Titre: Title:	Metric-based Mesh Adaptation with Curvilinear Triangles
Auteur: Author:	Arthur Bawin
Date:	2024
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Bawin, A. (2024). Metric-based Mesh Adaptation with Curvilinear Triangles [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/59198/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/59198/
Directeurs de recherche: Advisors:	André Garon, & Jean-François Remacle
Programme: Program:	Génie mécanique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

et

UNIVERSITÉ CATHOLIQUE DE LOUVAIN

Metric-based mesh adaptation with curvilinear triangles

ARTHUR BAWIN
Département de génie mécanique

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie mécanique

Juillet 2024

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

et

UNIVERSITÉ CATHOLIQUE DE LOUVAIN

Cette thèse intitulée :

Metric-based mesh adaptation with curvilinear triangles

présentée par **Arthur BAWIN**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Sandra SOARES-FRAZÃO, présidente
André GARON, membre et directeur de recherche
Jean-François REMACLE, membre et codirecteur de recherche
André FORTIN, membre
Ricardo CAMARERO, membre
Vincent LEGAT, membre
Adrien LOSEILLE, membre externe

REMERCIEMENTS

Mes premiers remerciements vont à mes deux co-directeurs, André Garon et Jean-François Remacle. Ayant passé plus de temps au Québec qu'en Belgique durant la thèse, j'ai eu davantage l'occasion de travailler avec André, mais Jean-François était toujours à portée d'un appel vidéo. Merci à vous deux: André pour ta disponibilité, tes idées et encouragements, et pour les sessions de grattage de papier et de tableaux; Jean-François pour ton soutien et ton enthousiasme infaillibles.

Je remercie chaleureusement André Fortin, Ricardo Camarero, Vincent Legat et Adrien Loseille pour l'intérêt qu'ils ont témoigné envers ma thèse en acceptant de faire partie du jury. Merci d'avoir pris le temps d'évaluer mon travail et pour vos commentaires qui en sont ressortis. Un grand merci également à Sandra Saores-Frazão d'assurer la présidence du jury.

Pour le temps qu'il m'a consacré, je tiens à remercier Marc Laforest, qui a toujours été disponible pour une explication ou une mise en contexte des sujets qui m'intéressaient, et de qui j'ai beaucoup appris.

Je remercie chaleureusement Lucien Rochery pour les conversations constructives que nous avons eues, et qui ont aiguillé plusieurs résultats de cette thèse. Merci également à Jean-Marie Mirebeau pour ses réponses et indications.

Viennent ensuite les habitants de l'Euler, avec qui j'ai passé de très bons mais trop courts moments. Merci aux doctorants, post-docs, chercheurs et secrétaires pour l'ambiance de travail agréable, les conversations enrichissantes et les parties de cartes de midi. En vrac et de manière non exhaustive : Alexis, Célestin, Jeanne, Kilian, Pierre-Alexandre, Christos, Maxence, Jovana, Alexandre, Ruili, Ruiyang, Amaury, Ange, Nathan, Insaf, Chiheb, David, Matthieu, Astrid et François. J'aimerais remercier particulièrement Jonathan Lambrechts pour ses conseils et son aide.

À Montréal, j'aimerais remercier chaleureusement Simon et Baptiste, mes collègues de bureau. Merci pour votre bonne humeur, vos coups de mains et pour les bons moments. Merci à Sujaat pour la bonne compagnie durant les pauses café et de midi. Merci évidemment à Michel et Solène, qui sont là depuis le début : c'est pas très sympa de m'avoir laissé finir tout seul, mais je ne vous en tiens pas rigueur, ne vous tracassez pas.

Je suis reconnaissant envers Toshiro Matsumoto qui m'a accueilli à Nagoya, ainsi qu'à tout son laboratoire pour leur accueil chaleureux. Un grand merci pour cette belle expérience.

Même si c'est peu commun, j'aimerais exprimer ma reconnaissance envers la communauté

de chercheurs, professeurs et étudiants de `math.stackexchange.com`, et les remercier pour le temps et l'énergie mis pour expliquer, illustrer et reformuler des notions plus ou moins abstraites et permettre à des non-mathématiciens (comme moi) de se les approprier.

Finalement, merci à ma famille, belle-famille et aux copains de toujours : Benjamin, Sébastien, Bastien, Emilio et Corentin.

Ces derniers mots sont pour Sarah, qu'il est impossible de remercier assez et avec qui le temps passe trop vite. Merci pour ta bonne humeur et ton soutien indéfectibles, et, même si c'est présomptueux de ma part, merci de faire de moi une meilleure personne.

Arthur Bawin

RÉSUMÉ

Les maillages courbes, ou d'ordre élevé, sont utilisés depuis plusieurs dizaines d'années comme support pour simulations par éléments finis ou volumes finis. La flexibilité inhérente aux éléments courbes a traditionnellement été mise à profit pour fournir une meilleure résolution des frontières CAD que celle offerte par des discrétisations linéaires par morceaux. Les maillages d'ordre élevé ont gagné en intérêt au cours de la dernière décennie, puisqu'il a été montré que les schémas numériques d'ordre élevé, bénéficiant d'une convergence accélérée par rapport aux schémas classiques d'ordre deux, pouvaient nécessiter des discrétisations géométriques également d'ordre élevé pour ne pas perdre leurs propriétés d'approximation. Ces maillages sont courbés a posteriori, en plaçant les sommets d'ordre élevé sur la géométrie CAD. Courber uniquement les éléments aux frontières peut les rendre invalides, cependant. Ils sont alors "démêlés", propageant ainsi de la courbure dans les éléments intérieurs. La courbure des éléments intérieurs n'est donc jamais activement recherchée, mais est la conséquence de la procédure de démêlage. Récemment, l'idée est apparue de mettre à profit ces degrés de liberté supplémentaires pour réduire l'erreur d'approximation dans le domaine de calcul, donnant lieu à l'adaptation de maillages curvilignes. Bien que des éléments courbés arbitrairement impactent négativement la convergence des méthodes d'interpolation, il a été montré que des maillages courbes optimisés apportaient un gain d'erreur substantiel, motivant leur utilisation pour l'adaptation de maillages.

Cette thèse est la continuité de travaux récents dans cette direction et aborde le problème de l'adaptation de maillages curvilignes en deux dimensions à l'aide de métriques. Plus particulièrement, le problème qui nous occupe est de générer, à l'aide de métriques riemanniennes, des maillages de triangles quadratiques minimisant l'erreur d'interpolation sur un champ scalaire. Cela requiert d'étendre les questions usuelles de l'adaptivité, c'est-à-dire, quantifier l'erreur commise en interpolant sur des éléments courbes, obtenir la métrique minimisant cette estimé d'erreur, décrire les propriétés des simplexes idéaux pour cette métrique, et enfin générer des triangulations courbes composées uniquement d'éléments idéaux. Chacune de ces questions est abordée de manière plus ou moins approfondie. Un récent estimateur d'erreur d'interpolation linéaire est d'abord étendu pour traiter des interpolants d'ordre arbitraire. De là, le problème à résoudre pour la métrique optimale est formulé, et un schéma numérique rudimentaire est proposé. La question des simplexes idéaux est traitée en proposant une nouvelle définition d'*éléments unités*, basée sur des isométries riemanniennes. Cette définition englobe les précédentes. Finalement, un algorithme de génération de maillage fournissant des triangulations courbes et quasi-idéales pour une métrique donnée est présenté.

ABSTRACT

Curvilinear, also called high-order, meshes have been used for decades as a support for numerical simulations with finite element and finite volume methods. The added flexibility of high-order elements has historically been used to provide a more accurate representation of curved CAD boundaries than that offered by piecewise linear meshes. High-order meshes have gained interest in the last decade, as it has been shown that high-order numerical schemes, which enjoy accelerated convergence compared to classical second-order methods, may require high-order discretizations so as not to lose their approximation properties. Such body-fitted meshes are curved a posteriori by placing high-order vertices on the geometry. Curving boundary elements only may cause them to become invalid, i.e., tangled. Untangling methods cause edge or face curvature to be propagated inside the domain. Thus, the curvature of interior elements is never sought; it appears instead as a by-product of the untangling procedure, and the ideal element is always straight. Recently, the idea emerged of using the additional degrees of freedom provided by high-order meshes to further reduce the approximation error inside the computational domain. This is the topic of high-order anisotropic mesh adaptation, or simply curvilinear mesh adaptation. The classical theory of interpolation discourages arbitrarily curved elements, as they negatively impact the convergence rate of interpolation operators. It has been shown, however, that optimized curvilinear meshes may actually provide a substantial reduction in interpolation error, motivating their use in adaptivity.

This thesis builds upon recent work in this direction and tackles the problem of metric-based curvilinear mesh adaptation in two dimensions. Specifically, we are concerned with the problem of generating, with the means of Riemannian metrics, meshes of quadratic triangles which minimize the interpolation error on a scalar field. This requires extending the classical questions of adaptivity, namely, quantifying the error committed on curved elements, deriving the Riemannian metric minimizing this error, describing the properties of ideal triangles for this metric, and finally, generating curved triangulations consisting of these ideal simplices. Each of these questions is addressed in more or less depth. We extend a recent linear error estimate to handle interpolation of arbitrary order on curved elements, then formulate the problem for the optimal metric, for which we provide a crude numerical scheme. The question of ideal simplices is treated by proposing a new definition of so-called *unit elements* based on Riemannian isometries, which encompasses the existing definitions. Finally, a meshing algorithm that outputs quasi-ideal triangulations for a given metric is presented.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

PDE	Partial differential equation
CAD	Computer-aided/assisted design
CFD	Computational fluid dynamics
SPD	Symmetric positive-definite

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

Real-life experiments in various disciplines, such as engineering, physics, or climatology, have been increasingly replaced by numerical simulations. These simulations involve translating the problem at hand into mathematical models, typically in the form of partial differential equations (PDE) in various fields (temperature, fluid velocity, material displacement, etc.), which are in turn discretized and solved on computers. In mechanical design, for instance, the process begins with creating a computer-assisted design (CAD) model of the studied component. When simulation is carried out with a finite element or finite volume method, the numerical solution is computed on a mesh, that is, a geometric discretization of the domain of interest. The quality of the mesh greatly impacts the precision and the robustness of the simulation. The initial mesh is usually not optimal, and is iteratively adapted based on the computed solution. This process can be repeated until convergence of both the mesh and the solution, from which useful design quantities (heat fluxes, lift and drag, various stresses, etc.) are then post-processed.

Meshes can be adapted by relocating the vertices to minimize an error functional of interest. Over the past few decades, *anisotropic* mesh adaptation has proven to be a cost-effective way of capturing complex simulation features, by creating meshes characterized by stretched elements in the directions of anisotropy of the problem at hand. Their use is particularly justified for strongly anisotropic phenomena, such as shocks and other complex fluid dynamics applications. Consequently, there is a rich literature regarding anisotropic mesh adaptation, especially for computational fluid dynamics (CFD) [1–7]. *Metric-based* adaptation is probably the most popular framework for the generation of anisotropic meshes: it relies on Riemannian metrics to modify how lengths and angles are computed, so that anisotropic and distorted meshes are viewed as isotropic and uniform for a given metric. The continuous mesh framework [8, 9] establishes a duality between discrete triangulations and Riemannian metrics, and derives optimal metrics for different error functionals, such as the interpolation error measured in L^p norm, or the error on specific quantities of interest. Anisotropic triangulations adapted for the former capture the overall features of the studied fields with an emphasis on the small scales depending on the chosen L^p norm, while meshes adapted to quantities of interest provide accurate predictions of outputs used in design, such as the lift and drag of an airfoil or the concentration of a tracer at a given location. This framework was initially developed for linear interpolation on straight-sided meshes, in both two and three dimensions [1], and was then extended to interpolation of arbitrary order on straight meshes [10–12].

Aside from mesh adaptation, high-order, or high-fidelity, numerical schemes, such as spectral and Discontinuous Galerkin (DG) methods, can be used to better approximate the governing equations, providing more accurate results for the same computational cost. It has been shown [13, 14], however, that high-order schemes require high-order (that is, curvilinear) meshes to fully take advantage of their inherent accelerated convergence. Using piecewise linear approximations of curved boundaries, for instance, can hinder the optimal convergence of such high-order methods. In this sense, the majority of the research on curvilinear meshes has been devoted to the generation of body-fitted meshes, whose curved elements are placed close to the boundaries to accurately capture the CAD model, to leverage these high-order numerical schemes. *Validity*, a constraint needed to perform meaningful interpolation and integration, becomes an issue with the elements of a high-order mesh. Linear elements are valid if they simply have a positive area, that is, if they are not inverted. For quadratic and higher-order elements, validity requires the strict positivity of their Jacobian determinant everywhere on the element. Assessing the validity of merely quadratic meshes is not an easy task [15], and neither is enforcing validity on all elements [16, 17]. State-of-the-art methods create body-fitted curved meshes from a linear mesh, snap the high-order vertices on the CAD boundaries, then untangle the curved elements by propagating curvature inside the domain [16, 18–20]. In these approaches, interior curvature is never sought and is only a by-product of the untangling of boundary elements, so that the ideal element is still the straight underlying triangle.

Recently, the idea of curving the mesh elements inside the domain emerged as a way of reducing the approximation error, in the same way that anisotropic meshes reduce the error compared to uniform or isotropic meshes [17, 21–30]. This was done incrementally in Zhang et al. [21–23] by proposing a meshing algorithm to produce valid curvilinear triangulations for a given Riemannian metric, then by devising such a metric from an input field, be it analytical or a numerical solution coming from CFD computations. In Aparicio-Estrems et al. [25–27], interior curvature is achieved by generating triangles and tetrahedra which are the image through the metric of an ideal equilateral simplex. The initially straight simplices are curved to minimize a metric-weighted distortion with respect to this ideal simplex. Sanjaya [28, 29], in contrast, builds upon the framework developed by Yano and Darmofal [31], and proposed to move the high-order vertices to locally minimize an error estimate for a quantity of interest. The work of Rochery [17, 24, 32, 33] aimed at extending the established continuous mesh framework to curvilinear mesh adaptation. Key ingredients of the method, such as error estimates for linear interpolation and a generalization of ideal

elements, were modified for curved simplices. The approach taken by Zhang falls within this framework, generalizing the *error–metric–ideal element* paradigm discussed in detail hereafter. Lastly, Caplan et al. [30] create curvilinear meshes of arbitrary order as projections of uniform meshes isometrically embedded in a higher-dimensional space, building on similar ideas for anisotropic mesh generation [34, 35].

This thesis aims at extending the continuous mesh framework to curvilinear adaptation, and thus builds on the work of Zhang [23], Rochery [17] and, to a lesser extent, Aparicio-Estrems [27]. Similarly to Zhang, our work is driven by three research questions, which are now motivated.

Research questions

Mesh adaptation is the process of translating **error indicators** into **size fields**, then converting size fields into **meshes**. Error indicators are post-processed from a numerical solution and drive the adaptation procedure. Mesh adaptation combines both *a priori* and *a posteriori* analyses: *a priori* estimates predict the error function associated with a given problem and discretization. These estimates study the underlying structure of the problem at hand, allowing to devise strategies to adapt the mesh to minimize the error indicator. The interpolation error over smooth functions, for instance, relies on Taylor's remainder and involves both high-order derivatives of the interpolated field and the local mesh size. The implementation of these adaptation strategies is made possible by *a posteriori* indicators, which quantify the error effectively obtained on the current discretization, allowing to compute size fields. Then, the mesh generator creates a geometric discretization that respects this size field as best as possible, according to a given criterion.

For some adaptation procedures, some of these aspects are trivial. For instance, in a tree-type mesh (e.g., quadtree/octree), it is common to compute a scalar error indicator and refine (resp. coarsen) the tree cells whose error lies within the top (resp. bottom) X% bracket, with the goal of distributing the error equally. Both the size field and mesh modification processes are trivial: either do nothing, merge neighbouring cells or divide a cell by 2^n. Moreover, these three aspects may be clearly separated, or, on the contrary, tightly linked. For instance, the anisotropic mesh adaptation procedures proposed by Bois et al. [36] and Fortin et al. [6, 37] rely on local mesh modifications to control the L^2 and H^1 norms of the interpolation error. Therefore, the size prescription and remeshing steps are coupled in an optimization problem. In metric-based frameworks, on the other hand, the (possibly anisotropic) size field is encoded in a Riemannian metric containing the ideal sizes and orientation at all points. The size field

is a distinct entity, over which operations such as smoothing or derivatives computations can be performed.

In this thesis, we consider **metric-based** mesh adaptation and we aim at generating two-dimensional meshes made of **curvilinear triangles**. As a first step, we focus on triangulations tailored to control the **interpolation error**. The characterization of the three given steps, error indicators, size fields and mesh generation, can be summarized for metric-based methods by the following questions:

1. **Error indicators:** How can we write local estimates for the interpolation error on simplices?
2. **Optimal size fields:** What are the Riemannian metrics minimizing the integral (e.g. L^p or $W^{s,p}$) norm of these estimates under a given constraint on the number of mesh vertices or elements?
3. **Mesh generation:** How can we generate meshes made of simplices whose size and orientation are prescribed by these metrics?

We build upon the **continuous mesh framework**, which answers these questions for straight-sided meshes. Fully extending the continuous mesh framework to high-order simplices is still an open problem, with current generalizations due to Rochery [17]. The following details the existing solution on linear meshes, and the state of the art for quadratic simplices:

1. **Error indicators:** Error estimates for linear and high-order interpolation error on linear simplices are based on Taylor's remainder. For linear interpolation, they involve the edges of the simplex and the Hessian of the function. To work in a fully continuous setup, discrete estimates are translated into continuous ones, involving only Riemannian metrics and no longer referring to discrete mesh edges. This is done using invariants shared by ideal simplices, also called **unit simplices**, for a given metric. These simplices are equilateral when viewed through the eyes of the metric, and can be described only with the information of the metric. More precisely, all instances of the edges in the linear error estimate can be replaced by the metric, no longer bounding the estimate to a particular simplex and making it a pointwise error estimate. Higher-order interpolation error estimates are derived for linear simplices by bounding the derivatives of order $m \geq 3$ around each mesh vertex by a positive-definite quadratic form. Then, the pointwise error estimate has the same form as for linear interpolation. A closed-form solution for this quadratic form exists for linear and quadratic interpolation in 2D and

is due to Mirebeau [38]. For linear interpolation, it is simply the absolute value of the Hessian matrix, so that the error estimate reduces to the one described above. For quadratic interpolation, a closed-form solution depending on the homogeneous polynomial describing the error around a mesh vertex exists. For higher-order interpolation, numerical optimization is required. Coulaud and Loseille proposed a method based on the simplex algorithm [11, 12] to derive this quadratic form.

Error estimates for linear interpolation over quadratic simplices were derived by Rochery [17], working exclusively on the reference simplex and considering the error on the composition with the nonlinear element transformation. The resulting error estimate features an additional term involving the Hessian of the element transformation, that is, the curvature of the simplex, which vanishes for linear elements. Similar estimates were also obtained by Zhang along quadratic curves [22].

2. **Optimal size fields:** The optimal Riemannian metric minimizes the L^p norm of the pointwise error field. For linear interpolation, the cost functional is simple and the minimization problem was solved analytically [8, 9], yielding a closed-form solution for the optimal Riemannian metric under a constraint on the number of mesh vertices. For higher-order interpolation, the problem can also be solved analytically and takes the same form as for the linear case, with different parameters.

   Alternatively, the minimization problem can be tackled numerically. This is done in the Mesh Optimization via Error Sampling and Synthesis (MOESS) approach from Yano and Darmofal [31], where a Riemannian metric is built iteratively to minimize any given error functional. This has been successfully applied to minimize various goal-oriented functionals [39, 40].

   In contrast, the metric minimizing the L^p norm of the pointwise error for quadratic simplices has not been derived, either analytically or numerically. This means that the closed-form of this metric, if it exists, is yet unknown. It requires minimizing a functional involving both the metric and its derivatives, with additional derivatives involved as the order of the polynomial representation of the simplices increases. Ad hoc Riemannian metrics were proposed in Zhang and our work [41], with mitigated success. These metrics are inspired by the *induced metric*, discussed in Chapter 3, and limit the interpolation error along the isolines and gradient lines of the field of interest.

3. **Mesh generation:** Unit linear simplices have edges of unit length when measured with respect to a Riemannian metric, and are thus equilateral for this metric. They are, as a result, anisotropic with respect to the Euclidean metric. Unit simplices are

the ideal elements for a given metric, and optimal triangulations should ideally consist only of such elements. It is generally impossible to generate meshes consisting only of truly unit simplices, hence **quasi-unit** simplices relax this condition. Their edges are of length within $[1/\sqrt{2}, \sqrt{2}]$, and are achievable for practical mesh generators. Quasi-unit meshes are generated by splitting edges and collapsing vertices until all edges have the required length with respect to the metric, and various topological operations are performed to enhance the mesh quality.

