k^*, is the set of linear forms that compute the coefficients of a polynomial in the primal basis. It is denoted by $\{\mathbf{d}^\alpha\}_{|\alpha|\leqslant k}$.
We may identify R^* with the (vector) space of formal power series, i.e., $\mathbb{K}[[\mathbf{d}]] = \mathbb{K}[[d_1, ..., d_n]]$. Any element $\Lambda \in R^*$ can be decomposed as $\Lambda = \sum_{\mathbf{a}} \Lambda(\mathbf{x}^\mathbf{a})\mathbf{d}^\mathbf{a}$.

A.4 HANKEL OPERATORS

For any $\Lambda \in R^*$, we define the bilinear form Q_Λ, such that:

$$\begin{array}{rccc} Q_\Lambda : & R \times R & \longrightarrow & \mathbb{K} \\ & (a, b) & \longmapsto & \Lambda(ab). \end{array}$$

The matrix Q_Λ in the monomial basis of R is $\mathbb{Q}_\Lambda = (\Lambda(\mathbf{x}^{\alpha+\beta}))_{\alpha,\beta}$, where α, $\beta \in \mathbb{N}^n$.

For any $\Lambda \in R^*$, we define the *Hankel operator* H_Λ from R to R^* as

$$\begin{array}{rccc} H_\Lambda : & R & \longrightarrow & R^* \\ & p & \longmapsto & p \star \Lambda, \end{array}$$

with

$$\begin{array}{rccc} p \star \Lambda : & R & \longrightarrow & \mathbb{K} \\ & q & \longmapsto & \Lambda(pq). \end{array}$$

The matrix of the linear operator H_Λ in the monomial basis, and in the dual basis, $\{\mathbf{d}^\alpha\}$, is $\mathbb{H}_\Lambda = (\Lambda(\mathbf{x}^{\alpha+\beta}))_{\alpha,\beta}$, where α, $\beta \in \mathbb{N}^n$.

Definition A.1 *Given two sets $B = \{b_1, ..., b_r\}$, $B' = \{b'_1, ..., b'_{r'}\} \subset R$ and $\left\langle B \right\rangle$, $\left\langle B' \right\rangle$ their corresponding vector space. We define*

$$H_\Lambda^{B,B'} : \quad \left\langle B \right\rangle \longrightarrow \left\langle B' \right\rangle,$$

as the restriction of H_Λ to the vector space $\left\langle B \right\rangle$ and inclusion of R^* in $\left\langle B' \right\rangle^*$. Let $\mathbb{H}_\Lambda^{B,B'} = (\Lambda(b_i\, b'_j))_{1\leqslant i\leqslant r, 1\leqslant j\leqslant r'}$ the matrix of $H_\Lambda^{B,B'}$. If $B' = B$, we also use the notation H_λ^B and $\mathbb{H}_\Lambda^B$.

Definition A.2 *Given a symmetric tensor $f \in S_k$ and a natural number $0 \leqslant r \leqslant k$, we define*

$$\begin{array}{rccc} \mathcal{C}_f^r : & S_r & \longrightarrow & S_{k-r}^* \\ & p & \longmapsto & p \star f^*, \end{array}$$

where $f^* \in S_k^*$ *is the dual form of* f*, i.e., if* $f = \sum_{|\alpha|=k} c_\alpha \binom{k}{\alpha} \mathbf{x}^\alpha$*, then* f *is mapped to the linear form* $f^* = \sum_{|\alpha|=k} c_\alpha \, \mathbf{d}^\alpha$*.*

A catalecticant matrix or Hankel matrix of order r*, we note* $\mathbb{C}_f^r$*, is the matrix of* $\mathcal{C}_f^r$ *in the monomial basis of* S_r *and in the dual basis of the monomial basis of* S_{k-r}^**.*

REMARK A.1 A Hankel operator on a Hilbert space is one whose matrix with respect to an orthonormal basis is a (possibly infinite) Hankel matrix.

B

Matlab codes of tensor decomposition algorithms

This appendix gathers Matlab functions used to validate the different methods to approximate the variations of a homogeneous polynomial by the variations of a quadratic function.