There is currently no off-the-shelf metric-based mesh generator that creates curved meshes, thus generation must be implemented within an in-house code. As linear meshing softwares rely on an efficient generation of **unit** and **quasi-unit** simplices, these notions must be extended to curvilinear elements. Rochery proposed a definition of high-order unit simplices based on their Jacobian matrix, which is usually taken as an invariant for linear simplices. Such unit elements have unit edges, and thus encompass the linear case. They recently proposed a one-step curving method to curve edges, so as to be unit according to this definition, giving a practical way of generating unit quadratic simplices [33]. Alternatively, Aparicio-Estrems [25,26] proposed to curve mesh simplices by minimizing a distortion measure with respect to the metric. This approach is closely related to Rochery's definition of unit simplices, as discussed in Chapter 4 and in Rochery's thesis.

Contributions of this thesis

The main contribution concerns curvilinear mesh adaptation with triangles in two dimensions:

- **Error indicators:** We expand Rochery's results and propose an error estimate for interpolation of arbitrary order on curved triangles. This estimate is a straightforward application of the high-order estimate described in Coulaud et al. [11, 12] to the high-order derivatives on the reference simplex of the field of interest.

- **Optimal size fields:** We discuss an improved version of the ad hoc metric proposed by Zhang [22], limiting the interpolation error on edges. Then, from the error estimate above and the log-simplex method of Coulaud et al., we propose an iterative method to derive the optimal metric for high-order interpolation on curved triangles. The presence of metric derivatives changes the nature of the problem from a nonlinear equation to a nonlinear system of PDE in the metric components. This system was not solved in this work, rather, we propose an ad hoc extension of the iterative scheme in [11, 12] to obtain a solution on a simple application.

- **Mesh generation:** We revisit unit elements in light of notions of Riemannian geometry. Observing that linear unit elements preserve the edge length of an equilateral triangle, and that Rochery's extension to curvilinear triangles involves the pointwise Jacobian matrix of the transformation, we define unit elements as transformations of a reference simplex by **Riemannian isometries**. These isometries preserve the lengths, angles and areas, but also the Riemannian metric, imposing the same condition on the Jacobian matrix as the one proposed by Rochery. This is the main contribution of this thesis. Such a definition of unitness in terms of Riemannian isometries has several advantages. The first is that, conceptually, it describes unit elements as what they are and not through their properties, either in terms of edge lengths or Jacobian matrix. Then, as isometries, unitness preserves the geodesic property of the edges of the reference simplex. In other words, the edges of isometric unit elements are also geodesics in their metric space, that is, curves of minimal length. This justifies the intuitive construction of curved edges by length minimization used in [17, 21, 22, 41]. Lastly, this offers a broader framework as unit triangulations can be based on other reference simplices, not only regular (equilateral) simplices. For instance, triangulations isometric to a regular tiling of the space with right triangles are constructed. Unit isometric triangles are generated by explicitly controlling the inner angles, in addition to the edge lengths, then curved to minimize either the metric distortion [25, 26] or the Riemannian length.

Aside from these recent results on curvilinear mesh adaptation, we present a small contribution to the computation and storage of a size field induced by the CAD model for 3D mesh generation. This work was done at the beginning of this thesis, and concerns isotropic and linear tetrahedral meshes, whose size is governed locally by the curvature and the thinness of the CAD model. It was published in the *International Journal for Numerical Methods in Engineering* in 2021.

Plan of the thesis

- High-order Lagrange simplices are introduced in Chapter 2. We define the notations adopted in this document, as well as interpolation and integration on Lagrange $\mathcal{P}^q$ triangles. The notion of validity is introduced, with practical criteria on the Bézier control coefficients [15, 17]. As a motivation to generate curvilinear meshes, we present the minimization problem considered by Rochery: starting from a linear mesh, the $\mathcal{P}^2$ edge vertices are moved to minimize the exact interpolation error in L^2 norm. The resulting meshes show a considerable reduction in both the L^2 norm and H^1 seminorm

of the error, going up to almost a full order of convergence. With the exception of this motivating example, the contents of this chapter are standard and the reader familiar with high-order meshes can skip this chapter without loss of continuity.

- Riemannian metrics are used throughout this thesis, and are formally introduced in Chapter 3. The definition of unit elements in Chapter 4 relies on Riemannian isometries, which *pull* the metric on a *smooth manifold* back to the metric of another smooth manifold. Such manifolds are said to be *isometric*, and thus have the same *curvature*. These notions are better appreciated after a proper introduction to the geometry of manifolds. We made the choice of presenting these textbook notions in a self-contained form, resulting in a theoretical and slightly tedious chapter, but which helps grasping some key aspects of Riemannian geometry. The reader who is already familiar with these notions and their application to mesh generation can skip this chapter without loss of continuity.

- In Chapter 4, the existing definitions of unit simplices are compared, then we introduce a new one for curved simplices in terms of Riemannian isometries. Simplices are said to be unit if they are an isometric transformation of an ideal simplex, which may be arbitrary and does not have to be equilateral. From the properties of isometries, the existing definitions of unitness are recovered, as well as the invariants shared by unit simplices. Criteria for the existence of high-order unit triangles are discussed: they exist on flat manifolds, that is, manifolds isometric to the Euclidean space $\mathbb{R}^n$. Most manifolds are not flat, so we discuss a generalization of quasi-unitness for isometric simplices. Error estimates are then derived on isometric simplices, following the linear error estimate derived by Rochery [17].

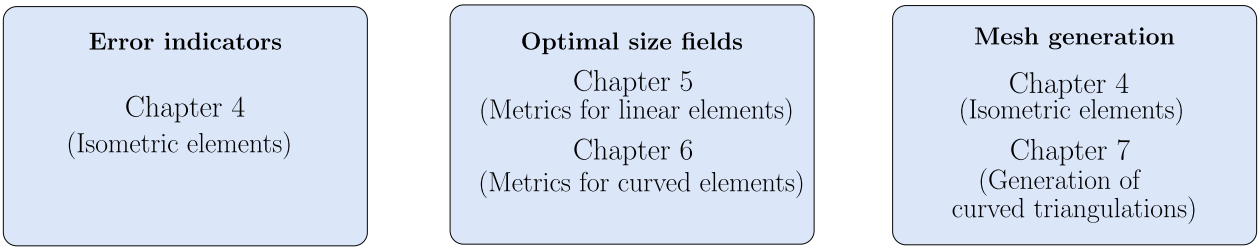
- We review in Chapter 5 existing optimal Riemannian metrics for anisotropic adaptation on linear meshes. These are the state-of-the-art metrics to create anisotropic meshes to control the interpolation error. The closed-form solution of Mirebeau [38] is recalled, as well as the log-simplex method for high-order interpolation [11, 12], for which an interpretation as a fixed-point method between manifolds is given. This interpretation is used in Chapter 6 to generalize the problem of the optimal metric to curved elements. Illustrative examples show the application of optimal metrics to perform mesh adaptation with linear elements, with respect to both analytical fields and CFD benchmarks.

- Chapter 6 tackles the problem of finding optimal metrics to minimize the interpolation error on curved meshes. The metric proposed in Zhang [22] is first discussed, and its

limitations are identified. Then, we apply the log-simplex method to the error estimate of Chapter 4 to derive the equations satisfied by the optimal metric. The exact system of PDE is not solved, but we propose a simple ad hoc iterative scheme based on the log-simplex method.

- Quasi-isometric triangulations are generated in Chapter 7. A frontal algorithm for the generation of quadratic triangulations is presented, following [21, 22, 42, 43]. In addition to the edge lengths, the inner angles are controlled with respect to the input metric, according to the selected ideal simplex. Results are presented by setting the ideal simplex as equilateral, thus the meshes have unit edges and inner angles centered around 60° with respect to the metric. Curvature is driven by minimizing either the Riemannian length or the distortion with respect to the metric and the ideal element.
- Lastly, early work of this thesis, unrelated to curvilinear mesh adaptation, is presented in Chapter 8. Isotropic size fields are computed for 3D CAD models based on their surface curvature and their medial axis. The approach is illustrated on geometric models from various datasets.

The role of each chapter with respect to the research questions is summarized below. Chapter 4 should be read before Chapters 5, 6 and 7. Chapters 4 and 7 can be read as a whole, where a definition of unitness and its practical implementation are presented. Chapter 8 is disconnected from the rest of the thesis and can be read separately.



Publication and conference proceedings

- "Automatic feature-preserving size field for three-dimensional mesh generation", A. Bawin, F. Henrotte, J.-F. Remacle, *Int. Journal for Numerical Methods in Eng.*, 2021.
- "Optimally Convergent Isoparametric $\mathcal{P}^2$ Mesh Generation", A. Bawin, A. Garon, J.-F. Remacle, SIAM IMR proceedings, 2023.
- "Isometric simplices for high-order mesh adaptation", A. Bawin, A. Garon, J.-F. Remacle, *SIAM Journal of Numerical Analysis*, in preparation.

Participation to conferences

- International Meshing Roundtable 2019, Buffalo. Presentation of a poster: "Mesh Size Field with Minimal User Input".
- Toronto Fields Institute – UOttawa Workshop, 2021, Virtual: "Generation of Quasi-Unit Curvilinear Meshes for 2D Numerical Simulations".
- WCCM-APCOM 2022, Virtual: "Curvilinear Grids Adapted from 2D Numerical Simulations.
- SIAM IMR 2022, Virtual: "Curving Strategy for $\mathcal{P}^2$ Mesh Generation".
- SIAM IMR 2023, Amsterdam: "Optimally Convergent Isoparametric $\mathcal{P}^2$ Mesh Generation".
- WCCM 2023, Cannes: "Metric-based Curvilinear Mesh Adaptation on Lagrangian Coherent Structures".
- Poster presentations during the GPS (Génie Par la Simulation) workshops at Polytechnique Montréal.

Code

- Development of an open-source, generic purpose, 0-, 1- and 2D unsteady finite element (continuous Galerkin) `C++` library, with the following features:
  - Natively high-order code, dealing with $\mathcal{P}^1$ or $\mathcal{P}^2$ triangulations.
  - Toolbox of differential operators (finite element weak forms) on scalar and vector fields inspired by the MFEM project [44], with interface with Intel MKL Pardiso and PETSc for linear solvers and exportation of simulation results for visualization with ParaView.
  - Recovery of solution and derivatives of arbitrary order with Zhang and Naga's polynomial preserving recovery (see Appendix B.1).
  - Computation of metric tensor fields for anisotropic adaptation, metric and derivatives interpolation, and interface with `mmg2d` for anisotropic remeshing.
- Collection of routines in Gmsh for the generation of curved triangulations: points generation, triangulation, edge curving, elements validity routines, topological operations.
- Collection of routines in Gmsh for automatic isotropic mesh size prescription for 2D and 3D models based on P4EST [45].

CHAPTER 2 HIGH-ORDER SIMPLICES

High-order simplices denote curves, triangles and tetrahedra represented by polynomials of degree 2 and higher. These polynomials are usually written using either the Lagrange or Bézier basis functions. In the Lagrange formulation, the control points coincide with the mesh vertices making the Lagrange description of a high-order simplex quite intuitive. With the exception of corner vertices, the Bézier control points are generally not constrained to lie on the boundary of the simplex. In this thesis, following the work of Zhang [23], simplices are represented with Lagrange polynomials, and the emphasis is put in this chapter on the Lagrange formulation. High-order simplices may curve to reduce the approximation error on the geometry or on the resolved fields. Historically, curved simplices were almost exclusively located near the boundaries to improve the representation of the CAD model. Inner simplices would only curve to avoid the inversion of boundary simplices. Recent works [17,21–23,27,41] explored applying the added flexibility of curved simplices to control e.g. the interpolation error inside the computational domain.

We introduce in this chapter well-known concepts and notations used throughout the thesis, including the transformation from reference to high-order simplices (Sections 2.1 through 2.3), Lagrange interpolation (Section 2.4), and integration (Section 2.5). These last operations constrain the simplices to be positively oriented everywhere, yielding the notion of validity, discussed in Section 2.6. Interpolation on curved meshes computed through the reference simplex is generally discouraged, as arbitrary edge curvature can degrade the optimal convergence order. In Section 2.7, we show as in Rochery [17] that triangulations curved to minimize the interpolation error exhibit reduced error in L^2 norm and H^1 seminorm.

This chapter is inspired by the excellent presentation of Rochery [17] on Bézier simplices. Similar notations are used for easier identification, and we refer to their work for the proof of some results.

2.1 Reference element

In standard finite element methods, the mesh is defined in the so-called **physical space** or world space Ω, which is a compact subset of $\mathbb{R}^n$ where the CAD model is defined. In this work, we use the notation $\boldsymbol{x} = (x^1, \dots, x^n)$ to denote coordinates for the physical space, which we call **physical coordinates**. In contrast, practical computations are performed on a

single reference element, after a change of variable, in the **reference space** $\widehat{\Omega}$. The reference space is also a compact subset of $\mathbb{R}^n$, with **reference coordinates** $\boldsymbol{\xi} = (\xi^1, \dots, \xi^n)$. In this thesis, we only consider meshes of triangles, but we present here definitions for simplices of arbitrary dimension n. When working with simplices, another useful set of numbers are the **barycentric coordinates** $\boldsymbol{u} = (u^1, \dots, u^{n+1})$, which are the ratios of the volume of each subsimplex formed by a point $\boldsymbol{p}$ and the n opposing points of the simplex to the volume of the whole simplex. They are not coordinates in the sense of *coordinates maps* introduced in Chapter 3 when looking at manifolds, because they are $n+1$-tuples, but they are constrained by the closure relation $\sum_i u^i = 1$ and thus only have n independent coordinates. The correspondence between reference and barycentric coordinates is:

$$\boldsymbol{\xi} \mapsto \boldsymbol{u}(\boldsymbol{\xi}) = \left(1 - \sum_{i=1}^{n} \xi^i, \xi^1, \dots, \xi^n\right), \qquad \boldsymbol{u} \mapsto \boldsymbol{\xi}(\boldsymbol{u}) = (u^2, \dots, u^{n+1}), \tag{2.1}$$

where the position of the closure relation is arbitrary and was set here as the first barycentric coordinate to match the linear Lagrange basis functions introduced hereafter. For instance, we have in two dimensions:

$$\begin{aligned} (\xi^1, \xi^2) \triangleq (\xi, \eta) &\mapsto (u, v, w) = (1 - \xi - \eta, \xi, \eta), \\ (u^1, u^2, u^3) \triangleq (u, v, w) &\mapsto (\xi, \eta) = (v, w). \end{aligned} \tag{2.2}$$

The **reference n-simplex** $\widehat{K}$, also called the **reference element** when the dimension is understood, is a convex combination of $n+1$ points of $\widehat{\Omega}$ and is defined by either of the following sets:

$$\widehat{K} \triangleq \left\{ \boldsymbol{\xi} \in [0,1]^n \;\middle|\; \sum_{i=1}^{n} \xi_i \leq 1 \right\} = \left\{ \boldsymbol{u} \in [0,1]^{n+1} \;\middle|\; \sum_{i=1}^{n+1} u_i = 1 \right\}. \tag{2.3}$$

In two dimensions, it is the triangle of vertices $(0,0)$, $(1,0)$ and $(0,1)$. On the reference element, the Lagrange polynomials of degree k, introduced hereafter, are defined from equally spaced nodes along each coordinate axis, see Fig. 2.1. These nodes have reference coordinates $\boldsymbol{\xi} = (\frac{i_1}{k}, \dots, \frac{i_n}{k})$ with $i_m \in [\![0, k]\!]$, the set of positive integers between 0 and k. Their barycentric coordinates are obtained from (2.1) and can be written in a compact way using *multi-indices*, as proposed by Rochery [17]. In dimension n, the **set of multi-indices of degree** k is defined by:

$$\widehat{K}_n^k \triangleq \left\{ \alpha \in [\![0, k]\!]^{n+1} \;\middle|\; |\alpha| \triangleq \sum_{i=1}^{n+1} \alpha_i = k \right\}. \tag{2.4}$$

Multi-indices are $n+1$-tuples of positive integer numbers, which are frequently used to write compact expressions for e.g. Newton's binomial or higher-order Taylor's expansions for several variables. For example, the elements of $\widehat{K}_2^1$ are the triplets $(1,0,0),(0,1,0)$ and $(0,0,1)$. This way, the barycentric coordinates of the Lagrange nodes are simply $\boldsymbol{u}_\alpha = (\alpha/k)$ for $\alpha \in \widehat{K}_d^k$. Similarly, $\boldsymbol{\xi}_\alpha = (\alpha_2, \ldots, \alpha_{n+1})/k$. As mentioned in [17], the notation $\widehat{K}_n^k$ comes from the fact that the set of multi-indices can be identified with the intersection of the grid $\mathbb{N}^n$ of naturals with the scaled reference element $n \times \widehat{K}$. This yields a compact identification of the vertices of the reference element of arbitrary order, as shown in Fig. 2.1. The cardinality $|\widehat{K}_n^k|$ of this set, also noted $N(k)$, is [17]:

$$|\widehat{K}_n^k| = \frac{(k+n)!}{k!n!}, \tag{2.5}$$

thus:

$$|\widehat{K}_1^k| = k+1, \quad |\widehat{K}_2^k| = \frac{(k+1)(k+2)}{2}, \quad |\widehat{K}_3^k| = \frac{(k+1)(k+2)(k+3)}{6}. \tag{2.6}$$

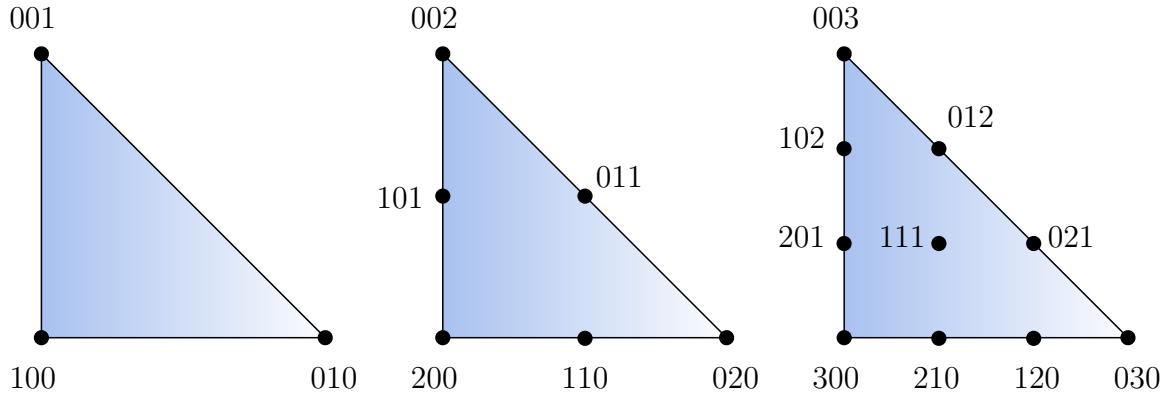


Figure 2.1 Reference nodes of triangles of order 1, 2 and 3.

2.2 Lagrange basis functions on the reference element

The Lagrange basis functions are a well-known polynomial basis defined on the reference element. They are unit at their associated node and zero at all other nodes, and are thus particularly easy to come by when interpolating fields. First, we define the space of homogeneous polynomials of degree k on $\widehat{K}$ in barycentric coordinates by:

$$\mathbb{H}^k(\widehat{K}) = \left\{ \text{span} \prod_{i=1}^{n+1} (u^i)^{m_i} \;\middle|\; \boldsymbol{u} \in \widehat{K}, \quad m \in \widehat{K}_n^k, \quad |m| = k \right\}. \tag{2.7}$$

These are the homogeneous polynomials of the form $u^k, u^{k-2}vw, vw^{k-1}$, etc. Then, the **Lagrange basis functions of degree k in barycentric coordinates** are the set of polynomials $\phi^b_\alpha \in \mathbb{H}^k(\widehat{K})$ such that for all $\alpha, \beta \in \widehat{K}^k_n$:

$$\phi^b_\alpha(\boldsymbol{u}_\beta) = \phi^b_\alpha\left(\frac{\beta}{k}\right) = \delta_{\alpha\beta}, \tag{2.8}$$

where the superscript b stands for *barycentric*. In other words, these are the homogeneous polynomials of degree k which are 1 at the barycentric coordinate β/k and 0 everywhere else, where β is a multi-index. The linear Lagrange basis functions in two dimensions are simply the barycentric coordinates, since they are 1 at their vertex and vanish at the two others:

$$\phi^b_{100} = u, \quad \phi^b_{010} = v, \quad \phi^b_{001} = w \tag{2.9}$$

The quadratic functions are given by:

$$\begin{aligned} \phi^b_{200} = u(u-v-w), \quad \phi^b_{110} = 4uv, \quad \phi^b_{020} = v(v-u-w), \\ \phi^b_{011} = 4vw, \quad \phi^b_{002} = w(w-u-v), \quad \phi^b_{101} = 4uw. \end{aligned} \tag{2.10}$$

These polynomials are also commonly written in terms of the reference coordinates. The **Lagrange basis functions of degree k in reference coordinates** are obtained by evaluating the ϕ^b_α basis at $\boldsymbol{u}(\boldsymbol{\xi})$. They are given explicitly by [17]:

$$\phi_\alpha(\boldsymbol{\xi}) \triangleq \phi^b_\alpha(\boldsymbol{u}(\boldsymbol{\xi})) = \frac{k^k}{\prod_{i=1}^{n+1}\alpha_i!}\prod_{i=1}^{n+1}\prod_{j=0}^{\alpha_i-1}\left(u^i(\boldsymbol{\xi}) - \frac{j}{k}\right), \tag{2.11}$$

where the first term is a normalization coefficient obtained by evaluating the double product at the grid node $\boldsymbol{u}_\alpha$. Graphically, (2.11) associates to a grid node the normalized product of the k straight lines passing through the remaining $|\widehat{K}^k_d| - 1$ grid nodes, which is sometimes the way these polynomials are introduced. For quadratic polynomials in two dimensions, either computing (2.11) or substituting $\boldsymbol{u}(\boldsymbol{\xi})$ into (2.10) yields:

$$\begin{aligned} \phi_{200} &= 2\,(1-\xi-\eta-0)\left(1-\xi-\eta-\frac{1}{2}\right) &&= (1-\xi-\eta)(1-2\xi-2\eta), \\ \phi_{110} &= 4\,(1-\xi-\eta-0)\,(\xi-0) &&= 4\xi(1-\xi-\eta), \\ \phi_{020} &= 2\,(\xi-0)\left(\xi-\frac{1}{2}\right) &&= \xi(2\xi-1), \\ \phi_{011} &= 4\,(\xi-0)\,(\eta-0) &&= 4\xi\eta, \\ \phi_{002} &= 2\,(\eta-0)\left(\eta-\frac{1}{2}\right) &&= \eta(2\eta-1), \end{aligned}$$

$$\phi_{101} = 4\,(1-\xi-\eta-0)\left(\eta-\frac{1}{2}\right) \qquad\qquad = 4\eta(1-\xi-\eta).$$

The Lagrange basis functions form a partition of unity, that is, $\sum_\alpha \phi_\alpha = 1$, thus for all i, $\sum_\alpha \partial_{u^i}\phi^b_\alpha(\boldsymbol{u}) = 0$, and similarly $\sum_\alpha \partial_{\xi^i}\phi_\alpha(\boldsymbol{\xi}) = 0$. The Lagrange basis functions can be positive or negative on $\widehat{K}$, unlike the Bernstein polynomials which are always nonnegative.

2.3 Physical and regular element

Mesh elements in the physical space are described by a set of control nodes and basis functions. In this thesis, mesh elements are represented by Lagrange functions, so that the associated control nodes are the **mesh vertices** $\boldsymbol{x}_i$, which all lie inside or on the boundary of the elements. Other choices are possible, such as the Bernstein polynomials used to describe mesh elements as Bézier curves. In that case, some control nodes lie outside of the elements.

Definition 2.1. *A d-dimensional **Lagrange simplex of degree** q, or simply a $\mathcal{P}^q$**-simplex**, is a curved simplex $K \triangleq F_K(\widehat{K})$ in the physical space $\Omega \subset \mathbb{R}^n$, where the map F, called the **reference-to-physical transformation**, is defined for a set of $\widehat{K}^q_d$ mesh vertices $\boldsymbol{x}_\alpha$ by:*

$$F_K \triangleq F_{\widehat{K}\to K} : \widehat{K} \to K : \boldsymbol{\xi} \mapsto F_K(\boldsymbol{\xi}) = \sum_{\alpha\in\widehat{K}^q_d} \phi_\alpha(\boldsymbol{\xi})\boldsymbol{x}_\alpha. \tag{2.12}$$

This definition covers the case of simplices whose intrinsic dimension d is lower than the dimension n of the physical space, for instance curves ($d = 1$) and triangles ($d = 2$) in $\mathbb{R}^3$. In this work, we mostly focus on triangles in $\mathbb{R}^2$, for which $n = d$. Such a triangle K is referred to as a **physical element** or a **physical triangle**. K is also called a **linear triangle** or a **straight triangle** if $q = 1$, a **quadratic triangle** if $q = 2$, and so on. If $q > 1$, we also use the generic appellation **high-order triangle** (of **degree** q). The transformation for a $\mathcal{P}^2$ triangle, on which we focus in this work, is illustrated in Fig. 2.2, along with the numbering convention introduced further.

For linear triangles, all vertices $\boldsymbol{x}_\alpha$ are regular mesh vertices, which we also call **corner vertices**. For quadratic triangles, we call **edge vertices** the three vertices $\boldsymbol{x}_{110}, \boldsymbol{x}_{011}$ and $\boldsymbol{x}_{101}$, and additionally, for cubic triangles, the vertex $\boldsymbol{x}_{111}$ is called an **element vertex**, with the straightforward generalization to edge and element vertices for higher-order triangles. Linear triangles are, obviously, always straight, whereas high-order triangles are a special case of **curved triangles**, i.e., triangles made of three smooth (C^∞) curve segments, for which the curves are polynomial. Depending on the localization of their vertices $\boldsymbol{x}_\alpha$ in the

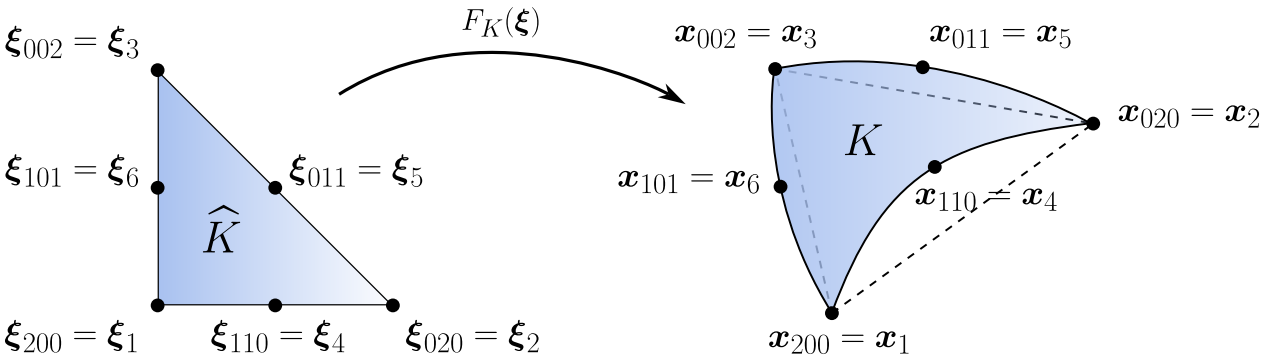


Figure 2.2 Reference-to-physical transformation for a $\mathcal{P}^2$ triangle and numbering convention.

physical space, high-order triangles can either be straight and linear, if their vertices are uniformly placed as in Fig. 2.1, or straight and nonlinear, if their edge vertices are aligned but not uniformly spaced, or curved and nonlinear. Interpolation in the physical space, introduced hereafter, requires the transformation F_K to be invertible. In general, however, there is no expression for the inverse F_K^{-1}, except for linear triangles. This is not a problem if the application of F_K^{-1} can be bypassed by manipulating the reference coordinates directly (e.g. when computing quadratures, see Section 2.5), but it matters for the localization of a physical point in a background high-order mesh. In that case, the reference coordinates of the queried point are obtained by solving numerically the nonlinear system $\boldsymbol{\xi} = F_K^{-1}(\boldsymbol{x})$, for instance with a Newton-Raphson method.

The **Jacobian matrix of the reference-to-physical transformation** is the $n \times d$ matrix J_K defined by:

$$J_K = J_K(\boldsymbol{\xi}) \triangleq \frac{\partial F_K}{\partial \boldsymbol{\xi}} \quad \longrightarrow \quad J_{K,ij} = \frac{\partial F_K^i}{\partial \xi^j}. \tag{2.13}$$

For a physical element K, $n = d$ and J_K is a square matrix. As mentioned previously, interpolation on K requires F_K to be invertible. By the inverse function theorem, this is guaranteed when J_K is invertible on K, thus when its determinant is nonzero. This is discussed in more details in Section 2.6.

The continuous mesh framework, introduced in Chapter 4, aims at generating elements with unit edges, which are essentially the images of a regular (equilateral) simplex. To this end, it is useful to define a **reference regular element** or **ideal element** $K_\triangle$ in $\mathbb{R}^n$, with coordinates $\boldsymbol{y} = (y^1, \ldots, y^n)$. The coordinates $\boldsymbol{\xi}, \boldsymbol{x}$ and $\boldsymbol{y}$ are fundamentally one and the same, but they are kept distinct to emphasize which elements we examine. In two dimensions, $K_\triangle$ is defined

as the equilateral linear triangle with vertices $\boldsymbol{y}_1 = (0,0), \boldsymbol{y}_2 = (1,0), \boldsymbol{y}_1 = (1/2, \sqrt{3}/2)$. Its edges have unit length, and its area is $|K_\triangle| = \sqrt{3}/4$. The transformations from the reference element to $K_\triangle$ and from $K_\triangle$ to a physical element K are noted $F_\triangle \triangleq F_{\widehat{K} \to K_\triangle}$ and $F_{\triangle \to K} \triangleq F_{K_\triangle \to K}$, respectively, with Jacobian matrices:

$$J_\triangle(\boldsymbol{\xi}) \triangleq \frac{\partial F_\triangle}{\partial \boldsymbol{\xi}} = \frac{\partial F_\triangle^i}{\partial \xi^j} = \begin{pmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{pmatrix}, \quad \text{and} \quad J_{\triangle \to K}(\boldsymbol{y}) \triangleq \frac{\partial F_{\triangle \to K}}{\partial \boldsymbol{y}} = \frac{\partial F_{\triangle \to K}^i}{\partial y^j}, \tag{2.14}$$

and satisfy, as illustrated in Fig. 2.3:

$$F_{\triangle \to K} = F_K \circ F_\triangle^{-1} \quad \text{and} \quad J_{\triangle \to K} = J_K J_\triangle^{-1}. \tag{2.15}$$

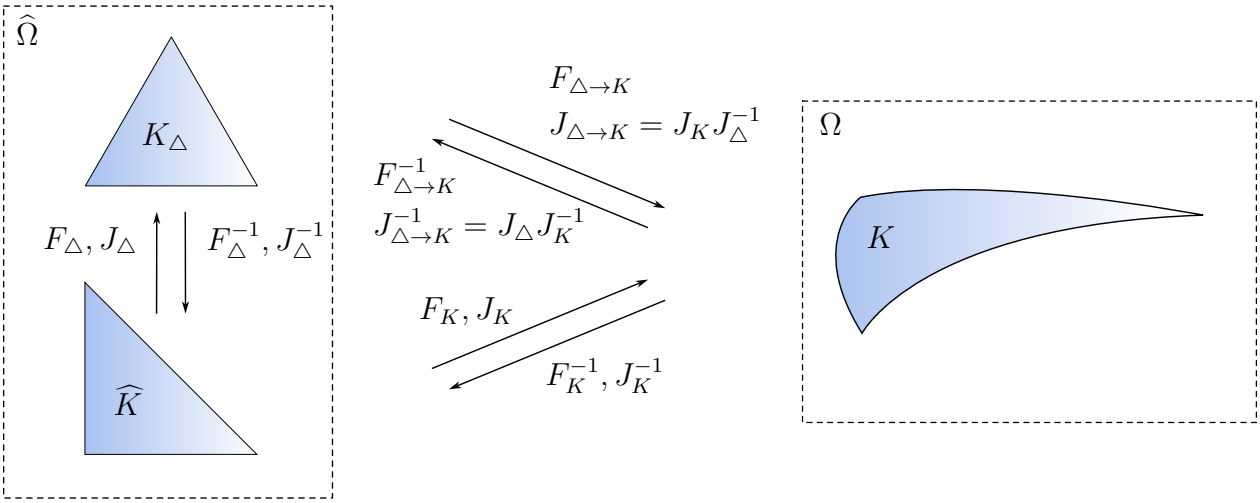


Figure 2.3 Reference elements $\widehat{K}$ and $K_\triangle$, physical element K and their transformations.

The boundary of a d-dimensional $\mathcal{P}^k$ simplex is the union of $d+1$ $\mathcal{P}^k$ simplices of dimension $d-1$: these boundary simplices are the **faces** in three dimensions and the **edges** in two dimensions. For instance, the boundary of a quadratic triangle is the union of three $\mathcal{P}^2$ edges noted ∂K_i. The mesh vertices of the i-th of these edges are the $\boldsymbol{x}_\alpha$ with the multi-index α such that $\alpha_i = 0$ [17].

Lastly, we define two types of **orientations** of a physical triangle as follows. The reference triangle $\widehat{K}$ is a subset of $\mathbb{R}^2$ with standard basis vectors $(\boldsymbol{e}_1, \boldsymbol{e}_2)$. This basis is said to be **positively oriented** because $\boldsymbol{e}_2$ is located counterclockwise to $\boldsymbol{e}_1$. Since the vector fields $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are defined at all points of $\widehat{K}$, we can say that $\widehat{K}$ is *locally positively oriented* everywhere. Similarly, we say that a physical triangle K is **locally positively oriented** at

$\boldsymbol{x} = F_K(\boldsymbol{\xi})$ if the image[1] by F_K of $\boldsymbol{e}_2(\boldsymbol{\xi})$ is counterclockwise to the image of $\boldsymbol{e}_1(\boldsymbol{\xi})$, as shown in Fig 2.4. When it is the case, we say that F_K is an **orientation-preserving** transformation, and one shows that the Jacobian matrix of F_K at $\boldsymbol{\xi}$ has strictly positive determinant ([46], Ex. 15.13). Alternatively, orientation in the meshing literature is often defined by the numbering of the vertices, edges and faces of an element. For instance, one can declare a triangle to be positively oriented if its vertices are numbered in counterclockwise order, as is done in Gmsh. With the multi-index convention used so far, a numbering is not relevant since there is no order relation between two multi-indices, however, for implementation purposes, it is useful to assign a sequential number to each vertex and edge of an element to store them in a table. Using the numbering convention hereafter, we say that a triangle is **globally positively oriented** if its vertices (and thus its edges) are numbered in counterclockwise order. For high-order triangles, the adopted convention numbers the corner vertices first, then the edge vertices, followed by the element vertices, thus several numbering loops are performed, all in counterclockwise order.

Linear triangles that are globally positively oriented are also locally positively oriented everywhere. This is not true for high-order triangles, which can be positively oriented globally but not locally, that is, which can be inverted despite being numbered counterclockwise. Such elements are invalid, as discussed in Section 2.6 further.

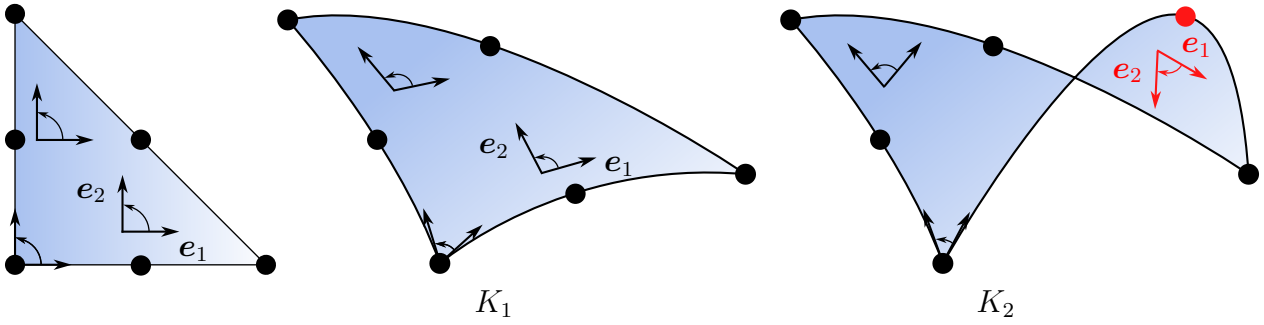


Figure 2.4 $\widehat{K}$ and K_1 are locally positively oriented everywhere, but K_2 is not.

Convention for the vertices and edges of $\mathcal{P}^2$ triangles

Since we work mostly with quadratic triangles, we define shorter notations to refer to their vertices and edges. Instead of using multi-indices, the vertices are numbered from 1 to

[1]The formal definition of the image of a vector by a map F is presented in Chapter 3 as the image by the *differential* of F, Section 3.1.1. We refer to e.g. Chapter 15 of Lee [46] for a formal presentation of orientation on manifolds.

6, starting with the three corner vertices, then the edge vertices. The edge vertices are alternatively noted with two indices, relative to their edge's endpoints, so that:

$$\begin{gathered}\boldsymbol{x}_1 = \boldsymbol{x}_{200}, \quad \boldsymbol{x}_2 = \boldsymbol{x}_{020}, \quad \boldsymbol{x}_3 = \boldsymbol{x}_{002}, \\ \boldsymbol{x}_4 = \boldsymbol{x}_{12} = \boldsymbol{x}_{110}, \quad \boldsymbol{x}_5 = \boldsymbol{x}_{23} = \boldsymbol{x}_{011}, \quad \boldsymbol{x}_6 = \boldsymbol{x}_{31} = \boldsymbol{x}_{101}.\end{gathered} \tag{2.16}$$

These notations are used interchangeably, and we use whichever is most convenient depending on the context. The same notation is adopted for the associated basis functions. The three edges are noted either e_i or e_{ij}, are oriented, and originate from $\boldsymbol{x}_i$, so that:

$$e_1 = e_{12} = \partial K_3, \quad e_2 = e_{23} = \partial K_1, \quad e_3 = e_{31} = \partial K_2. \tag{2.17}$$

For instance, for reference coordinates $\boldsymbol{\xi}$ such that $u^3(\boldsymbol{\xi}) = \eta = 0$, we have:

$$\begin{aligned}e_1(\boldsymbol{\xi}) = \sum_{\alpha \in \widehat{K}_1^2,\ \alpha_3 = 0} \phi_\alpha(\boldsymbol{\xi}) \boldsymbol{x}_\alpha &= \phi_{200}(\boldsymbol{\xi})\boldsymbol{x}_{200} + \phi_{110}(\boldsymbol{\xi})\boldsymbol{x}_{110} + \phi_{020}(\boldsymbol{\xi})\boldsymbol{x}_{020} \\ &= \phi_1(\boldsymbol{\xi})\boldsymbol{x}_1 + \phi_4(\boldsymbol{\xi})\boldsymbol{x}_4 + \phi_2(\boldsymbol{\xi})\boldsymbol{x}_2.\end{aligned} \tag{2.18}$$

The basis functions above for $\eta = 0$ reduce to the 1-dimensional basis functions for $\xi \in [0, 1]$:

$$\phi_1(\xi) = (1 - \xi)(1 - 2\xi), \quad \phi_4(\xi) = 4\xi(1 - \xi), \quad \phi_2(\xi) = \xi(2\xi - 1), \tag{2.19}$$

and similarly for the other two edges, so that each edge can be parameterized by $t \in [0, 1]$:

$$\begin{aligned}e_{ij}(t) &= (1 - t)(1 - 2t)\boldsymbol{x}_i + 4t(1 - t)\boldsymbol{x}_{ij} + t(2t - 1)\boldsymbol{x}_j, \\ e'_{ij}(t) &= (4t - 3)\boldsymbol{x}_i + 4(1 - 2t)\boldsymbol{x}_{ij} + (4t - 1)\boldsymbol{x}_j.\end{aligned} \tag{2.20}$$

Finally, we also note $\overline{\boldsymbol{x}}_{ij} = (\boldsymbol{x}_i + \boldsymbol{x}_j)/2$ the middle of the straight segment $\boldsymbol{x}_i\boldsymbol{x}_j$, and define the **displacement** $\boldsymbol{a}_{ij} \triangleq \boldsymbol{x}_{ij} - \overline{\boldsymbol{x}}_{ij}$ from this midpoint to the edge vertex.

2.4 Interpolation and interpolation error

Next, we introduce interpolation with Lagrange basis functions on elements of arbitrary order. Interpolation can be performed by a change of variable, using the reference element, or directly on the physical element: these approaches are known in Botti [47] as *reference space interpolation* and *physical space interpolation*, respectively.