B.1 BINARY FORM DECOMPOSITION

This is the Matlab code of **Algorithm 1** (cf. **Chapter 2**, **Section 2.3.1**) proposed to decompose a binary form.

```
function [mu,Q] = binarydec2(p)
% Decomposition of a generic binary polynomial p
% into the sum of Nth powers of linear polynomial p
% mu : vector of N coefficients
% Q : N by 2 matrices whose rows are the sought forms

s = 1; r = 0; d = length(p)-1; eta = 1.e-4;
fd = facto(d);
c = ones(1,d+1); p0 = p; p = p./c; v = [ ];

for i = 1:d-1
  c(i+1) = fd/(facto(i)*facto(d-1));
end

while s > eta & r < d-r+2
  r = r+1;
```

```
  M = hankel(p(1:d-r+1),p(d-r+1:d+1));
  [U,S,V] = svd(M);
  J = find(diag(S) < eta);
  if length(J) > 0
   s = S(J,J);
   J = J(1);
  elseif r+1> d-r+1;
   s = 0;
   J = r+1;
  end
end

v = V(:,J);
q = roots(v);
Q = [q,ones(length(q),1)];
mu = convd(Q,d)' \ p0';
sol = mu'*convd(Q,q);
W = diag(ones(1,d+1)./c);
% Output of the reconstruction error
err = sqrt((sol-p0)*W*(sol-p0)');
```

Matlab functions: facto and convd

```
function m = facto(n)
%m = n! (n factorielle)
if n == 0
  m = 1;
elseif n == 1
  m = 1;
elseif n == 2
  m = 2;
elseif (floor(n)-abs(n)) == 0
  m = n*facto(n-1);
else
  error('n must be a positive integer');
end

function P = convd(q,d)
%Raising of a polynomial q to the d^th power
[a, b] = size(q);
P=[ ];

for i = 1:a
 pd = q(i,:);
```

```
 for t = 1:d-1
  pd = conv(pd,q(i,:));
 end
 P = [P;pd];
end
```

B.2 SYMMETRIC TENSOR DECOMPOSITION

This is the partial Matlab code of **Algorithm 2** (cf. **Chapter 2**, **Section 2.3.2**) we proposed to decompose any third-order homogeneous polynomials in three variables as a sum of third power of three linear terms.

```
%Decompose of third-order homogeneous polynomials in three variables
deg = 3;

% p = Homogeneous polynomial written as a row vector
p = [p(1) , 3*p(2) , 3*p(3) , p(4) , 3*p(5) , 3*p(6) , p(7) , 3*p(8) , 3*p(9) , 6*p(10)];

% p1 = homogeneous polynomial without the binomial coefficients
p1 = [p(1) p(2) p(3) p(4) p(5) p(6) p(7) p(8) p(9) p(10)];

% Normalization of the homogeneous polynomial
if (p(1)  = 0)
  a = abs(p(1));
else
  a = max(abs(p1));
end

pmod = (1/a)*pol1;

a300 = pmod(1); a210 = pmod(2);
a120 = pmod(3); a030 = pmod(4);
a201 = pmod(5); a102 = pmod(6);
a003 = pmod(7); a021 = pmod(8);
a012 = pmod(9); a111 = pmod(10);

% Fix the rank equal to 3 % Basis: x*x xy xz;
Delta0 = [a300, a210, a201; a210, a120, a111; a201, a111, a102];
% Multiplied Delta0 by y/x
Delta1 = [a210, a120, a111; a120, a030, a021; a111, a021, a012];
% Multiplied Delta0 by z/x
Delta2 = [a201, a111, a102; a111, a021, a012; a102, a012, a003];
```

```
% Rank of the minor Delta0
k = rank(Delta0);

% Check if Delta is invertible
res = det(Delta0);

% Check if (res == 0)
 Delta0 = Delta0+Delta1;
 Somme = Delta1+Delta2;
else
 Delta0 = Delta0;
 Somme = Delta1;
end

A = Somme*inv(Delta0);

% Check the commutativity
My = inv(Delta0)*Delta1;
Mz = inv(Delta0)*Delta2;

MyMz = My*Mz;
MzMy = Mz*My;

% Solve the generalized eigenvalue problem
[v, vp] = eig(A);

% Normalize the generalized eigenvectors
mat = [v(:,1)/v(3,1), v(:,2)/v(3,2), v(:,3)/v(3,3)];

% Q matrix of normalized eigenvectors
Q = mat';
mu = a*diag(inv(mat)*Delta0*inv(mat)');
```