Reference space interpolation

For $\hat{f} : \widehat{K} \to \mathbb{R}$, the **Lagrange interpolant of degree k of $\hat{f}$ on the reference element** is the polynomial $\Pi^k_{\widehat{K}} \hat{f}(\boldsymbol{\xi})$ defined by the value of $\hat{f}$ at the uniformly spaced **interpolation nodes $\boldsymbol{\xi}_\alpha$**:

$$\Pi^k_{\widehat{K}} \hat{f} : \widehat{K} \to \mathbb{R} : \boldsymbol{\xi} \mapsto \Pi^k_{\widehat{K}} \hat{f}(\boldsymbol{\xi}) = \sum_{\alpha \in \widehat{K}^k_n} \phi_\alpha(\boldsymbol{\xi}) \hat{f}(\boldsymbol{\xi}_\alpha). \tag{2.21}$$

Then, Lagrange interpolation is defined on the physical element K by composing the function $f : K \to \mathbb{R}$ with the reference-to-physical transformation F_K. If K is a $\mathcal{P}^q$-simplex of dimension n and if F_K is invertible for all $\boldsymbol{x} \in K$, the **reference space Lagrange interpolant of degree k of f on K** is defined by:

$$\begin{aligned} \Pi^k_K f : K \to \mathbb{R} : \boldsymbol{x} \mapsto \Pi^k_K f(\boldsymbol{x}) &= \Pi^k_{\widehat{K}} (f \circ F_K)(F_K^{-1}(\boldsymbol{x})) \\ &= \sum_{\alpha \in \widehat{K}^k_n} \phi_\alpha(F_K^{-1}(\boldsymbol{x}))(f \circ F_K)(\boldsymbol{\xi}_\alpha) \\ &= \sum_{\alpha \in \widehat{K}^k_n} \phi_\alpha(\boldsymbol{\xi}(\boldsymbol{x})) f(F_K(\boldsymbol{\xi}_\alpha)) \\ &= \sum_{\alpha \in \widehat{K}^k_n} \phi_\alpha(\boldsymbol{\xi}(\boldsymbol{x})) f(\boldsymbol{x}_\alpha). \end{aligned} \tag{2.22}$$

This is how real-valued functions are interpolated in this thesis, so we simply refer to $\Pi^k_K f$ as the **interpolant of f**. When it is clear from context that we refer to the interpolant on physical elements, we omit the K subscript and write $\Pi^k f$. The degree k of the interpolant $\Pi^k f$ need not be the same as the degree q of the geometric transformation F_K: when $k > q$, the interpolant (or the associated finite element) is usually called **subparametric**. When $k = q$, the interpolant is called **isoparametric**, and the case $k < q$ is called **superparametric**. Sub- and isoparametric elements are used for most high-order discretizations of PDE, with a linear mesh everywhere except close to curved boundaries. To distinguish between both polynomial degrees, we write $(\mathcal{P}^q, \Pi^k)$ to denote degree k interpolation on a $\mathcal{P}^q$-simplex. In this work, the term *node* always refers to an interpolation node $\boldsymbol{\xi}_\alpha$ on the reference element, whereas points $(\boldsymbol{x}_i)_{i=1}^{N_v}$ in the mesh are called *vertices*. Nodes do not always have a physical counterpart in the mesh: for instance, for cubic interpolation on quadratic triangles, i.e., a $(\mathcal{P}^2, \Pi^3)$ configuration, the geometry is interpolated using nodes indexed by $\alpha \in \widehat{K}^q_d = \widehat{K}^2_2$, while the field f is interpolated with nodes indexed by $\beta \in \widehat{K}^k_d = \widehat{K}^3_2$, and only the corner nodes of $\widehat{K}^3_2$ have an associated vertex in the mesh, Fig. 2.1.

From the last line of (2.22), the basis effectively used to interpolate on the physical space

is $\phi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}(\boldsymbol{x})) = (\phi_{\boldsymbol{\alpha}} \circ F_K^{-1})(\boldsymbol{x})$, which is well-defined as long as F_K is invertible. The inverse transformation F_K^{-1} is not polynomial unless F_K is linear, so the functions $\phi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}(\boldsymbol{x}))$ are arbitrary and *not* polynomial in the physical space. This is illustrated in Fig. 2.5 for linear interpolation on both linear and quadratic triangles. This is a major difference between interpolation on linear or on curved meshes. As a result, polynomials in the physical space are only preserved by interpolation if they are of maximum degree $\lfloor k/q \rfloor$ (see e.g. [17] for a proof). This means, for instance, that quadratic interpolation over a quadratic mesh can no longer represent exactly the function x^2. For such function, however, it is up to the error estimate to prescribe straight triangles yielding zero interpolation error, as discussed in Chapter 6.

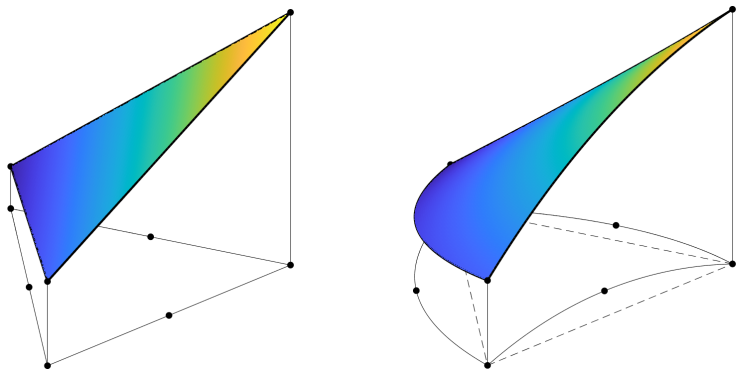


Figure 2.5 Linear interpolation on linear and quadratic triangles.

The **Lagrange interpolation error of degree** k **of** f **on** K is the real-valued function:

$$e_K^k f : K \to \mathbb{R} : \boldsymbol{x} \mapsto e_K^k(\boldsymbol{x}) = f(\boldsymbol{x}) - \Pi_K^k f(\boldsymbol{x}). \tag{2.23}$$

Following the discussion above, the interpolation error is identically zero only for polynomial functions f of degree up to $\lfloor k/q \rfloor$ in the physical space. In particular, Moxey et al. [48] predict a reduced convergence rate of $\lfloor k/q \rfloor + 1$ in L^∞ norm as soon as curved elements are used, and Botti [47] predicts a similar rate for L^2 projection using curved meshes. However, these results are on the conservative, and slightly pessimistic, side, and each of their numerical experiments involves curving the elements in a uniform and arbitrary way. Results of Sanjaya [28], Rochery [17] and of this thesis show, on the one hand, that direct minimization of the interpolation error on quadratic meshes yields an improved convergence rate compared to linear meshes, and, on the other hand, that mesh curving driven by Riemannian metrics obtained from all the relevant derivatives of f yields meshes recovering the $k+1$ convergence

rate for the interpolation error.

Physical space interpolation

Alternatively, Lagrange interpolation can be defined to be polynomial on the physical element. In this case, the basis varies from one element to another and is determined by solving a linear system on each element:

$$\phi^{\text{phys}}_{K,\alpha}(\boldsymbol{x}_{K,\beta}) = \delta_{\alpha\beta}. \tag{2.24}$$

The **physical space Lagrange basis functions** $\phi^{\text{phys}}_{K,\alpha}(\boldsymbol{x})$ span the space of continuous and piecewise-polynomial functions on Ω:

$$\mathbb{P}^k(\mathcal{T}) = \left\{ f \in L^2(\Omega) \mid f|_K \in \mathbb{P}^k(K), \quad \forall K \in \mathcal{T} \right\}, \tag{2.25}$$

where $\mathcal{T}$ is a conforming (i.e., without hanging vertices) triangulation of Ω and $\mathbb{P}^k(K)$ is the space of polynomials of degree k on K. Thus, the global interpolant on the physical space is polynomial, unlike interpolation with reference space basis functions. Similarly to (2.21), the **physical space Lagrange interpolant of degree k of f on K** is defined by:

$$\Pi^{k,\text{phys}}_K f : K \to \mathbb{R} : \boldsymbol{x} \mapsto \Pi^{k,\text{phys}}_K f(\boldsymbol{x}) = \sum_{\alpha \in \widehat{K}^k_n} \phi^{\text{phys}}_\alpha(\boldsymbol{x}) f(\boldsymbol{x}_\alpha). \tag{2.26}$$

Although the resolution of (2.24) yields a physical space basis, more numerically efficient methods are usually implemented. For instance, an orthonormal basis can be obtained from the monomial basis on each element through a Gram-Schmidt procedure, yielding unit diagonal mass matrices [47, 49, 50].

2.5 Integration

As for reference space interpolation, integrals on physical elements are computed on the reference element by a change of variable. The integral of a function $f : \widehat{K} \to \mathbb{R}$ is approximated by a quadrature with n_q points $\boldsymbol{\xi}_i$ and weights w_i:

$$\int_{\widehat{K}} f \, d\boldsymbol{\xi} \simeq \sum_{i=1}^{n_q} w_i f(\boldsymbol{\xi}_i), \qquad \int_{\widehat{K}} \Pi^k_{\widehat{K}} f \, d\boldsymbol{\xi} \simeq \sum_{i=1}^{n_q} w_i \left(\sum_{\alpha \in \widehat{K}^k_n} \phi_\alpha(\boldsymbol{\xi}_i) f(\boldsymbol{\xi}_\alpha) \right) \tag{2.27}$$

The change of variable formula yields for the integral over K:

$$\begin{aligned}
\int_K f \, d\boldsymbol{x} &= \int_{\widehat{K}} (f \circ F_K) \det J_K \, d\boldsymbol{\xi} \simeq \sum_{i=1}^{n_q} w_i f(F_K(\boldsymbol{\xi}_i)) \det J_K(\boldsymbol{\xi}_i), \\
\int_K \Pi_K^k f \, d\boldsymbol{x} &= \int_{\widehat{K}} \sum_{\alpha \in \widehat{K}_n^k} (\phi_\alpha \circ F_K^{-1} \circ F_K) f(\boldsymbol{x}_\alpha) \det J_K \, d\boldsymbol{\xi} \\
&= \int_{\widehat{K}} \sum_{\alpha \in \widehat{K}_n^k} \phi_\alpha(\boldsymbol{\xi}) f(\boldsymbol{x}_\alpha) \det J_K \, d\boldsymbol{\xi} \simeq \sum_{i=1}^{n_q} w_i \left(\sum_{\alpha \in \widehat{K}_n^k} \phi_\alpha(\boldsymbol{\xi}_i) f(\boldsymbol{x}_\alpha) \right) \det J_K(\boldsymbol{\xi}_i).
\end{aligned} \tag{2.28}$$

In particular, the transformation F_K from the change of variable cancels out with the inverse transformation F_K^{-1}, since the interpolant is defined on the reference element. For $\mathcal{P}^q$ elements, the elements of the Jacobian matrix are polynomials of degree $q-1$, hence the determinant is a polynomial of degree $2(q-1)$. Together with the interpolant of degree k, the last integral involves a polynomial of degree $2k(q-1)$ in $\boldsymbol{\xi}$. Quadrature points and weights to integrate exactly polynomials of arbitrary degree on simplices can be found in e.g. Solin et al. [51]. Integrals on spaces equipped with a *Riemannian metric* are discussed in Section 3.2.6.

2.6 Validity of high-order simplices

Interpolation and integration on high-order simplices bring two constraints on the reference-to-physical transformation F_K: (i) it should be invertible for all $\boldsymbol{x} \in K$, so that interpolation using the basis functions $\phi_\alpha(F_K^{-1}(\boldsymbol{x}))$ is well-defined, and (ii) the orientation should be consistent on K, so that the integrals are meaningful. The inverse function theorem guarantees that F_K is invertible if its Jacobian determinant $\mathcal{J}_K(\boldsymbol{\xi}) \triangleq \det J_K$ is nonzero on $\widehat{K}$, and, from the discussion on orientation, K is locally positively oriented everywhere if additionally $\mathcal{J}_K$ is positive. Therefore, we say that a $\mathcal{P}^q$-simplex K is **valid** if both interpolation and integration are well-defined on K, that is, if its Jacobian determinant $\mathcal{J}_K$ is strictly positive for all $\boldsymbol{\xi} \in \widehat{K}$.

As discussed in the previous section, the elements of the Jacobian matrix J_K for $\mathcal{P}^q$-simplices in $\mathbb{R}^n$ are polynomials of degree $q-1$, yielding a polynomial determinant $\mathcal{J}_K(\boldsymbol{\xi})$ of degree $n(q-1)$ in $\boldsymbol{\xi}$. As a polynomial on the reference element, $\mathcal{J}_K(\boldsymbol{\xi})$ can be expanded using the Lagrange basis functions ϕ_α of appropriate degree, however, the ϕ_α can be negative and do not yield easily computable lower bounds for $\mathcal{J}_K(\boldsymbol{\xi})$. In Johnen et al. [15] and Rochery [17], Bernstein basis polynomials $B_\alpha(\boldsymbol{\xi})$ are used instead, so that the Jacobian determinant has

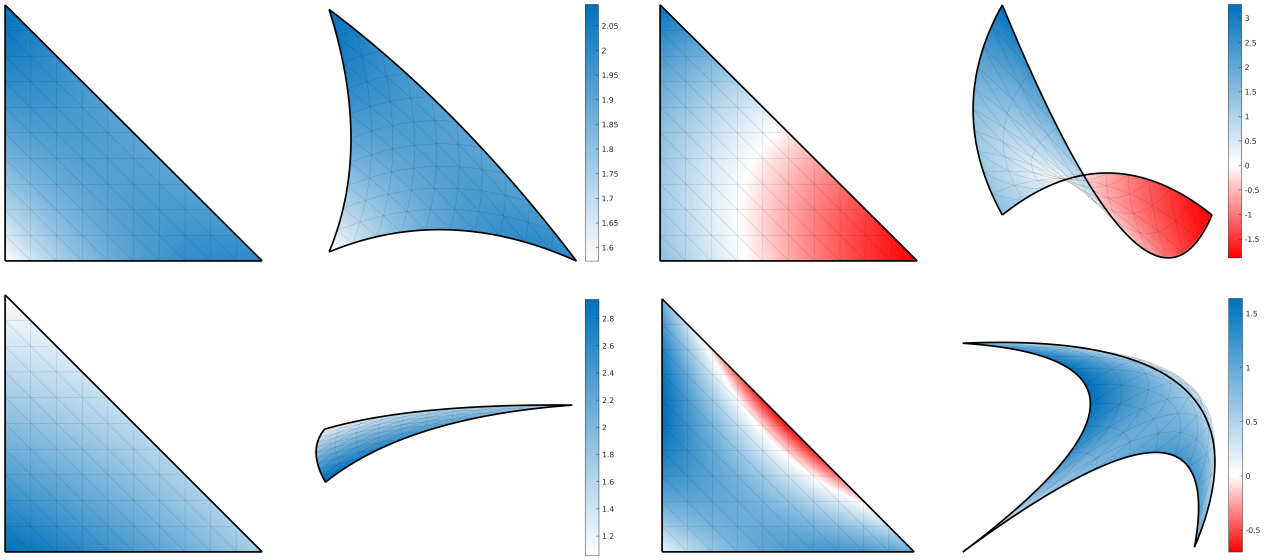


Figure 2.6 Jacobian determinant $\mathcal{J}_K(\boldsymbol{\xi})$ of some valid and invalid $\mathcal{P}^2$ triangles.

the two representations:

$$\mathcal{J}_K(\boldsymbol{\xi}) = \sum_{\alpha \in \widehat{K}_n^{n(q-1)}} \mathcal{J}_\alpha \phi_\alpha(\boldsymbol{\xi}) = \sum_{\alpha \in \widehat{K}_n^{n(q-1)}} N_\alpha B_\alpha(\boldsymbol{\xi}), \tag{2.29}$$

where the $\mathcal{J}_\alpha = \mathcal{J}_K(\boldsymbol{\xi}_\alpha)$ are the evaluations of the determinant at the interpolation nodes, and the N_α are the **Bézier control coefficients**. These can be obtained either using the change-of-basis matrix described in [15], or computed directly using the expression hereafter provided in [17]. Notably, the corner coefficients of both representations match: $\mathcal{J}_\alpha = N_\alpha$ for all corner indices noted α_{corners}, for instance $\mathcal{J}_{200} = N_{200}$, $\mathcal{J}_{020} = N_{020}$ and $\mathcal{J}_{002} = N_{002}$ for $\mathcal{P}^2$ triangles. The Bernstein basis functions B_α are always positive and form a partition of unity, thus they offer the following lower and upper bounds on the pointwise value of $\mathcal{J}_K(\boldsymbol{\xi})$:

$$N_{\min} \triangleq \min_{\alpha \in \widehat{K}_n^{n(q-1)}} N_\alpha \leq \mathcal{J}_K(\boldsymbol{\xi}) \leq N_{\max} \triangleq \max_{\alpha \in \widehat{K}_n^{n(q-1)}} N_\alpha. \tag{2.30}$$

It follows that:

$$N_{\min} \leq \min_{\boldsymbol{\xi} \in \widehat{K}} \mathcal{J}_K(\boldsymbol{\xi}) \leq \min_{\alpha \in \alpha_{\text{corners}}} N_\alpha = \min_{\alpha \in \alpha_{\text{corners}}} \mathcal{J}_\alpha. \tag{2.31}$$

To assess the positivity of $\mathcal{J}_K(\boldsymbol{\xi})$ on $\widehat{K}$, we thus have a sufficient condition: if $N_{\min} > 0$, then $\mathcal{J}_K$ is positive everywhere on $\widehat{K}$, and a necessary condition: if $\mathcal{J}_\alpha < 0$ at corner nodes, then $\mathcal{J}_K$ is negative at that node. If $N_{\min} \leq 0$, then a sharper bound must be obtained by dividing iteratively the triangle, as described in [15] and implemented in Gmsh.