B.3 APPROXIMATION OF OPTIMAL LOCAL METRICS

B.3.1 Optimal local metrics in 2D

The following Matlab function shows how we numerically compute the optimal local metric from binary form decomposition.

```
function [met,U,V] = metopt2D(deg,p,Q)
% met = optimal local metric
% U, V= axis of metric met
% deg = order of the homogeneous polynomial
```

```
% p = homogeneous polynomial

iQ = inv(Q); % inverse of Q
u = iQ(:,1);
v = iQ(:,2);
Eu1 = abs(evalPoly2D(deg,p,u(1),u(2)));
Eu2 = abs(evalPoly2D(deg,p,v(1),v(2)));

h1 = (1/Eu1)^(1/deg);
h2 = (1/Eu2)^(1/deg);

U = h1*[u(1); v(2)];
V = h2*[v(1); v(2)];

met = Q'*diag([(1/(h1*h1) , 1/(h2*h2)])*Q;

% For PARAFAC model, to avoid imaginary part
met = real(met);

if (isreal(Qbis) == 0)
  met = (1/2)^1/3*real(met);
end
```

Matlab function: evalPoly2D

```
function e = evalPoly2D(deg,p,x,y)
% e = homogeneous polynomial written as a polynomial function
% e = p(1) x^3 + p(2) x^2y + p(3) xy^2 + p(4) y^3
% deg = order of the homogeneous polynomial
% p = homogeneous polynomial written as a row vector
% x, y = variables of the homogeneous polynomial

e = zeros(size(x));
for k = 0:deg
   e = e + p(k+1)*x.^(deg-k).*y.^k;
end
```

B.3.2 Optimal local metrics in 3D

The following Matlab function shows how we numerically compute the optimal local metric from Sylvester's algorithm extended to 3D.

```
function [met,U,V,W] = metopt3D(deg,p,Q)
% met = optimal local metric
% U, V, W = axis of metric met
% deg = order of the homogeneous polynomial
% p = homogeneous polynomial

iQ = inv(Q); % inverse of Q
u = iQ(:,1);
v = iQ(:,2);
w = iQ(:,3);
Eu1 = abs(evalPoly3D(deg,p,u(1),u(2),u(3)));
Eu2 = abs(evalPoly3D(deg,p,v(1),v(2),v(3)));
Eu3 = abs(evalPoly3D(deg,p,w(1),w(2),w(3)));

h1 = (1/Eu1)^(1/deg);
h2 = (1/Eu2)^(1/deg);
h3 = (1/Eu3)^(1/deg);

U = h1*[u(1); v(2): u(3)];
V = h2*[v(1); v(2); v(3)];
W = h2*[w(1); w(2); w(3)];

met = Q'*diag([(1/(h1*h1) , 1/(h2*h2), 1/(h3*h3)])*Q;

% For PARAFAC model, to avoid imaginary part
met = real(met);
```

Matlab function: evalPoly3D

```
function e = evalPoly3D(deg,p,x,y,z)
% e = homogeneous polynomial written as a polynomial function
% e = p(1) x^3 + p(2) x^2y + p(3) xy^2 + p(4) y^3 + p(5) x^2z + p(6) xz^2
%   + p(7) z^3 + p(8) y^2z + p(9) yz^2 + p(10) xyz
% deg = order of the homogeneous polynomial
% p = homogeneous polynomial written as a row vector
% x, y, z = variables of the homogeneous polynomial

u = zeros(size(x));
for k = 0:deg
   u = u + p(k+1)*x.^(deg-k).*y.^k;
end

v = zeros(size(x));
for k = 1:deg
```

```
    v = v + p(deg+1+k)*x.^(deg-k).*z.^k;
end

w = zeros(size(x));
for k = 1:deg-1
    w = w + p(2*deg+k+1)*y.^(deg-k).*z.^k;
end

e = u + v + w + p(3*deg+1).*x.*y.*z;
```

Bibliography

[1] R. Abgrall. *Residual distribution schemes : current status and future trends. Computers and Fluids*, 35:641–669, 2006.

[2] F. Alauzet and P.J. Frey. *Estimateur d'erreur géométrique et métrique anisotropes pour l'adaptation de maillages. Partie I : aspects théoriques. RR-4759, INRIA. (in French)*, March 2003.

[3] F. Alauzet, P.J. Frey, P.L. George, and B. Mohammadi. 3*D Transient Fixed-Point Mesh Adaptation for Time-Dependent Problems: Application to CFD simulations. J. Comp. Phys.*, 222:592–623, 2007.

[4] F. Alauzet and A. Loseille. *High Order Sonic Boom Modeling by Adaptive Methods. J. Comp. Phys.*, 229:561–593, 2010.