In this thesis, we use these conditions to verify the validity of curved triangles during mesh generation. We implemented the explicit form of the Bézier control points given by Rochery [17] for $\mathcal{P}^q$ triangles and for $\beta \in \widehat{K}_2^{2(q-1)}$:

$$N_\beta = \frac{q!^2}{(2(q-1))!} \sum_{\substack{\alpha,\alpha' \in \widehat{K}_2^{q-1} \\ \alpha+\alpha'=\beta}} \left(\prod_{i=1}^{3} \frac{\beta_i!}{\alpha_i!\alpha'_i!} \right) |P_{\alpha+e^1} - P_{\alpha+e^3} \quad P_{\alpha'+e^2} - P_{\alpha'+e^3}|\,, \tag{2.32}$$

where the P_α are the Bézier control points in the Bézier representation of the reference-to-physical transformation $F_K = \sum_\alpha P_\alpha B_\alpha$, the e^i are the multi-indices with 1 in i-th position and 0 otherwise and $|\boldsymbol{x} \quad \boldsymbol{y}| = x^1y^2 - x^2y^1$ is the determinant of the matrix of column vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. For $\mathcal{P}^2$ triangles, the Bézier control points for F_K are given by $P_\alpha = \boldsymbol{x}_\alpha$ for corner vertices, and for instance $P_{110} = (-\boldsymbol{x}_{200} + 4\boldsymbol{x}_{110} - \boldsymbol{x}_{020})/2$ for edge control points. Those points do not lie on the boundary of K, rather, they are the intersection of the two tangent lines to their edge originating from each endpoint. With our convention for $\mathcal{P}^2$ triangles and using the displacement $\boldsymbol{a}_{12}$ associated to the edge vertex $\boldsymbol{x}_{12}$, P_{110} also writes:

$$P_{12} = 2\boldsymbol{x}_{12} - \frac{\boldsymbol{x}_1 + \boldsymbol{x}_2}{2} = \boldsymbol{x}_{12} + \boldsymbol{a}_{12}. \tag{2.33}$$

The evaluation of (2.32) for $\mathcal{P}^2$ triangles is provided by Rochery [17] and yields, with our convention in terms of the mesh vertices rather than Bézier control points:

$$\begin{aligned}
N_1 = N_{200} &= 4\,|\boldsymbol{x}_1 - (\boldsymbol{x}_{31} + \boldsymbol{a}_{31}) \quad \boldsymbol{x}_{12} + \boldsymbol{a}_{12} - (\boldsymbol{x}_{31} + \boldsymbol{a}_{31})|\,, \\
N_2 = N_{020} &= 4\,|\boldsymbol{x}_2 - (\boldsymbol{x}_{12} + \boldsymbol{a}_{12}) \quad \boldsymbol{x}_{23} + \boldsymbol{a}_{23} - (\boldsymbol{x}_{12} + \boldsymbol{a}_{12})|\,, \\
N_3 = N_{002} &= 4\,|\boldsymbol{x}_3 - (\boldsymbol{x}_{23} + \boldsymbol{a}_{23}) \quad \boldsymbol{x}_{31} + \boldsymbol{a}_{31} - (\boldsymbol{x}_{23} + \boldsymbol{a}_{23})|\,, \\
N_4 = N_{110} &= 2\,|\boldsymbol{x}_1 - (\boldsymbol{x}_{31} + \boldsymbol{a}_{31}) \quad \boldsymbol{x}_2 - (\boldsymbol{x}_{23} + \boldsymbol{a}_{23})| \\
&\quad + 2\,|\boldsymbol{x}_{12} + \boldsymbol{a}_{12} - (\boldsymbol{x}_{23} + \boldsymbol{a}_{23}) \quad \boldsymbol{x}_{12} + \boldsymbol{a}_{12} - (\boldsymbol{x}_{31} + \boldsymbol{a}_{31})|\,, \\
N_5 = N_{011} &= 2\,|\boldsymbol{x}_2 - (\boldsymbol{x}_{12} + \boldsymbol{a}_{12}) \quad \boldsymbol{x}_3 - (\boldsymbol{x}_{31} + \boldsymbol{a}_{31})| \\
&\quad + 2\,|\boldsymbol{x}_{23} + \boldsymbol{a}_{23} - (\boldsymbol{x}_{31} + \boldsymbol{a}_{31}) \quad \boldsymbol{x}_{23} + \boldsymbol{a}_{23} - (\boldsymbol{x}_{12} + \boldsymbol{a}_{12})|\,, \\
N_6 = N_{101} &= 2\,|\boldsymbol{x}_3 - (\boldsymbol{x}_{23} + \boldsymbol{a}_{23}) \quad \boldsymbol{x}_1 - (\boldsymbol{x}_{12} + \boldsymbol{a}_{12})| \\
&\quad + 2\,|\boldsymbol{x}_{31} + \boldsymbol{a}_{31} - (\boldsymbol{x}_{12} + \boldsymbol{a}_{12}) \quad \boldsymbol{x}_{31} + \boldsymbol{a}_{31} - (\boldsymbol{x}_{23} + \boldsymbol{a}_{23})|\,.
\end{aligned} \tag{2.34}$$

2.7 Motivation example: $\mathcal{P}^2$ edge curving to minimize the interpolation error

It is known since Ciarlet and Raviart [52] that the displacement $\boldsymbol{a}_{ij} = \boldsymbol{x}_{ij} - (\boldsymbol{x}_i + \boldsymbol{x}_j)/2$ should remain asymptotically small to preserve the optimal convergence rate of the interpolation error. For $\mathcal{P}^2$ triangles for instance, $\boldsymbol{a}$ should remain $\mathcal{O}(h^2)$ or higher to preserve the third

order convergence in e.g. L^2 norm. Indeed, as soon as a displacement of order h^k with $k < 2$ is imposed to a uniform mesh, the optimal rate is lost. This is illustrated in Fig. 2.7: the optimal convergence rate is lost for $k = 1$. Here, the inner edges of a structured mesh are curved by applying the displacement orthogonal to $\boldsymbol{\gamma} = \boldsymbol{x}_j - \boldsymbol{x}_i$:

$$\boldsymbol{a} = 0.1 \times \left(\frac{1}{\sqrt{N_v}}\right)^k \hat{\boldsymbol{\gamma}}^\perp \propto 0.1 \times h^k \hat{\boldsymbol{\gamma}}^\perp \quad \text{for} \quad k = 0, 1, 2, 3, \tag{2.35}$$

where $\hat{\boldsymbol{\gamma}}^\perp$ is unit and orthogonal to $\boldsymbol{\gamma}$ and the interpolated function is $f = (x^2 + y^2)^2 = r^4$. This behaviour is also reported in Botti [47] for L^2 projection and in Moxey et al. [48], and rightly discourage the use of curvilinear triangulations with reference-space interpolation when the elements are arbitrarily curved.

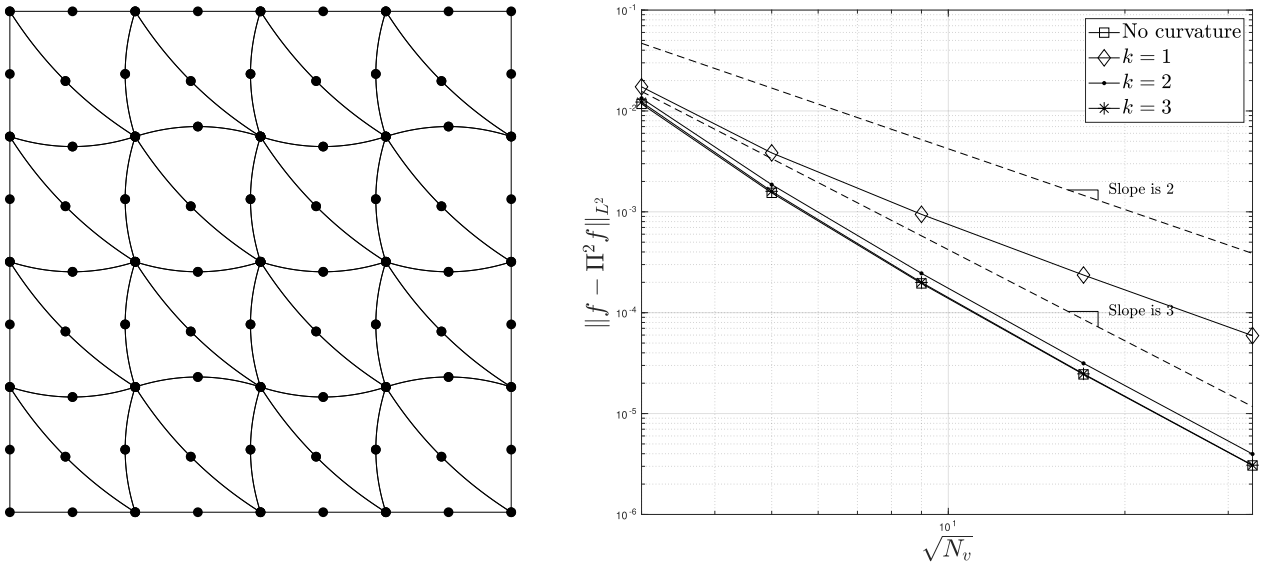


Figure 2.7 Arbitrarily curved mesh and convergence of the interpolation error.

As a motivating example, as was done by Rochery [17], it is shown here that when the edges are curved to minimize the analytical interpolation error, not only is this rate loss avoided, but the error convergence is accelerated by a rate of 0.5 to 1. Curving is thus beneficial even as the meshes becomes finer, even though the elements become asymptotically straighter. To show this, we let f be a smooth function defined on $\Omega \subset \mathbb{R}^2$ and we consider linear and quadratic interpolation. Starting from an anisotropic triangulation $\mathcal{T}$ adapted to f, the edges are curved to minimize the exact L^2 norm of the interpolation error:

$$\mathcal{E} = \|e^k f\|_{L^2(\Omega)} = \sqrt{\int_\Omega (e^k f)^2 \, dx} \simeq \sqrt{\sum_{K \in \mathcal{T}} \sum_{i=1}^{n_q} (e^k f(\xi_i))^2 \, \mathcal{J}_K(\xi_i) \, w_i}. \tag{2.36}$$

This yields superparametric $(\mathcal{P}^2, \Pi^1)$ and isoparametric $(\mathcal{P}^2, \Pi^2)$ elements respectively on the curved meshes. The edge vertices influence $\mathcal{E}$ through the transformation $F_K(\xi_i) = \sum_j \phi_j(\xi_i)\boldsymbol{x}_j$, required to evaluate $e^k f(\xi_i) = e^k f(F_K(\xi_i))$, and through the Jacobian determinant $\mathcal{J}_K(\xi_i)$. In [17], $\mathcal{E}$ is minimized sequentially, edge by edge, over pairs of triangles sharing the current edge, and the loop over the edges is repeated until the change in total error becomes lower than a given threshold. Here, we propose to minimize $\mathcal{E}$ globally by moving all the edge vertices in a single unconstrained optimization problem. Moving all edge vertices may be more expensive, but helps avoid local error minima.

Enforcing validity

Minimizing $\mathcal{E}$ alone may lead to invalid triangles. To enforce validity throughout the optimization without resorting to constrained optimization, we do as in Toulorge et al. [16] and consider a smooth logarithmic barrier [53] that prevents the Bézier control coefficients of any Jacobian determinant to become negative. A term $\mathcal{F}$ is thus added to the interpolation error to form the cost function. It writes:

$$\mathcal{F} = \sum_{K\in\mathcal{T}} \sum_{j=1}^{6} F_\epsilon\left(\frac{N_{K,j}}{\mathcal{J}_{K,0}}\right) \quad \text{with} \quad F_\epsilon(x) = \left[\ln\left(\frac{x-\epsilon}{1-\epsilon}\right)\right]^2 + (x-1)^2\,, \tag{2.37}$$

where the $N_{K,j}$ are the 6 Bézier control coefficients described in Section 2.6 and such that the degree $n(q-1) = 2$ polynomial Jacobian determinant of the $\mathcal{P}^2$ triangle K writes $\mathcal{J}_K(\boldsymbol{\xi}) = \sum_j N_{K,j} B_j(\boldsymbol{\xi})$, and $\mathcal{J}_{K,0}$ is the Jacobian determinant of the underlying $\mathcal{P}^1$ triangle associated to K. The $N_{K,j}$ depend on the edge vertices according to (2.34) while the $\mathcal{J}_{K,0}$ depend only on the corner ($\mathcal{P}^1$) vertices and are thus constant during the optimization. The log-barrier F_ϵ with $\epsilon < 1$ has a logarithmic term to prevent the Bézier coefficients to become negative and a quadratic term to penalize large coefficients, Fig. 2.8.

The sum $\mathcal{F}$ blows up as soon as any control coefficient of the mesh becomes lower than $\epsilon\mathcal{J}_{K,0}$. In the following, we simply set $\epsilon = 0$ to ensure $N_{K,j} > 0$ and that all triangles remain valid during the minimization. This is a sufficient, but not necessary condition for the associated determinant $\mathcal{J}_K$ to remain positive. The objective function is thus overly conservative and forbids valid configurations with $\min N_{K,j} \leq 0$. In [16], $\mathcal{F}$ is used to untangle invalid $\mathcal{P}^2$ triangles and the ideal Jacobian is 1 to preserve straight elements. The log-barrier F_ϵ vanishes for $x = 1$ accordingly. Here, the $\mathcal{P}^2$ triangles minimizing the interpolation error are not expected to preserve the Jacobian determinant of their underlying $\mathcal{P}^1$ triangles, thus $\mathcal{E}$ and $\mathcal{F}$ compete for different objectives. To prioritize interpolation error in the cost function while ensuring smooth increase for validity control, $\mathcal{F}$ is multiplied by a coefficient $\mu\mathcal{E}_0/\mathcal{F}_0$,

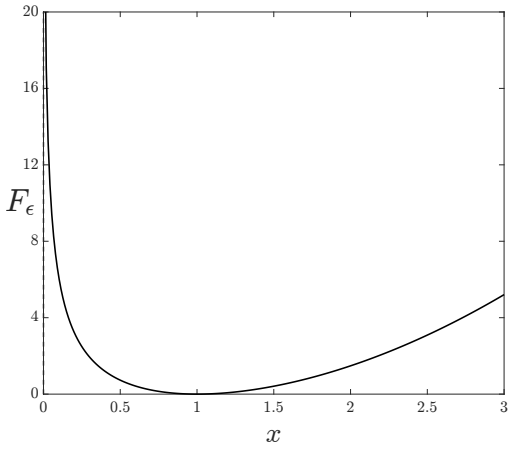


Figure 2.8 Log-barrier function for $\epsilon = 0$.

where the zero subscript indicates quantities evaluated on the initial mesh. This way, the validity term is scaled to the order of magnitude of $\mathcal{E}$, then scaled down by e.g. $\mu = 10^{-2}$. The parameter μ can be lowered on finer meshes to reach curved triangles with lower positive Jacobian determinants. The complete cost function defined this way is:

$$E = \mathcal{E} + \mu \frac{\mathcal{E}_0}{\mathcal{F}_0} \mathcal{F}. \tag{2.38}$$

Minimization and results

The starting $\mathcal{P}^1$ triangulation $\mathcal{T}$ is an anisotropic straight mesh adapted to the optimal Riemannian metric (5.48) described in detail in Chapter 5. This metric yields quasi-unit meshes minimizing the interpolation error for f and the prescribed degree k. The $\mathcal{P}^1$ vertices are kept fixed and only the edge vertices are moved to minimize the cost function. The minimization is performed with a quasi-Newton L-BFGS method via the library Ceres [54], by providing the exact gradient ∇E w.r.t. the edge vertices. This gradient is given in Appendix A.1.1. The minimization terminates when either the relative decrease in the cost function, the maximum norm of its gradient or the change in the problem parameters $\|\Delta \boldsymbol{x}_i\|$ passes below the default threshold of Ceres, which are $10^{-6}, 10^{-10}$ and 10^{-8} respectively.

We consider the following three scalar fields, shown in Fig. 2.9:

1. $f_1(x, y) = \sin x \sin y$ on $\Omega = [0, 2\pi]^2$,
2. $f_2(x, y) = \sqrt{\dfrac{1 + b^2}{(1 + b^2) \sin^2 x \cos^2 y}} \sin x \cos y$ on $\Omega = [0, 2\pi]^2$, with $b = 15$,

3. $f_3(x,y) = x^3 + x^2 y + \tanh\left(\dfrac{2x - \sin(5y)}{\delta}\right)$ on $\Omega = [-1,1]^2$, with $\delta = 0.1$.

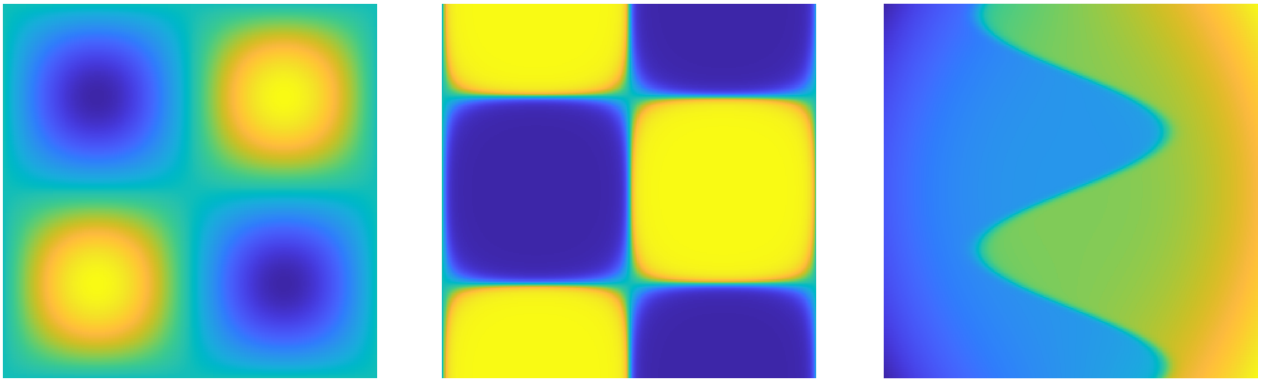


Figure 2.9 The fields f_1, f_2 and f_3.

In addition to the Lagrange interpolation error, the L^2-orthogonal projection error is reported on the curved meshes, as in Botti [47]. We are interested in a projection using reference space Lagrange basis functions. Following [47], we define the function space:

$$V_{\mathbb{P}}^k \triangleq \left\{ f_h \in L^2(\Omega) \mid \hat{f}_h(\boldsymbol{\xi}) = f_h(F_K(\boldsymbol{\xi})) \in \mathbb{P}^k(\widehat{K}), \quad \forall K \in \mathcal{T} \right\}. \tag{2.39}$$

If $\mathcal{T}$ is a curved triangulation, the elements of $V_{\mathbb{P}}^k$ are polynomial of degree k in $\boldsymbol{\xi}$, but not in $\boldsymbol{x}$ unless the reference-to-physical transformation F_K is linear. It differs from $\mathbb{P}^k(\mathcal{T})$ introduced earlier, whose elements are piecewise polynomial of degree k in $\boldsymbol{x}$. The considered projection is the operator $\pi_{V_{\mathbb{P}}^k} : L^2(\Omega) \to V_{\mathbb{P}}^k$ which preserves functions in $\mathbb{P}^{\lfloor k/q \rfloor}(K)$:

$$\pi_{V_{\mathbb{P}}^k} f = f, \quad \text{for all} \quad f \in \mathbb{P}^{\lfloor k/q \rfloor}(K). \tag{2.40}$$

For instance, let $k = q = 2$ corresponding to the projection of a square-integrable function onto the space of functions defined on a curved (quadratic) triangulation and which are piecewise quadratic w.r.t. $\boldsymbol{\xi}$. As F_K is already quadratic, $f \circ F_K$ is quadratic in $\boldsymbol{\xi}$ only if f is linear. More generally, if $f(\boldsymbol{x})$ is polynomial of degree at most p in $\boldsymbol{x}$, then $f(F_K(\boldsymbol{\xi})) \in \mathbb{P}^p(K) \circ \mathbb{P}^q(\widehat{K}) \subseteq \mathbb{P}^{pq}(\widehat{K})$, which coincides with $\mathbb{P}^k(\widehat{K})$ when $p = \lfloor k/q \rfloor$. The projection $\pi_{V_{\mathbb{P}}^k} f$ is computed by solving:

$$(\pi_{V_{\mathbb{P}}^k} f - f, v_h) = 0, \quad \text{for all} \quad v_h \in V_{\mathbb{P}}^k, \tag{2.41}$$

whose discrete finite element formulation consists in finding $f_h = \pi_{V_{\mathbb{P}}^k} f$ such that:

$$(f_h, v_h) = (f, v_h), \quad \text{for all} \quad v_h \in V_{\mathbb{P}}^k. \tag{2.42}$$

Using reference-space Lagrange finite elements of degree 1 or 2, the nodal values F_j of the projection are obtained by solving the linear system $A_{ij}F_j = b_i$, with the mass matrix $A_{ij} = (\phi_j, \phi_i)$ and the source term $b_i = (f, \phi_i)$. All the scalar products are understood in the L^2 sense, that is, $(f, g) = \int_\Omega fg\, d\boldsymbol{x}$.

From classical theory of interpolation on smooth functions, the error on straight meshes with degree k interpolants is expected to behave as $\|e^k f\|_{L^2} \leq Ch^{k+1}$ and $|e^k f|_{H^1} \leq Ch^k$ respectively, that is, convergence with an order $k+1$ and k with respect to the characteristic size h of the mesh elements. The L_2-projection satisfies the same bounds on straight meshes, but is affected by element curvature so that these estimates reduce to $\|f - \pi_{V_{\mathbb{P}}^k} f\|_{L^2} \leq Ch^{\lfloor k/q \rfloor + 1}$ when $q > 1$ [47], similarly to the interpolation error discussed in Section 2.4. Here, the starting triangulation is anisotropic and no longer described by a unique mesh size. Instead, the error is written in terms of the number of mesh vertices N_v, which asymptotically satisfies $N_v = Ch^{-n}$ for isotropic meshes, with n the space dimension. This yields a priori estimates $\|e^k f\|_{L^2} = CN_v^{-(k+1)/n}$ and $|e^k f|_{H^1} = CN_v^{-k/n}$ in terms of N_v, so that the optimal convergence rates in 2D are expected with respect to $\sqrt{N_v}$. They are illustrated hereafter as the negative slope of the error in a $(\sqrt{N_v}, \|e\|)$ log graph.

For each field, the curved meshes for $(\mathcal{P}^2, \Pi^1)$ and $(\mathcal{P}^2, \Pi^2)$ discretizations are shown in Fig. 2.14. The interpolation and projection error in L^2 norm and H^1 seminorm are reported in Figs. 2.10 through 2.12. For $(\mathcal{P}^1, \Pi^1)$ and $(\mathcal{P}^1, \Pi^2)$ interpolation, that is, interpolation on the starting linear meshes (blue lines), the optimal rates in L^2 norm of 2 and 3 respectively are observed. Similarly, rates of 1 and 2 in H^1 seminorm are reported for both the interpolation and the projection error. The starting anisotropic triangulations for f_2 and f_3 even exhibit convergence of order up to 3.3 (L^2) and 2.3 (H^1), which is not unusual for higher-order and metric-based adaptation with linear meshes, as also reported in Chapter 5.

On all optimized meshes, a gain of at least half an order of convergence is observed for both norms for the considered mesh densities. This gain is observed for both the interpolation and the L_2-orthogonal projection error, using reference space basis functions in each case. It directly translates into less vertices (hence, fewer degrees of freedom) needed by a curved mesh to reach an error level comparable to the one obtained with a straight mesh. Moreover, an improved convergence rate indicates that curving becomes more beneficial as the meshes are refined, whereas one could a priori expect the opposite. The same observations were made by Rochery [17]. Although the meshes were optimized for the L^2 norm, a matching gain is observed for the H^1 seminorm. For the tested applications, curving appears to be more

beneficial for linear $(\mathcal{P}^2, \Pi^1)$ rather than quadratic interpolation, with gains going almost up to a full order for the H^1 seminorm and the L^2 norm of f_1 and f_2. For f_1, this amounts to a ratio of up to 36 between error $\|e^1 f\|_{L^2}$ norm on the finest straight and curved meshes. Lastly, the max and 2-norm of the displacement $\boldsymbol{a}$ of the optimized curves meshes for f_1 are shown in Fig. 2.13. These are defined from the magnitude of the displacement vector at each edge as follows:

$$\|\boldsymbol{a}\|_{\ell^2} \triangleq \sqrt{\sum_{i=1}^{N_{\text{edges}}} \|\boldsymbol{a}_i\|_2^2}, \qquad \|\boldsymbol{a}\|_{\ell^\infty} \triangleq \max_{i=1\to N_{\text{edges}}} \|\boldsymbol{a}_i\|_2. \tag{2.43}$$

For both linear and quadratic interpolation, the curvature decreases in $\mathcal{O}(\sqrt{N_v}^{-1}) \sim \mathcal{O}(h)$, slower than required in [52]. Unlike Fig. 2.7, this does not prevent the curved meshes to exhibit the optimal convergence rates, as the curvature is guided by the field of interest and is no longer arbitrary. These are encouraging results, which show that the reduced $\lfloor k/q \rfloor + 1$ convergence rate for both errors can be mitigated by curving along privileged directions.

2.8 Conclusion

This introductory chapter presented the notations associated to high-order simplices, with a strong emphasis on $\mathcal{P}^2$ Lagrange triangles. Because it plays a fundamental role in mesh adaptation, transformations and Jacobian matrices associated to the reference regular element $K_\triangle$ (i.e., the ideal element) were discussed. Reference and physical space interpolation, as well as the interpolation error, were introduced on Lagrange simplices. Interpolation error plays a key role in the rest of this thesis, as the developed adaptation methods aim at minimizing this quantity. Interpolation and integration require the triangles to be positively and consistently oriented, yielding the definition of validity, which can be assessed on high-order simplices from their Bézier control coefficients. Lastly, as a motivation to create optimal curvilinear meshes, we presented, as did Rochery [17], examples of $\mathcal{P}^2$ meshes which considerably decrease the interpolation error in both L^2 norm and H^1 seminorm. As a proof of concept, these meshes are obtained by direct minimization of the interpolation error and not through metric-based generation. The goal of this thesis is precisely to work towards extending the metric-based framework developed by Alauzet and Loseille [1, 8, 9], so that meshes with similar properties can be created in a cheaper, robust and automatic fashion.

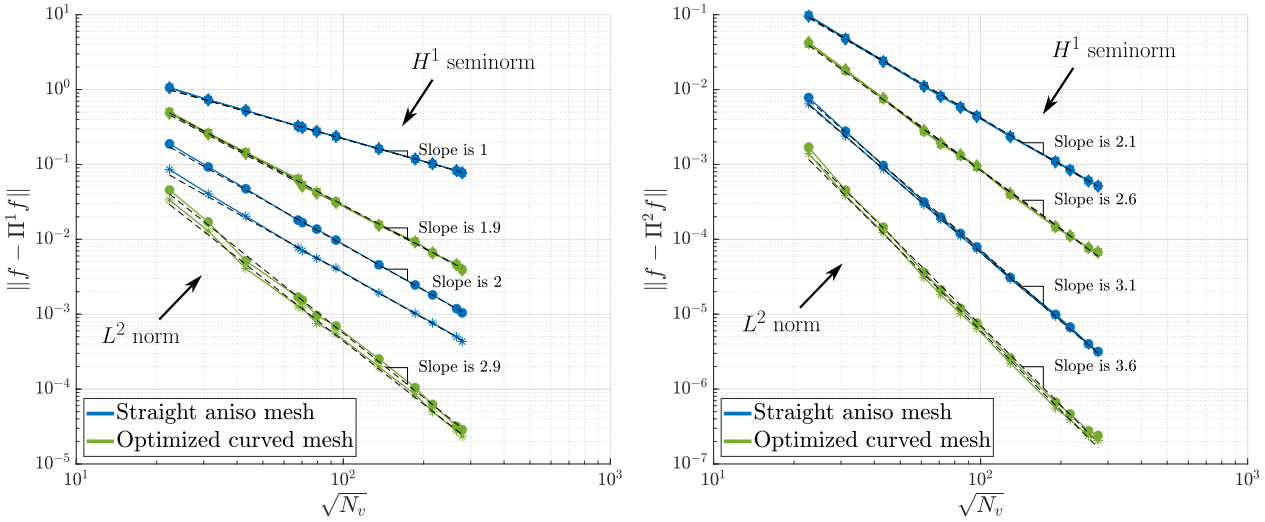


Figure 2.10 Error for f_1 with linear (left) and quadratic (right) interpolation. Interpolation error in L^2 norm (●) and H^1 seminorm (♦) and L_2-orthogonal projection error in L^2 norm (∗) and H^1 seminorm (■).

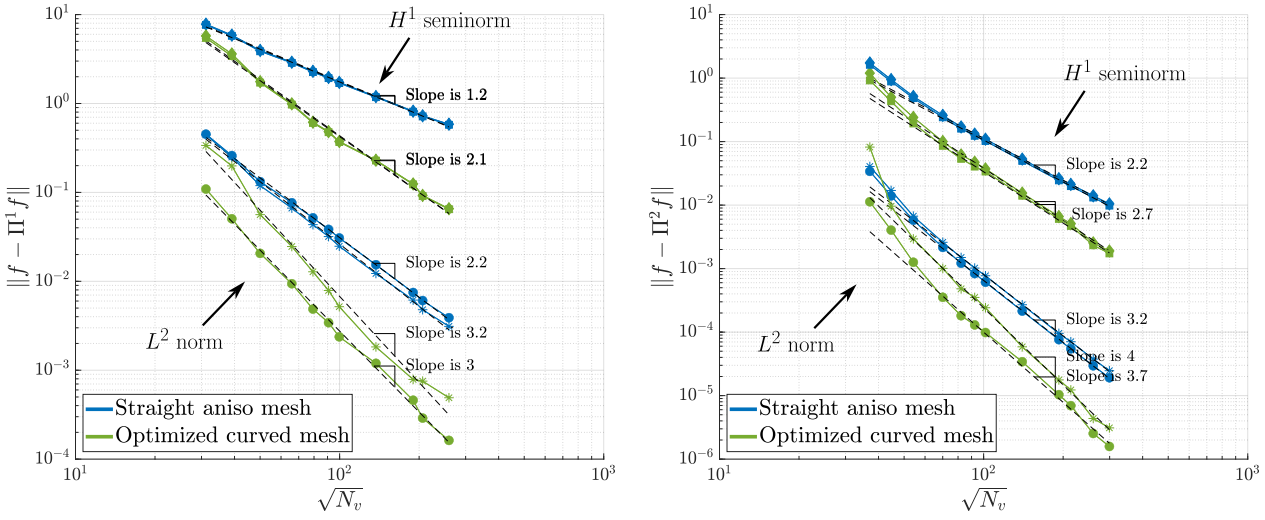


Figure 2.11 Error for f_2 with linear (left) and quadratic (right) interpolation. Interpolation error in L^2 norm (●) and H^1 seminorm (♦) and L_2-orthogonal projection error in L^2 norm (∗) and H^1 seminorm (■).

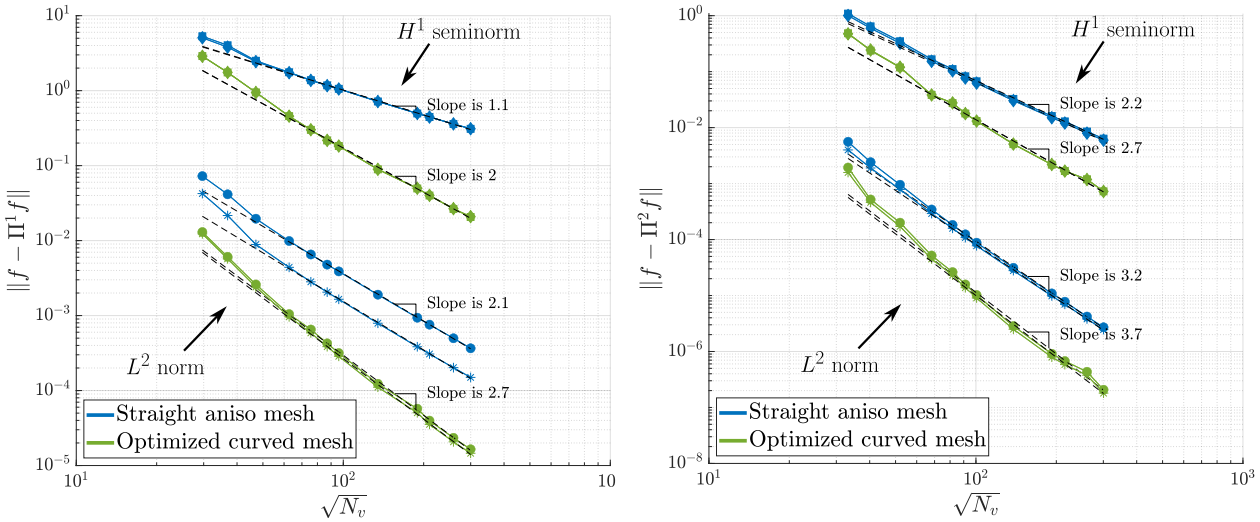


Figure 2.12 Error for f_3 with linear (left) and quadratic (right) interpolation. Interpolation error in L^2 norm (●) and H^1 seminorm (♦) and L_2-orthogonal projection error in L^2 norm ($*$) and H^1 seminorm (■).

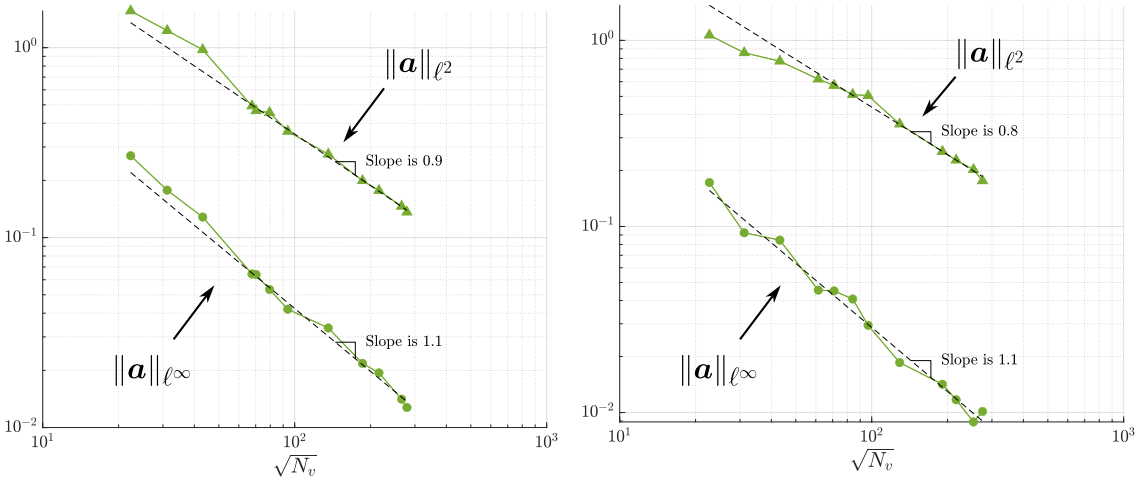


Figure 2.13 Displacement of the edge vertices for f_1 and for $(\mathcal{P}^2, \Pi^1)$ (left) and $(\mathcal{P}^2, \Pi^2)$ (right) discretizations.

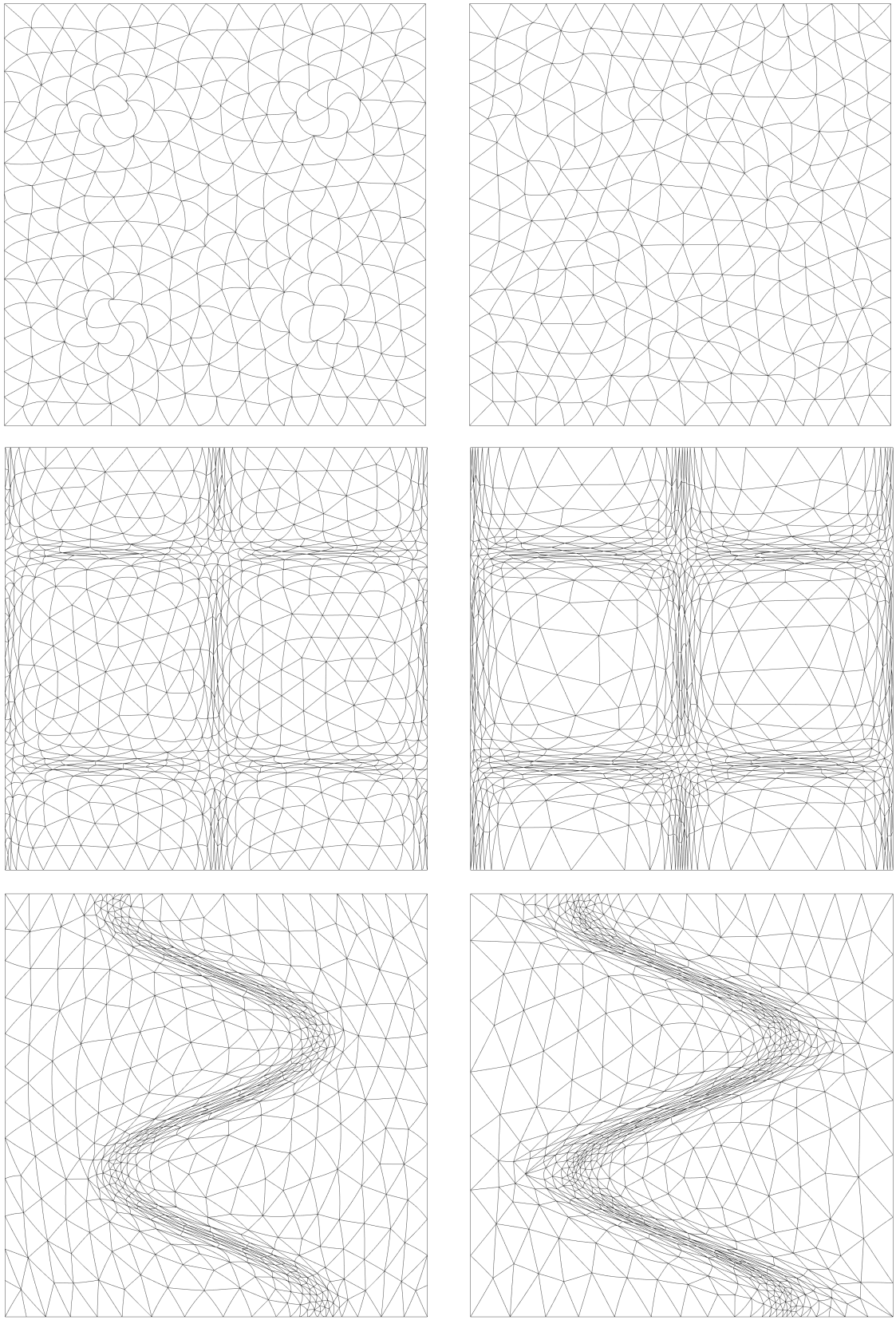


Figure 2.14 Optimized meshes for $(\mathcal{P}^2, \Pi^1)$ (left) and $(\mathcal{P}^2, \Pi^2)$ (right) discretizations.

CHAPTER 3 RIEMANNIAN METRICS

Anisotropic adaptation produces stretched and distorted meshes, where elements may exhibit high aspect ratios. In its simplest form, a *Riemannian metric* prescribes how inner products, and thus lengths and angles, are computed on surfaces and volumes. The shape and aspect ratio of mesh elements can be locally encoded in a Riemannian metric, making it a natural tool for anisotropic adaptation. It is the choice made in this thesis, which builds on the *continuous mesh framework* based on Riemannian metrics [1, 8, 9, 55]. We should stress that anisotropic adaptation need not be performed with Riemannian metrics, see for example the methods developed by Bois et al. [36] and Fortin et al. [6, 37]. Formally, Riemannian metrics are *tensor fields* defined on *smooth manifolds*, turning them into *Riemannian manifolds*. In the mesh adaptation literature, it is usual to present Riemannian metrics as fields of symmetric positive-definite (SPD) matrices defined on $\mathbb{R}^n$. Nonetheless, useful insight can be gained from a proper introduction using manifolds. More importantly, certain notions of Riemannian geometry motivate the results of this thesis. In Chapter 4, for instance, we propose a broader definition of *unit elements*, a key concept of the continuous mesh framework, based on Riemannian isometries. Such isometries are the isomorphisms between Riemannian manifolds, that is, the structure-preserving applications. To properly define these isomorphisms, elements of differential geometry first need to be introduced: smooth manifolds, tangent vectors and covectors as well as the spaces they span, tensors and pullbacks of tensors. Then, *Riemannian geometry*, which studies manifolds related by isometries, provides results justifying their use in mesh generation. In particular, *unit meshes*, when viewed as isometric transformation of a reference tiling, only exist on manifolds isometric to $\mathbb{R}^2$. These manifolds are characterized by their absence of *curvature*, a local quantity invariant under isometries.

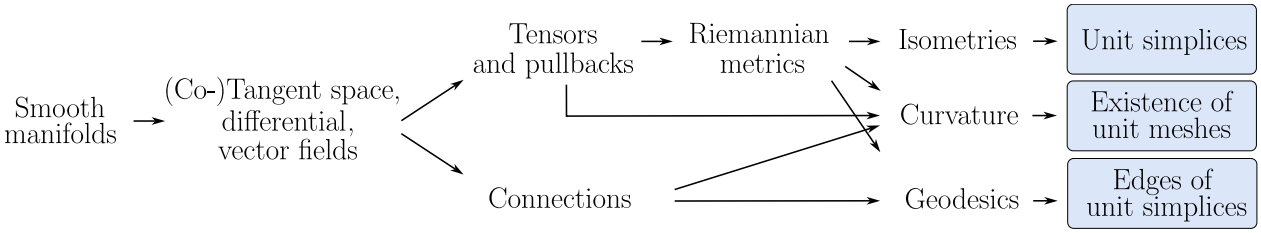
The goal of this chapter is to present these ideas in a self-contained form. They are introduced in a formal manner, but we try to give an intuitive interpretation of each quantity and to provide illustrative examples. The material of this chapter is basic notions of differential geometry, without original contribution. The presentation is adapted from Lee's books on smooth and Riemannian manifolds [46, 56], to which we refer for proofs, precisions and additional examples. Lee's convention for the definitions is also adopted here: the terms are written in bold when they are defined. Recurring concepts of geometry used in mesh adaptation, such as Riemannian metrics, additionally have an emphasized Definition.

Plan of the chapter

Before diving into the presentation, we give here the big picture of the chapter. We start in Section 3.1 with smooth manifolds, which are abstract generalizations of curves and surfaces, on which we can define smooth functions. At each point $\boldsymbol{p}$ of a smooth manifold is attached another manifold of the same dimension, made of all the vectors tangent to the manifold at $\boldsymbol{p}$, called the tangent space. The tangent space is a linear approximation of the manifold at $\boldsymbol{p}$, and is where inner products are defined. Tangent vectors at $\boldsymbol{p}$ can be thought of as the velocity vectors of curves passing through $\boldsymbol{p}$. This intuitive view of vectors makes sense because we naturally consider manifolds as part of an ambient $\mathbb{R}^m$ of higher dimension. For instance, a sphere is usually viewed as a surface of $\mathbb{R}^3$, even though it is, in itself, a 2-dimensional object. Some more abstract manifolds do not come with such a natural ambient space, so a more abstract interpretation is given to tangent vectors instead: they are no longer interpreted as *arrows* with a starting point and a length, but as directions along which one can evaluate directional derivatives. Hence, vectors become operators that one *applies* to sufficiently regular functions to yield a real number. *Covectors*, or cotangent vectors, are real-valued linear functionals taking tangent vectors as input. They are *dual* quantities to vectors, and are defined on the dual of the tangent space, called the cotangent space. The tangent and cotangent spaces are the domains of definition of tensors: these are real-valued multilinear maps taking an arbitrary number of arguments in either space. Pullbacks of tensors bring tensors on a manifold back to another, thus providing a way of comparing them. Riemannian metrics, which are of great interest in this thesis, are tensor fields taking two tangent vectors and returning a real number which is the result of their inner product. They are presented in Section 3.2, along with Riemannian isometries, which are differentiable maps preserving the Riemannian metric and are the basis of Riemannian geometry.