[5] F. Alauzet and A. Loseille. *Metrix User Guide. Error Estimates and Mesh Control Anisotropic Mesh Adaptation. Rapport technique, INRIA*, Février 2009.

[6] F. Alauzet, A. Loseille, A. Dervieux, and P.J. Frey. *Multi-dimensional continuous metric for mesh adaptation.* In *Proceedings of the 15th International Meshing Roundtable*, pages 191–214. Springer, 2006.

[7] O. Allain, D. Guegan, and F. Alauzet. *Studying the impact of unstructured mesh adaptation on free surface flow simulations.* In *Proceedings of the ASME 28th International Conference on Ocean, Offshore and Arctic Engineering*, 2009.

[8] F. Bassi ans S. Rebay. *High-order accurate discontinuous Finite Element solution of the* 2*D equations. J. Comp. Phys.*, 138(2):251–285, 1997.

[9] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. *Log-Euclidean metrics for fast and simple calculus on diffusion tensors. Magn. Reson. Med.*, 2(56):411–421, 2006.

[10] G. Karniadakis B. Cockburn and C.-W. Shu. *Discontinuous Galerkin methods: theory, computation and application.* Springer Verlag, Berlin, lecture notes in computational science and engineering edition, 2000.

[11] T.J. Baker. *Mesh adaptation strategies for problems in fluid dynamics. Finite Elem. Anal. Des.*, 25:243–273, 1997.

[12] Y. Belhamadia, A. Fortin, and E. Chamberland. *Three-dimensional anisotropic mesh adaptation for phase change problems. J. Comp. Phys.*, 201:753–770, 2004.

[13] A. Belme. *Aérodynamique instationnaire et méthode adjointe.* PhD thesis, Université de Nice Sophia Antipolis, Sophia Antipolis, France. (in French), 2011.

[14] G.M. Bergman. *Ranks of tensors and change of base fields. Journal of Algebra*, 11:613–621, 2004.

[15] C.L. Bottasso. *Anisotropic mesh adaption by metric-driven optimization. Int. J. Numer. Meth. Engng*, 60:597–639, 2004.

[16] J. Brachat. *Schémas de Hilbert et Décomposition de tenseurs.* PhD thesis, Université de Nice-Sophia Antipolis, France. (in French), 2008.

[17] J. Brachat, P. Comon, B. Mourrain, and E. Tsigaridas. *Symmetric tensor decomposition. Linear Algebra and Applications*, 433(11-12):1851–1872, 2010.

[18] R. Bro. *Parafac: Tutorial and Applications. Chemom. Intell. Lab. Syst.*, 38:149–171, 1997.

[19] J. Bruchon, H. Digonnet, and T. Coupez. *Using a signed distance function for the simulation of metal forming process: formulation of the contact condition and mesh adaptation. From Lagrangian approach to an Eulerian approach. Int. J. Numer. Meth. Engng*, 78(8):980–1008, 2009.

[20] G.C. Buscaglia and E.A. Dari. *Anisotropic Mesh Optimization and its Application in Adaptivity. Int. J. Numer. Meth. Engng*, 40:4119–4136, 1997.

[21] W. Cao. *On the error of linear interpolation and the orientation, aspect ration and internal angles of a triangle. SIAM J. Numer. Anal.*, 43(1):19–40, 2005.

[22] W. Cao. *An interpolation error estimate in $\mathcal{R}^2$ based on the anisotropic measures of higher derivatives. Math. Comp.*, 77:265–286, 2008.

[23] A. Carabias. *Analyse et adaptation de maillages pour des schémas non-oscillatoires d'ordre élevé.* PhD thesis, Université de Nice Sophia Antipolis, France. (in French), 2013.

[24] A. Carabias and A. Dervieux. *La revanche d'un schéma ENO d'order élevé sur les équations d'Euler en maillages non structurés adaptatifs. Rapport de recherche, INRIA*, Janvier 2001.

[25] J.D. Carrol and J.J. Chang. *Analysis of Individual Differences in Multidimensional Scaling via an n-way Generalization of "Eckart-Young" Decomposition. IPsychometrika*, 35:283–319, 1970.

[26] M.J. Castro-Diáz, F. Hecht, B. Mohammadi, and O. Pironneau. *Anisotropic Unstructured Mesh Adaptation for Flow Simulations. Int. J. Numer. Meth. Fluids*, 25:475–491, 1997.

[27] P. Chevalier, L. Albera, A. Ferreol, and P. Comon. *On the virtual way concept for higher order array processing. IEEE Trans. Sig. Proc.*, 53(4):1254–1271, April 2005.

[28] Ph. Clément. *Approximation by Finite element functions using local regularization. Revue Français d'Automatique, Informatique et Recherche Opérationnelle*, R-2:77–84, 1975.

[29] P. Comon. *Independent Component Analysis.* Elsevier, Amsterdam, London, in j-l. lacoume, editor, higher order statistics edition, 1992.

[30] P. Comon. *Tensor Decomposition. State of the Art and Applications. Mathematics in Signal Processinf V, J. G. McWhirter and I. K. Proudler Eds., Oxford University Press*, pages 1–24, 2002.

[31] P. Comon, G. Golub, L-H. Lim, and B. Mourrain. *Symmetric tensor and symmetric tensor rank. SIAM Journal on Matrix Analysis Appl.*, 30(3):1254–1279, 2008.

[32] P. Comon, X. Luciani, and A.L.F. Almeida. *Tensor Decompositions, Alternating Least Squares and other Tales. Jour. Chemometrics*, 23:393–405, August 2009.

[33] P. Comon and B. Mourrain. *Decomposition of quantics in sums of powers of linear forms. Signal Processing, Elsevier*, 53(2-3):93–107, 1996.

[34] G. Compère, E. Marchandise, and J.-F. Remacle. *Transient adaptivity applied to two-phase incompressible flows. J. Comp. Phys.*, 227:1923–1942, 2007.

[35] G. Compère, J.-F. Remacle, J. Jansson, and J. Hoffman. *A Mesh Adaptation Framework for Dealing with Large Deforming Meshes. Int. J. Numer. Meth. Engng.*, 82(7):843–867, 2010.

[36] T. Coupez. *Génération de maillages et adaptation de maillage par optimisation locale. Revue Européenne des Eléments Finis*, 9:403–423, 2000.

[37] E.F. D'Azevedo and B. Simpson. *On the optimal triangular meshes for minimizing the gradient error. Numer. Math.*, 59(4):321–348, 1991.

[38] C. Dobrzynski and P.J. Frey. *Anisotropic Delaunay Mesh Adaptation for Unsteady Simulations.* In *Proceedings of the 17th International Meshing Roundtable*, pages 177–194. Springer, 2008.

[39] J. Dompierre, M.G. Vallet, M. Fortin, Y. Bourgault, and W.G. Habashi. *Anisotropic mesh adaptation: towards a solver and user independent CFD.* In *AIAA 35th Aerospace Sciences Meeting and Exhibit, AIAA-1997-0861*, Reno, NV, USA, Jan 1997.

[40] L. Formaggia, S. Micheletti, and S. Perotto. *Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection-diffusion-reaction and the Stokes problems. Appl. Numer. Math.*, 51(4):511–533, 2004.

[41] L. Formaggia and S. Perotto. *New anisotropic a priori error estimates. Numer. Math.*, 89:641–667, 2001.

[42] M. Fortin, J. Dompierre, M.G. Vallet, Y. Bourgault, and W.g. Habashi. *Anisotropic mesh adaptation: theory, validation and applications.* In *Proceedings of ECCOMAS CFD*, 1996.

[43] P.J. Frey. *Yams, A fully automatic adaptive isotropic surface remeshing procedure. RT-0252, INRIA*, Nov 2001.

[44] P.J. Frey and F. Alauzet. *Anisotropic mesh adaptation for CFD computations. Comput. Methods Appl. Mech. Engrg.*, 194(48-49):5068–5082, 2005.

[45] P.J. Frey and H. Borouchaki. *Surface meshing using a geometric error estimate. Int. J. Numer. Methods Engng.*, 58(2):227–245, 2003.

[46] P.L. George, F. Hecht, and M.G. Vallet. *Creation of internal points in Voronoi's type method. Control and adaptation. Adv. Eng. Software*, 13(5-6):303–312, 1991.

[47] P.L. George, F. Hecht, and M.G. Vallet. *Gamanic3d, an adaptive anisotropic tetrahedral mesh generator. RT-0252, INRIA*, Nov 2003.

[48] C. Gruau and T. Coupez. *3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. Comput. Methods. Appl. Mech. Engrg.*, 194(48-49):4951–4976, 2005.