The following sections cover the topics of Riemannian geometry needed to discuss the curvature of manifolds. The first ingredient is a derivative operator for vector fields, in the form of the *covariant derivative*. A *connection*, and in particular the *Levi-Civita connection*, is a machine that produces such covariant derivatives. They are the generalization of the *nabla* operator for smooth and Riemannian manifolds, and are introduced in Section 3.3. Covariant differentiation yields a well-defined acceleration for curves, which in turn generalizes straight lines to Riemannian manifolds, in the form of *geodesics*, discussed in Section 3.4. Geodesics are of great significance in this work, as unit mesh elements defined as isometries have geodesic edges, see Chapter 4. The notion of curvature is introduced in Section 3.5: it measures how much a manifold deviates from the Euclidean space of the same dimension. Curvature is

invariant under isometries. In other words, two Riemannian manifolds are, in some sense, indistinguishable if they have the same curvature. As we aim at generating meshes that are essentially indistinguishable from a regular tiling of the plane, curvature plays a key role in determining when those meshes exist. Lastly, this chapter concludes in Section 3.6 on well-known useful operations on metrics for practical mesh adaptation: interpolation, intersection and gradation of metrics.



Notations and summing convention

In the following, both points on a manifold and tangent vectors are written in bold, for instance $\boldsymbol{p} \in M$ or $\boldsymbol{v} \in T_{\boldsymbol{p}}M$. The components of basis vectors are written with a lower index, such as e_i, and the components of a vector in this basis are written with an upper index, so that $\boldsymbol{v} = \sum_i v^i e_i$. Although e_i is used to denote both basis vectors and the edges of a simplex, it should always be clear from context what the symbol refers to. Einstein's summation convention is used and the summation symbol is ommitted whenever an index appears exactly once as an upper index and once as a lower index inside a product, so that $\boldsymbol{v}$ simply writes $v = v^i e_i$. The only exception to this rule concerns matrices which are not associated to a tensor, such as Jacobian matrices. These are written with two lower indices, e.g. J_{ij}, as it is standard in engineering. As a result, whenever matrices are involved in a product, the sum acts over repeated indices independently of their position. All partial derivatives are written with an upper index at the "denominator", for instance $\partial/\partial x^i$, but this index is still considered a lower index. The Kronecker delta is written either δ_{ij} or δ^i_j depending on the context, and is 1 if $i = j$ independently of the position of i and j, and 0 otherwise.

3.1 Smooth manifolds

In essence, a **manifold** is the generalization of geometric objects such as curves, surfaces and volumes, which can then be classified as manifolds of dimension 1, 2, and 3 respectively. As an informal introduction, an $n-$dimensional manifold M, or simply an n**-manifold**, is a space that "looks locally like the Euclidean space $\mathbb{R}^n$, but globally might be more complicated, such

as a sphere or a torus" [57]. Here, "looks locally like $\mathbb{R}^n$" means that the space is an assembly of subsets $U \subseteq M$ that can each be mapped to $\mathbb{R}^n$ by a bijection φ, called a **coordinate map**. The pair (U, φ) is called a **chart**, Fig. 3.1, and how these charts overlap with one another determines the smoothness of the manifold. The inverse map $\varphi^{-1} : \varphi(U) \to U$ is called a **local parameterization** of M. If $\boldsymbol{p}$ is a point in U, the functions $\varphi(\boldsymbol{p}) = (x^1, \ldots, x^n)$ are called the **local coordinates** of $\boldsymbol{p}$ on U. When $U = M$, they are also **global coordinates** on M. A collection of charts is (rightfully) named an **atlas**.

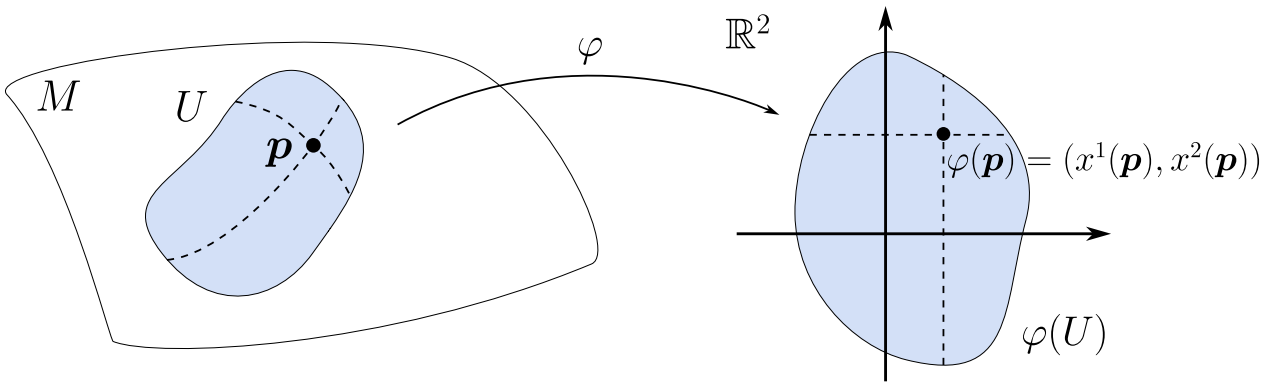


Figure 3.1 A chart (U, φ) defined on $U \subset M$.

Examples and counterexample of manifolds

- The Euclidean space $\mathbb{R}^n$ is the simplest example of manifold: it is described by an atlas made of a single chart, whose domain is $U = \mathbb{R}^n$ and the local coordinates are obtained everywhere from the identity map $\varphi = \mathrm{Id}$ such that $\varphi(\boldsymbol{p}) = x^i = p^i$. They are also global coordinates.
- The surface described by the graph G of a function $f : \mathbb{R}^n \to \mathbb{R}$, defined by the set of $(\boldsymbol{x}, f(\boldsymbol{x}))$ in $\mathbb{R}^{n+1}$ is another example of manifold, which is frequently used in this thesis. When $n = 2$, it is also called a *Monge patch*. Defining the projection $\varphi(\boldsymbol{x}, f(\boldsymbol{x})) = \boldsymbol{x}$, the chart (G, φ) provides global coordinates for G by simply projecting the first term onto $\mathbb{R}^n$.
- The sphere in $\mathbb{R}^3$, on the other hand, is a manifold that needs to be covered by at least two charts (using e.g. stereographic projections), hence there exist no global coordinates on the sphere.
- In numerical geometry, meshes can sometimes form nonmanifold configurations, such as the rightmost image of Fig. 3.2. This surface is a 2-manifold everywhere except at the common edge shared by the 4 triangles: one can travel on all triangles with two parameters, but not at the common edge.

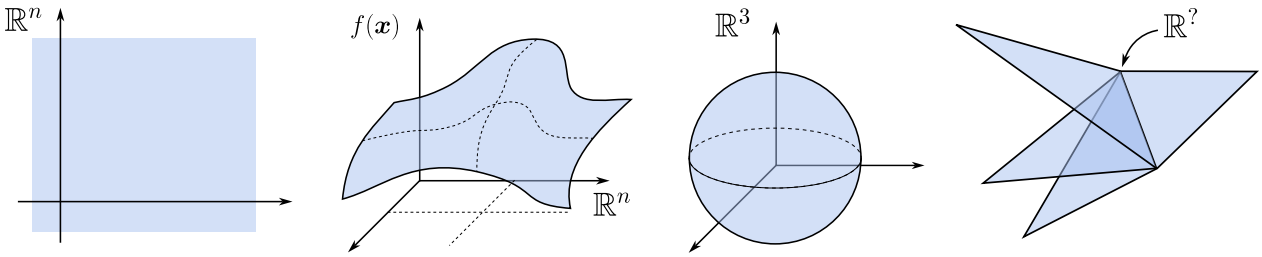


Figure 3.2 Examples and counterexample of manifolds.

Smooth structure and smooth maps

The simplest manifolds are called *topological manifolds*. They can be equipped with additional structures depending on the target application. In this thesis, we are concerned with Riemannian manifolds, which are **smooth manifolds** equipped with a way of computing inner products, called a Riemannian metric. Smooth manifolds are a very important special case of topological manifolds: they are those on which one can define calculus over smooth (i.e., infinitely differentiable or C^∞) functions, that is, compute integral and derivatives. On a smooth manifold, the charts are glued together seamlessly. Formally, this means that in the overlapping region of two neighbouring charts (U, φ) and (V, ψ), one can go from the image $\varphi(U \cap V) \in \mathbb{R}^n$ to the image $\psi(U \cap V) \in \mathbb{R}^n$ by a C^∞ composition $\psi \circ \varphi^{-1} : \mathbb{R}^n \to \mathbb{R}^n$, called the **transition map** from φ to ψ. This is illustrated in Fig. 3.3. The transition map is a **change of coordinates** for the points in $U \cap V$, as these points can be described by both their coordinates $\varphi(\boldsymbol{p})$ and $\psi(\boldsymbol{p})$.

An atlas is said to be smooth if all of its charts overlap smoothly, and the largest smooth atlas on a manifold constitutes the **smooth structure** of the manifold. A smooth manifold is formally a pair $(M, \mathcal{A})$, where M is a topological manifold and $\mathcal{A}$ is a smooth atlas. In other words, a smooth structure on a manifold is simply a requirement of compatibility between its charts. Thanks to this compatibility, we can define **smooth maps** on and between manifolds by going back and forth to $\mathbb{R}^n$. These maps are defined as follows. Let M and N be smooth manifolds of dimension m and n, respectively. A map $f : M \to N$ is smooth if for $U \subseteq M$ and $V \subseteq N$ and for every $\boldsymbol{p} \in M$, there are charts (U, φ) containing $\boldsymbol{p}$ and (V, ψ) containing $f(\boldsymbol{p})$ such that the composition $\psi \circ f \circ \varphi^{-1} : \mathbb{R}^m \to \mathbb{R}^n : \varphi(U) \mapsto \psi(V)$ is smooth. All the maps in the following are smooth, so that the image of a manifold has the same compatibility between its charts and is also a smooth manifold. If the map has a smooth inverse, it is called a **diffeomorphism**. Because smooth manifolds do not yet have a notion of lengths or angles, two such manifolds are indistinguishable if they are related by a diffeomorphism. Smooth

real-valued functions $f : M \to \mathbb{R}$ are the special case of smooth maps with $N = \mathbb{R}$ and $\psi = \mathrm{Id}$, and we denote by $C^\infty(M)$ the set of all smooth real-valued functions whose domain is in M.

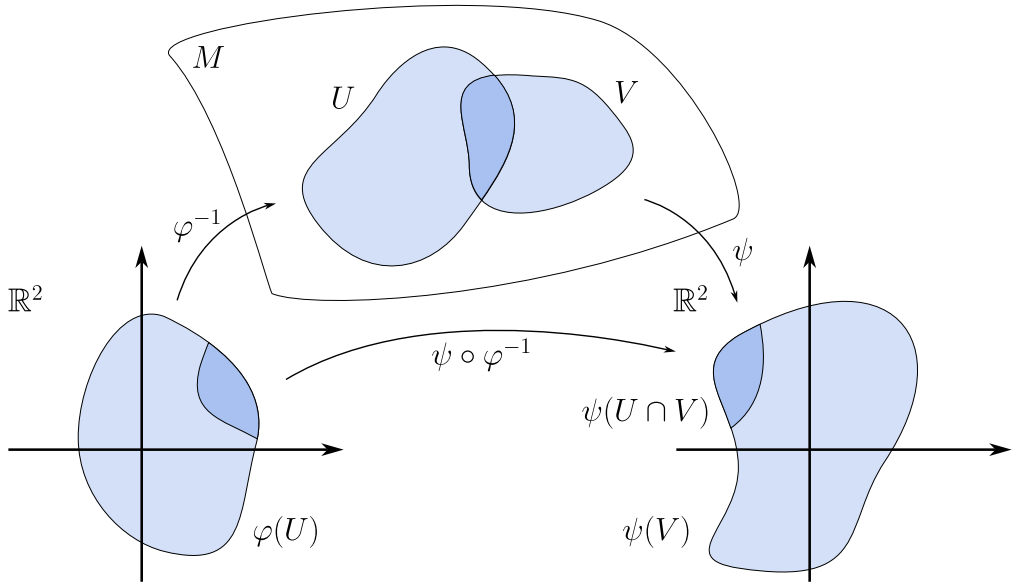


Figure 3.3 Compatible charts and transition map.

3.1.1 Tangent vectors, tangent spaces and differential

A tangent vector at a point $\boldsymbol{p} \in M$ is intuitively an arrow originating at $\boldsymbol{p}$ and that is tangent to M, forming a linear approximation of M at $\boldsymbol{p}$. In the more abstract context of smooth manifolds, it is actually more tractable to identify tangent vectors to directional derivatives of smooth functions, called **derivations** and satisfying Leibniz's rule. This comes from the fact that a direction $\boldsymbol{v}$ defines the directional derivative of functions in $C^\infty(M)$ by:

$$D_{\boldsymbol{v}} f(\boldsymbol{p}) = \frac{d}{dt}\bigg|_{t=0} f(\boldsymbol{p} + t\boldsymbol{v}) = v^i \frac{\partial f}{\partial x^i}(\boldsymbol{p}), \tag{3.1}$$

which satisfies Leibniz's rule:

$$D_{\boldsymbol{v}}(fg)(\boldsymbol{p}) = D_{\boldsymbol{v}} f(\boldsymbol{p}) g(\boldsymbol{p}) + f(\boldsymbol{p}) D_{\boldsymbol{v}} g(\boldsymbol{p}). \tag{3.2}$$

With this in mind, we call a linear map[1] $\boldsymbol{v} : C^\infty(M) \to \mathbb{R}$ a **derivation** at $\boldsymbol{p}$ if for any $f, g \in C^\infty(M)$, it satisfies:

$$\boldsymbol{v}(fg) = \boldsymbol{v}(f)g(\boldsymbol{p}) + f(\boldsymbol{p})\boldsymbol{v}(g). \tag{3.3}$$

The set of all derivations at $\boldsymbol{p}$ is called the **tangent space to M at $\boldsymbol{p}$** and is noted $T_{\boldsymbol{p}}M$. Its elements are called **tangent vectors at $\boldsymbol{p}$**. As illustrated further, it turns out that any derivation at $\boldsymbol{p}$ can be identified with the velocity vector of a curve on M passing through $\boldsymbol{p}$, thus reconciling the intuitive interpretation of a tangent vector with its more abstract definition as a derivation. The tangent space $T_{\boldsymbol{p}}M$ is a vector space that has the same dimension as the manifold itself. For example, the tangent space to any point $\boldsymbol{p} \in \mathbb{R}^2$ is a copy of $\mathbb{R}^2$ centered at $\boldsymbol{p}$, and the tangent space to a surface (a 2-manifold) in $\mathbb{R}^3$ is a plane, Fig. 3.4. The union of all tangent spaces at all points of an $n-$manifold forms another manifold of dimension $2n$, called the **tangent bundle of** M, noted TM. The tangent bundle, as well as its dual, the cotangent bundle T^*M introduced hereafter, are of particular significance because they are the spaces over which tensor fields (such as Riemannian metrics) are defined.

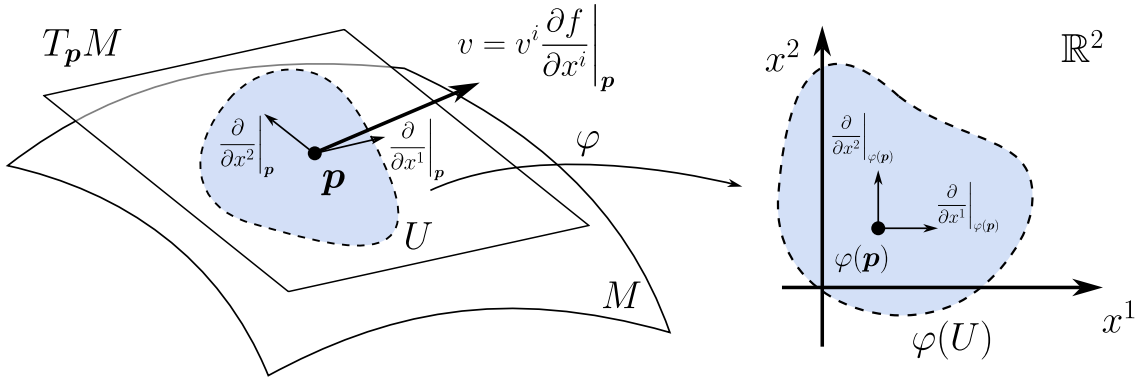


Figure 3.4 The tangent space to of a surface M at $\boldsymbol{p}$.

In light of (3.3), it makes sense to say that we *apply* a tangent vector to a smooth function, and, although it is somewhat unusual in engineering, a basis for the vector space $T_{\boldsymbol{p}}M$ is given by the operators of partial derivatives $\partial/\partial x^i|_{\boldsymbol{p}}$. Indeed, once applied to a smooth function f, these vectors are the partial derivatives $\partial f/\partial x^i(\boldsymbol{p})$ and span the directional derivatives (3.1). A tangent vector $\boldsymbol{v}$ can then be written as:

$$\boldsymbol{v} = v^i \frac{\partial}{\partial x^i}\bigg|_{\boldsymbol{p}} \quad \text{such that} \quad \boldsymbol{v}f \triangleq \boldsymbol{v}(f) = v^i \frac{\partial f}{\partial x^i}(\boldsymbol{p}) = D_{\boldsymbol{v}}f(\boldsymbol{p}) \in \mathbb{R}. \tag{3.4}$$

[1]It is unusual to write a real-valued map in bold, but this allows to be consistent with the engineering notations where vectors are usually in bold.

The x^i with respect to which we take the partial derivatives are the local coordinates of points on the manifold. In other words, any choice of coordinates x^i on $U \subseteq M$ *induces* a basis $\partial/\partial x^i$ for the tangent spaces at each point $\boldsymbol{p} \in U$, known as a **coordinate basis**. These different notions are illustrated in Fig. 3.4. The tangent space plays an important role in anisotropic mesh adaptation: this is where norms and angles are measured once the manifold is endowed with a metric. The length of a curve is obtained by integration, summing the contributions of lengths computed along tangent vectors.

Differential of a smooth map

In calculus, the derivative or Jacobian matrix of a function F locally describes its best linear approximation. On manifolds, this is generalized by the **differential** of F, which acts linearly between tangent spaces. More precisely, if M and N are smooth manifolds, the differential of a smooth map $F : M \to N$ at $\boldsymbol{p}$ is a linear map $dF_{\boldsymbol{p}} : T_{\boldsymbol{p}}M \to T_{F(\boldsymbol{p})}N$ that associates to the derivation $\boldsymbol{v}$ at $\boldsymbol{p}$, another derivation $dF_{\boldsymbol{p}}(\boldsymbol{v})$ at $F(\boldsymbol{p})$. The latter acts on smooth functions of $C^\infty(N)$ as follows:

$$dF_{\boldsymbol{p}}(\boldsymbol{v})(f) = \boldsymbol{v}(f \circ F), \tag{3.5}$$

that is, it applies $\boldsymbol{v}$ to the composition $f \circ F$ viewed as a smooth function on M, Fig 3.5.

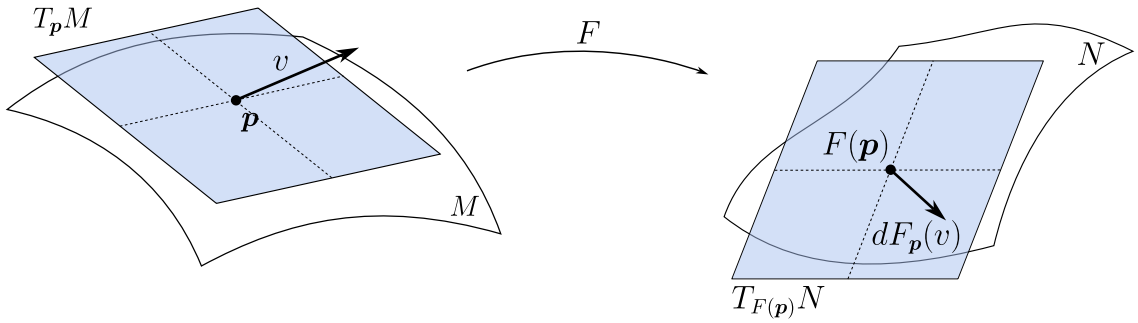


Figure 3.5 Differential of a smooth map $F : M \to N$.