[49] D. Guégan, O. Allain, A. Dervieux, and F. Alauzet. *An L^∞-L^p mesh adaptive method for computing unsteady bi-fluid flows. Int. J. Numer. Meth. Engng*, 84(11):1376–1406, 2010.

[50] R. A. Harshman. *Fondations of the PARAFAC Prodedure: Models and Conditions for an "Explanatory" Multi-Modal Factor Analysis. UCLA Working Papers in Phonetics*, 1970.

[51] R. A. Harshman. *Determination and proof of minimum uniqueness conditions for PARAFAC1. UCLA Working Papers in Phonetics*, 22:111–117, 1972.

[52] F. Hecht. *BAMG: bidimensional Anisotropic Mesh Generator. Available from http://www-rocq.inria.fr/gamma/cdrom/www/bamg/eng.htm. INRIA-Rocquencourt, France*, 1998.

[53] F. Hecht. *Mesh generation and error indicator. Summer school: More efficiency in finite element methods*, 2008.

[54] F. Hecht and R. Kuate. *An approximation of anisotropic metrics from higher order interpolation error for triangle meshes adaptation. Journal of Computational and Applied Mathematics*, 258:99–115, 2014.

[55] F. Hecht and B. Mohammadi. *Mesh adaptation by metric control for multi-scale phenomena and turbulence.* In *35th Aerospace Sciences Meeting and Exhibit, AIAA-1997-0859*, Reno, NV, USA, Jan 1997.

[56] W. Huang. *Metric tensors for anisotropic mesh generation. J. Comp. Phys.*, 204(2):633–665, 2005.

[57] T. Jiang and N. Sidiropoulos. *Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear models. IEEE Trans. Sig. Proc.*, 52(9):2625–2636, Sep 2004.

[58] W.T. Jones, E.J. Nielsen, and M.A. Park. *Validation of 3D Adjoint Based Error Estimation and Mesh Adaptation for Sonic Boom Reduction.* In *44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2006-1150*, Reno, NV, USA, Jan 2006.

[59] H.A.L. Kiers and W.P. Krijnen. *An efficient algorithm for PARAFAC of three-way date with large numbers of observation units. Psychometrica*, 56:147, 1991.

[60] J.B. Kruskal. *Three-way arrays: Rank and uniqueness of trilinear decompositions. Linear Algebra and Applications*, 18:95–138, 1977.

[61] J.-F. Lagüe and F. Hecht. *Optimal mesh for P_1 interpolation in H^1 semi norm.* In *Proceedings of the 15th International Meshing Roundtable*, pages 259–270, Birmingham, Al, USA, 2006. Springer.

[62] L. De Lathauwer. *Signal Processing Based on Multilinear Algebra.* PhD thesis, K. U. Leuven, E. E. Departement -ESAT, Belgium, 1997.

[63] L. De Lathauwer and J. Castaing. *Tensor-based techniques for the blind separation of DS-CDMA signals. Signal Processing*, 87(2):322–336, Feb 2007.

[64] P. Laug and H. Borouchaki. *BL2D-V2, Mailleur bidimensionnel adaptatif. RT-0275, INRIA*, 2003.

[65] E. Leurgans, R. T. Ross, and R.B. Abel. *A decomposition for three-way arrays. SIAM J. Matrix Anal. Appl.*, 14:1064–1083, 1993.

[66] X. Li, M.S. Shephard, and M.W. Beal. *3D anisotropic mesh adaptation by mesh modification. Comput. Methods. Appl. Mech. Engrg.*, 194(48-49):4915–4950, 2005.

[67] R. Löhner. *Adaptative Remeshing for Transient Problems. Comput. Methods Appl. Mesh. Engrg.*, 75(1-3):195–214, 1989.

[68] R. Löhner. *Three-Dimensional Fluid-Structure Interaction Using a Finite Element Solver and Adaptive Remeshing. Computing Systems in Engineering*, 1(2-4):257–272, 1990.

[69] A. Loseille. *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, France. (in French), 2008.

[70] A. Loseille and F. Alauzet. *Continuous mesh framework, Part I: well-posed continuous interpolation error and Part II: validations and applications. SIAM Journal in Numerical Analysis*, 49(issue 1), 2010.

[71] A. Loseille and F. Alauzet. *Fully Anisotropic Goal-Oriented Mesh Adaptation for 3D Steady Euler Equations. J. Comp. Phys.*, 229(8):2866–2897, 2010.

[72] A. Loseille and R. Löhner. *Adaptive Anisotropic Simulations in Aerodynamics.* In *48th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2010-169*, Orlando, FL, USA, Jan 2010.

[73] D.J. Mavriplis. *Adaptive mesh generation for viscous flows using Delaunay triangulation. J. Comp. Phys.*, 90:271–291, 1990.

[74] J-M. Mirebeau. *Optimal Meshes For Finite Elements Of Arbitrary Order. Springer Science+Business Media, LLC 2010*, Feb. 2010.

[75] D. Nion and L. De Lathauwer. *An Enhanced Line Search Scheme for Complex-Valued Tensors Decompositions. Application in DS-CDMA. Signal Processing*, 88(3):749–755, March 2008.

[76] C.C. Pain, A.P. Humpleby, C.R.E. de Oliveira, and A.J.H. Goddard. *Tetrahedral mesh optimization and adaptivity for steady-state and transient finite element calculations. Comput. Methods Appl. Mech. Engrg.*, 190:3771–3796, 2001.

[77] J. Peraire, J. Peiro, and K. Morgan. *Adaptative Remeshing for Three-Dimensional Compressible Flow Computations. J. Comp. Phys.*, 103:269–285, 1992.

[78] J. Peraire, M. Vahdati, K. Morgan, and O.C. Zienkiewicz. *Adaptative Remeshing for Three-Dimensional Compressible Flow Computations. J. Comp. Phys.*, 72:449–466, 1987.

[79] M. Picasso. *An anisotropic error indicator based on Zienkiewicz-Zhu error estimator: Application to elliptic and parabolic problems. SIAM J. Sci. Comput.*, 24(4):1328–1355, 2003.

[80] M. Rajih, P. Comon, and R.A. Harshman. *Enhanced Line Search: A novel method to accelerate PARAFAC. SIAM Journal on Matrix Analysis and Applications*, 30(3):1148–1171, 2008.

[81] J.-F. Remacle, X. Li, M.S. Shephard, and J.E. Flaherty. *Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods. Int. J. Numer. Meth. Engng*, 62:899–923, 2005.

[82] B. Reznick. *Suns of powers of complex linear forms. Preprint*, Aug. 1992.

[83] V. Selmin. *Simulation of hypersonic flows on unstructured grids. Int. J. Numer. Meth. Engng*, 34:569–606, 1992.

[84] N.D. Sidiropoulos, , and R. Bro. *On the uniqueness of multilinear decomposition of n-way arrays. Journal of Chemometrics*, 14:229–239, 2000.

[85] N.D. Sidiropoulos, G.B. Giannakis, and R. Bro. *Blind PARAFAC Receivers for DS-CDMA Systems. IEEE Trans. Signal Processing*, 48:810–823, 2000.

[86] A. Smilde, R. Bro, and P. Geladi. *Multi-way Analysis. Applications in the chemical sciences. John Wiley and Sons*, 2004.

[87] A. Smilde, R. Bro, and P. Geladi. *Rank and optimal computation of generic tensors. Linear Algebra and Applications*, 52:645–685, July 1983.

[88] A. Swami, G. Giannakis, and S. Shamsunder. *Multichannel ARMA processes. IEEE Trans. Sig. Proc.*, 42(4):898–913, April 1994.

[89] A. Tam, D. Ait-Ali-Yahia, M.P. Robichaud, M. Moore, V. Kozel, and W.G. Habashi. *Anisotropic mesh adaptation for 3D flows on structured and unstructured grids. Comput. Methods Appl. Mech. Engrg*, 189:1205–1230, 2000.

[90] D.A. Venditti and D.L. Darmofal. *Anisotropic grid adaptation for functional outputs application to two-dimensional viscous flows. J. Comp. Phys.*, 187(1):22–46, 2003.

[91] M. Yano and D.L. Darmofal. *An Optimization framework for anisotropic simplex mesh adaptation: Application to aerodynamic flows.* In *50th AIAA Aerospace Sciences Meeting*, Nashville, TN, Jan 2011.

[92] M. Yano, J.M. Modisette, and D.L. Darmofal. *The importance of mesh adaptation for higher-order discretizations of aerodynamic flows.* In *20th AIAA CFD Conference, AIAA-2011-3852*, Honolulu, HI, June 2011.

[93] O.C. Zienkiewicz and J. Wu. *Automatic directional refinement in adaptive analysis of compressible flows. Int. J. Numer. Meth. Engng*, 37:2189–2210, 1994.

Mesh adaptation for very high order numerical schemes

Abstract: Mesh adaptation is an iterative process which consists in changing locally the size and orientation of the mesh according the behavior of the studied physical solution. It generates the best mesh for a given problem and a fix number of degrees of freedom. Mesh adaptation methods have proven to be extremely effective in reducing significantly the mesh size for a given precision and reaching quickly an second-order asymptotic convergence for problems containing singularities when they are coupled to high order numerical methods.
In metric-based mesh adaptation, two approaches have been proposed: Multi-scale methods based on a control of the interpolation error in L^p-norm and Goal oriented methods that control the approximation error of a functional through the use of the adjoint state. However, with the emergence of very high order numerical methods such as the discontinuous Galerkin method, it becomes necessary to take into account the order of the numerical scheme in mesh adaptation process. Mesh adaptation is even more crucial for such schemes as they converge to first-order in flow singularities. Therefore, the mesh refinement at the singularities of the solution must be as important as the order of the method is high.
This thesis deals with the extension of the theoretical and numerical results getting in the case of mesh adaptation for piecewise linear solutions to high order piecewise polynomial solutions. These solutions are represented using k^{th}-order Lagrangian finite elements ($k \geqslant 2$). This thesis will focus on modeling the local interpolation error of order $\mathbb{P}^{k+1}$ ($k \geqslant 2$) on a continuous mesh. However, for metric-based mesh adaptation methods, the error model must be a quadratic form, which shows an intrinsic metric space. Therefore, to be able to produce such an area, it is necessary to decompose the homogeneous polynomial and to approximate it by a quadratic form taken at power $\frac{k}{2}$. This modeling allows us to define a metric field necessary to communicate with the mesh generator. The decomposition method will be an extension of the diagonalization method to high order homogeneous polynomials. Indeed, in 2D and 3D, symmetric tensor decomposition methods such as Sylvester decomposition and its extension to high dimensions will allow us to decompose locally the error function, then, to deduce the quadratic error model. Then, this local error model is used to control the overall error in $\mathbf{L}^p$-norm and the optimal mesh is obtained by minimizing this error.
In this thesis, we seek to demonstrate the k^{th}-order convergence of high order mesh adaptation method for analytic functions and numerical simulations using k^{th}-order solvers ($k \geqslant 3$).

Keywords: Unstructured mesh adaptation, a priori error estimates, high order interpolation error, high order multi-scale adaptation, symmetric tensor decomposition, computation fluid dynamics simulations.

UPMC
SORBONNE UNIVERSITÉS

Bureau d'accueil des doctorants
15 rue de l'Ecole de Médecine
75006 PARIS
Tél. 01 44 27 28 10

Doctorat de l'Université PARIS 6

Spécialité :

SCIENCES MATHEMATIQUES DE PARIS CENTRE

RAPPORT de SOUTENANCE de THESE

Thèse soutenue le 20 Décembre 2013

Par **Mlle MBINKY, ESTELLE CARINE**

Sujet de la thèse

Adaptation de maillages pour des schémas numériques d'ordre très élevé

Jury M. PICASSO
M. COUPEZ
M. LOSEILLE
M. GEORGE
M. DERVIEUX
M. FREY

Rapport de soutenance

(utiliser le verso de ce document pour le rapport de soutenance)

Mention accordée au candidat *
par le jury : TRES HONORABLE

Paris, le 20/12/2013
Le président et les membres du jury :

*

L'université Pierre et Marie Curie, conformément à la décision du conseil scientifique du 8 novembre 2010, validée par le conseil d'administration du 29 novembre 2010, décide de ne plus délivrer que la mention " très honorable ".

Paris, le 20 décembre 2013

Le jury a apprécié l'effort de présentation didactique du travail de thèse de Estelle Corinne MAINKY. Elle a su clairement mettre en évidence ses contributions, par une approche nouvelle, à un sujet d'actualité : l'adaptation de maillages basés sur des métriques.

Les membres du jury ont particulièrement apprécié les contributions numériques et algorithmiques que Estelle Corinne Mainky a su intégrer dans un code utilisé par les membres de l'équipe.

Les réponses pertinentes aux questions du jury confirment sa maîtrise des concepts mathématiques et sa connaissance des enjeux de l'analyse numérique de ces problèmes.

Pour ces raisons, le jury prononce l'admission et accorde le grade de docteur de l'Université Pierre et Marie Curie, spécialité Sciences Mathématiques avec la mention très Honorable.

Pascal Frey.