This rather abstract and coordinate-free form of the differential becomes clearer when written in coordinates. For instance, letting F be a map from $U \subseteq \mathbb{R}^n$ with coordinates x^i to $V \subseteq \mathbb{R}^m$ with coordinates y^i, the action of the differential on the basis vector $\partial/\partial x^i|_{\boldsymbol{p}} \in T_{\boldsymbol{p}}M$ writes:

$$dF_{\boldsymbol{p}}\left(\left.\frac{\partial}{\partial x^i}\right|_{\boldsymbol{p}}\right)(f) = \left.\frac{\partial}{\partial x^i}\right|_{\boldsymbol{p}}(f \circ F) = \left.\frac{\partial f}{\partial y^j}\right|_{F(\boldsymbol{p})}\left.\frac{\partial F^j}{\partial x^i}\right|_{\boldsymbol{p}} = \left(\left.\frac{\partial F^j}{\partial x^i}\right|_{\boldsymbol{p}}\left.\frac{\partial}{\partial y^j}\right|_{F(\boldsymbol{p})}\right) f. \tag{3.6}$$

The term in the last parentheses is the result of the differential written as an element of $T_{F(\boldsymbol{p})}N$, i.e., as a linear combination of the basis vectors $\partial/\partial y^j|_{\boldsymbol{p}}$: its components $\partial F^j/\partial x^i|_{\boldsymbol{p}}$

are precisely the $n \times m$ Jacobian matrix of F at $\boldsymbol{p}$. Hence, the differential of a map can be seen as a coordinates-free generalization of its Jacobian matrix. Having identified the action of $dF_{\boldsymbol{p}}$ on the basis vectors, its action on any tangent vector v writes:

$$dF_{\boldsymbol{p}}(\boldsymbol{v}) = dF_{\boldsymbol{p}}\left(v^i \left.\frac{\partial}{\partial x^i}\right|_{\boldsymbol{p}}\right) = v^i dF_{\boldsymbol{p}}\left(\left.\frac{\partial}{\partial x^i}\right|_{\boldsymbol{p}}\right) = v^i \left.\frac{\partial F^j}{\partial x^i}\right|_{\boldsymbol{p}} \left.\frac{\partial}{\partial y^j}\right|_{F(\boldsymbol{p})} \in T_{F(\boldsymbol{p})}N. \tag{3.7}$$

Since the differential "pushes" tangent vectors from a manifold to another, it is also referred to as the **pushforward**. This differential is somewhat different from the usual total variation of a real-valued function f:

$$df = \frac{\partial f}{\partial x^i} dx^i. \tag{3.8}$$

In the next section, the symbols dx^i are properly introduced, and we shall see that both concepts really represent the same thing.

Velocity vector of a curve

A **curve (segment) on** M is a continuous map $\gamma : I \to M$, where $I \subseteq \mathbb{R}$ is an interval of real numbers, Fig. 3.6. The usual velocity vector of γ extends to manifolds using the differential introduced above. The **velocity of** γ **at** $t_0 \in I$ is the tangent vector defined by:

$$\boldsymbol{\gamma}'(t_0) \triangleq d\gamma_{t_0}\left(\left.\frac{d}{dt}\right|_{t_0}\right) \in T_{\gamma(t_0)}M. \tag{3.9}$$

Here, $d/dt = \partial/\partial t$ is the coordinate basis vector for the tangent space $T_{t_0}I = T_{t_0}\mathbb{R}$. In other words, the velocity is the pushforward on $T_{\gamma(t_0)}M$ of the tangent vector $d/dt \in T_{t_0}I$. The velocity acts on smooth functions by returning the total derivative of the composition with the curve:

$$\gamma'(t_0)f \overset{(3.5)}{=} \left.\frac{d}{dt}\right|_{t_0}(f \circ \gamma) = (f \circ \gamma)'(t_0). \tag{3.10}$$

From (3.6) and letting $\gamma(t) = (\gamma^1(t), \ldots, \gamma^n(t))$ denote the representation in coordinates of the curve γ, the velocity writes:

$$\boldsymbol{\gamma}'(t_0) = \left.\frac{d\gamma^i}{dt}\right|_{t_0} \left.\frac{\partial}{\partial x^i}\right|_{\gamma(t_0)}. \tag{3.11}$$

Thus, the components of the velocity vector in the tangent space are the derivatives of each of its components with respect to the parameter t, as expected from geometry in the Euclidean space. As we will see in Section 3.3, this is not the case for the acceleration vector, which

requires a new tool, called a connection, to be defined in a coordinate-independent way.

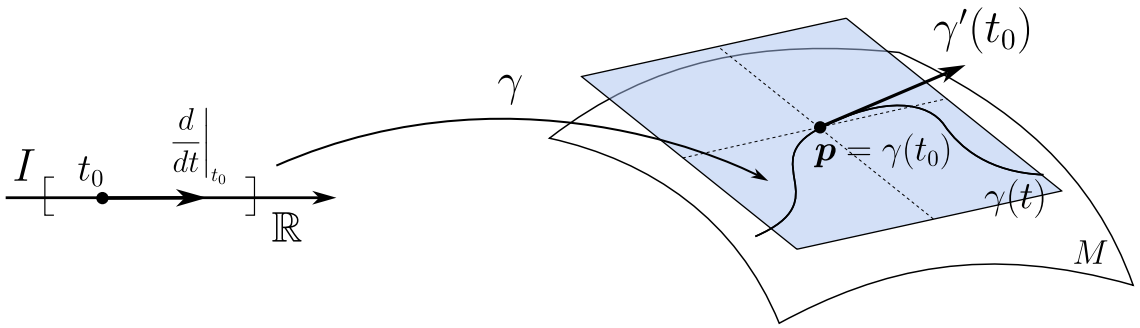


Figure 3.6 The velocity of a curve as a tangent vector $\boldsymbol{\gamma}'(t_0) \in T_{\boldsymbol{p}}M$.

Vector fields and local frames

Similarly, vector fields on smooth manifolds are a natural extension of vector fields in multivariable calculus. A **vector field** X on M is an application that assigns to each point of M an element of its tangent space, i.e., a tangent vector:

$$X : M \to TM : \boldsymbol{p} \mapsto X_{\boldsymbol{p}} = \boldsymbol{v}_{\boldsymbol{p}} \in T_{\boldsymbol{p}}M. \tag{3.12}$$

In coordinates x^i, the value of X at $\boldsymbol{p}$ writes:

$$X_{\boldsymbol{p}} = X^i(\boldsymbol{p}) \frac{\partial}{\partial x^i}\bigg|_{\boldsymbol{p}}, \tag{3.13}$$

where the X^i are the component functions of X. The vector field X is smooth if its component functions are smooth, and the set of all **smooth vector fields** on M is noted $\mathfrak{X}(M)$. The simplest example of smooth vector fields are the **coordinate vector fields**, noted $\partial/\partial x^i$. They point in the direction along which the i-th coordinate increases and are defined by:

$$\frac{\partial}{\partial x^i} : \boldsymbol{p} \mapsto \frac{\partial}{\partial x^i}\bigg|_{\boldsymbol{p}}. \tag{3.14}$$

The coordinate vector fields form a basis for the tangent space at each point. More generally, a **local frame for** M is an ordered n-tuple of vector fields $(E_1, \ldots, E_n)$, simply noted (E_i), defined on an open subset $U \subseteq M$ that spans the tangent bundle of M. That is, $(E_i|_{\boldsymbol{p}})$ is a basis for the tangent space at $\boldsymbol{p} \in U$. If $U = M$, then (E_i) is also a **global frame**. Given the chart (U, x^i), the coordinate vector fields $(\partial/\partial x^i)$ are thus a local frame for U, called a **coordinate frame**. In $\mathbb{R}^n$, a local frame whose vector fields are orthonormal (with respect

to the standard inner product of $\mathbb{R}^n$) is called an **orthonormal frame**. Orthonormal frames can be extended to abstract manifolds with a Riemannian metric, introduced hereafter, as a way of computing the inner product. It is always possible to turn a *local frame* into an orthonormal frame by applying Gram-Schmidt's procedure, however, there does not usually exist a choice of coordinates for which the *coordinate frame* is orthonormal. The obstruction to the existence of orthonormal coordinate frames is precisely the curvature of the manifold, discussed in Section 3.5.

Two vector fields $X, Y \in \mathfrak{X}(M)$ do not usually commute, that is, if $f \in C^\infty(M)$, the real-valued function $YXf = Y(Xf) = Y(X(f))$ obtained by applying Y after X is not usually equal to XYf. In fact, the compositions XY and YX are not even always vector fields, because they may not satisfy the product rule, that is, their result may not be a vector in the sense of a derivation. The operator $[X, Y] \triangleq XY - YX$, called the **Lie bracket of X and Y**, is however always a vector field. In coordinates, its action on f writes:

$$[X, Y]f = X^i \frac{\partial Y^j}{\partial x^i} \frac{\partial f}{\partial x^j} - Y^j \frac{\partial X^i}{\partial x^j} \frac{\partial f}{\partial x^i}. \tag{3.15}$$

In particular, as mixed derivatives of f commute, the Lie bracket of coordinate vector fields always vanishes:

$$\left[\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right] f = \frac{\partial}{\partial x^i}\frac{\partial f}{\partial x^j} - \frac{\partial}{\partial x^j}\frac{\partial f}{\partial x^i} = 0. \tag{3.16}$$

A local frame that satisfies $[E_i, E_j] = 0$ for all i, j is called a **commuting frame** or a **holonomic frame**. It follows that all coordinate frames are commuting frames. Lie brackets are used in Section 3.5 to help define the curvature of a manifold.

Example: Cartesian and polar basis vectors

The Euclidean space $\mathbb{R}^2$ with Cartesian coordinates (x, y) is trivially described by the global chart $(U = \mathbb{R}^2, x^i)$, with $x^1 = x$ and $x^2 = y$. The coordinate vector fields are $(\partial_x, \partial_y) \triangleq (\partial/\partial x, \partial/\partial y)$. They form an orthonormal coordinate frame that is a global frame for $\mathbb{R}^2$, which means that they form a basis of the tangent space $T_{\boldsymbol{p}}\mathbb{R}^2$ everywhere.

These vectors are also often written $(\boldsymbol{e}_x, \boldsymbol{e}_y)$. The basis vectors in polar coordinates (r, θ), with $(x, y) = (r\cos\theta, r\sin\theta)$, are particularly interesting. The coordinate vector fields are $(\partial_r, \partial_\theta) \triangleq (\partial/\partial r, \partial/\partial\theta)$, and are related to the Cartesian coordinate vectors by:

$$\begin{aligned} \partial_r &= \frac{\partial x}{\partial r}\partial_x + \frac{\partial y}{\partial r}\partial_y = \quad \cos\theta\ \partial_x + \ \sin\theta\ \partial_y \\ \partial_\theta &= \frac{\partial x}{\partial \theta}\partial_x + \frac{\partial y}{\partial \theta}\partial_y = -r\sin\theta\ \partial_x + r\cos\theta\ \partial_y. \end{aligned} \tag{3.17}$$

They are by definition a coordinate frame and they are orthogonal, but they are not orthonormal because ∂_θ is not a unit vector. An orthonormal frame can be obtained by dividing ∂_θ by r, yielding the basis vectors $(\boldsymbol{e}_r, \boldsymbol{e}_\theta)$:

$$\begin{aligned} \boldsymbol{e}_r = \partial_r = &\quad \cos\theta\ \partial_x + \sin\theta\ \partial_y \\ \boldsymbol{e}_\theta = \frac{\partial_\theta}{r} = &-\sin\theta\ \partial_x + \cos\theta\ \partial_y. \end{aligned} \tag{3.18}$$

The basis $(\boldsymbol{e}_r, \boldsymbol{e}_\theta)$ is orthonormal, but no longer derives from coordinates: it is an example of **noncoordinate basis**. It is also a noncommuting/nonholonomic basis: this can be interpreted by the fact that, since $\boldsymbol{e}_\theta$ has constant length but θ-lines are longer as we move further from the origin, moving by a unit displacement on a rectangle along the basis vectors does not form a closed loop.

3.1.2 Covectors, cotangent spaces and pullback

At each $\boldsymbol{p} \in M$, the vector space dual to $T_{\boldsymbol{p}}M$ is called the **cotangent space at $\boldsymbol{p}$**, and is noted $T^*_{\boldsymbol{p}}M$. Its elements are called **covectors at $\boldsymbol{p}$**, and are thus linear functionals $\omega : T_{\boldsymbol{p}}M \to \mathbb{R}$, taking tangent vectors as input. The union of all cotangent spaces at all points of a manifold forms the **cotangent bundle** of the manifold, noted T^*M. It is a property of dual spaces V and V^* of finite dimension that (i) they have the same dimension, hence the same number of basis vectors, and (ii) if e_j are basis vectors for V, the set of $\lambda^i \in V^*$ defined by $\lambda^i(e_j) \triangleq \delta^i_j$ is a basis for V^*. Applying this to the tangent space of basis vectors $\partial/\partial x^j$, the covectors $\lambda^i|_{\boldsymbol{p}}$ defined by:

$$\lambda^i|_{\boldsymbol{p}} \left(\frac{\partial}{\partial x^j}\right) \triangleq \delta^i_j \tag{3.19}$$

form a basis of the cotangent space. It follows that any covector can be written as:

$$\omega = \omega_i \lambda^i|_{\boldsymbol{p}} = \omega\left(\frac{\partial}{\partial x^i}\right)\lambda^i|_{\boldsymbol{p}}, \qquad \text{since} \qquad \omega\left(\frac{\partial}{\partial x^j}\right) = \omega_i \lambda^i|_{\boldsymbol{p}}\left(\frac{\partial}{\partial x^j}\right) = \omega_i \delta^i_j = \omega_j, \tag{3.20}$$

that is, applying ω to the j-th basis vectors selects the j-th component of ω. The action of a covector ω on a tangent vector $\boldsymbol{v}$ is thus:

$$\omega(\boldsymbol{v}) = \omega_i \lambda^i|_{\boldsymbol{p}}(\boldsymbol{v}) = \omega_i \lambda^i|_{\boldsymbol{p}}\left(v^j \frac{\partial}{\partial x^j}\right) = \omega_i v^j \lambda^i|_{\boldsymbol{p}}\left(\frac{\partial}{\partial x^j}\right) = \omega_i v^j \delta^i_j = \omega_i v^i \in \mathbb{R}. \tag{3.21}$$

The basis covectors $\lambda^i|_{\boldsymbol{p}}$ are for now not especially evocative of anything, but their interpretation will become clearer very shortly.

Differential of a function

The strength of differential geometry on manifolds is the ability to generalize concepts from calculus in a coordinates-free setting. In calculus, the gradient of a real-valued function f is a *vector* whose components are the partial derivatives of f. On manifolds, this is still true for the gradient in $\mathbb{R}^n$ in Cartesian coordinates, but not in general, e.g. for polar coordinates[2]:

$$\nabla f^{\text{cart}} = \frac{\partial f}{\partial x}\partial_x + \frac{\partial f}{\partial y}\partial_y, \qquad \nabla f^{\text{polar}} = \frac{\partial f}{\partial r}\partial_r + \frac{1}{r^2}\frac{\partial f}{\partial \theta}\partial_\theta \neq \frac{\partial f}{\partial r}\partial_r + \frac{\partial f}{\partial \theta}\partial_\theta. \tag{3.22}$$

The above shows that the polar gradient does *not* write as a vector whose components are simply the partial derivatives $(\partial_r f, \partial_\theta f)$. The motivation behind covectors is that partial derivatives are, in fact, always the components of a covector field called the **differential of the real-valued function** f, which matches the usual definition (3.8). To see this, we introduce the differential of a real-valued function f as the covector $df : T_{\boldsymbol{p}}M \to \mathbb{R}$ defined by:

$$df_{\boldsymbol{p}}(\boldsymbol{v}) = \boldsymbol{v}f, \tag{3.23}$$

thus simply the application of a tangent vector to the function f. As an element of the cotangent space, $df_{\boldsymbol{p}}$ can be written as the linear combination:

$$df_{\boldsymbol{p}} \overset{(3.20)}{=} df_{\boldsymbol{p}}\left(\frac{\partial}{\partial x^i}\right)\lambda^i|_{\boldsymbol{p}} \overset{(3.23)}{=} \left.\frac{\partial}{\partial x^i}\right|_{\boldsymbol{p}} f\,\lambda^i|_{\boldsymbol{p}} = \frac{\partial f}{\partial x^i}(\boldsymbol{p})\lambda^i|_{\boldsymbol{p}}. \tag{3.24}$$

In particular, given a chart (U, x^j), consider the coordinate functions $f = x^j : U \to \mathbb{R}$ which assign to $\boldsymbol{p} \in U \subseteq M$ its j-th coordinate. Their differentials write:

$$dx^j_{\boldsymbol{p}} = \frac{\partial x^j}{\partial x^i}(\boldsymbol{p})\lambda^i|_{\boldsymbol{p}} = \delta^j_i\lambda^i|_{\boldsymbol{p}} = \lambda^j|_{\boldsymbol{p}}. \tag{3.25}$$

In other words, the basis covectors λ^j are precisely the differentials dx^j from usual calculus, seen as linear forms acting on tangent vectors according to $dx^i_{\boldsymbol{p}}(\boldsymbol{v}) = v^i$. As shown in the next section, they are the building blocks to construct covariant tensors, such as Riemannian metrics, which is why they are presented here. The usual definition (3.8) can thus be interpreted as an equality between the covector fields df and dx^i. As their names suggest,

[2]Here, ∇f^{polar} is written with respect to the coordinate basis $(\partial_r, \partial_\theta)$ instead of the basis $(\boldsymbol{e}_r, \boldsymbol{e}_\theta)$, which changes the usual $1/r$ factor for $1/r^2$, since $\partial_\theta = r\boldsymbol{e}_\theta$.

the differentials of a smooth function $f : M \to \mathbb{R}$ introduced in the previous section and in this section are the same object, see Appendix A.2.1 for a proof and discussion.

Pullback of covector

We briefly mention an important tool associated to cotangent spaces: the **pullback** of covectors. As its name hints, the pullback *pulls* covectors from one manifold to another, and provides a way of comparing them. The pullback is the dual map[3] of the pushforward, that is, of the differential. Since the differential is a linear map $dF_{\boldsymbol{p}} : T_{\boldsymbol{p}}M \to T_{F(\boldsymbol{p})}M$, the pullback is a linear map $dF^*_{\boldsymbol{p}} : T^*_{F(\boldsymbol{p})}M \to T^*_{\boldsymbol{p}}M$ between cotangent spaces. When we will define unit elements in Chapter 4, we will mostly be interested in comparing Riemannian metrics (that is, covariant tensors) on different manifolds, rather than covectors. Thus, we postpone the detailed presentation of pullbacks to the next section.

3.1.3 Tensors

We are almost in the position to introduce tensors, which are multilinear applications between vector spaces. Let $V_1, \ldots, V_k$ and W denote real vector spaces. We say that a map $F : V_1 \times \ldots \times V_k \to W$ is **multilinear** if it is linear with respect to each of its arguments, and we note $L(V_1, \ldots, V_k; W)$ the set of all multilinear maps from $V_1 \times \ldots \times V_k$ to W. Letting now $\omega, \eta \in V^*$ denote covectors taking their argument in V, their **tensor product** is the real-valued function $\omega \otimes \eta : V \times V \to \mathbb{R}$ defined by:

$$\omega \otimes \eta(\boldsymbol{v}_1, \boldsymbol{v}_2) \triangleq \omega(\boldsymbol{v}_1)\eta(\boldsymbol{v}_2). \tag{3.26}$$

The tensor product is associative and preserves linearity: since ω, η are linear functionals, $\omega \otimes \eta$ is linear with respect to $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, hence $\omega \otimes \eta \in L(V, V; \mathbb{R})$. It follows that the tensor product of *an arbitrary number* of real-valued multilinear maps is also a multilinear map between the Cartesian product of the relevant vector spaces and $\mathbb{R}$. For instance, if F and G are multilinear maps of two vectors each and H is a multilinear map of three vectors, their tensor product is:

$$F \otimes G \otimes H(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_7) = F(\boldsymbol{v}_1, \boldsymbol{v}_2)G(\boldsymbol{v}_3, \boldsymbol{v}_4)H(\boldsymbol{v}_5, \boldsymbol{v}_6, \boldsymbol{v}_7). \tag{3.27}$$

An important property of the set $L(V_1, \ldots, V_k; \mathbb{R})$ is that its elements are linear combinations of tensor products of the form $e^{i_1}_{(1)} \otimes \ldots \otimes e^{i_k}_{(k)}$, where $1 \leq i_1 \leq \dim(V_1), \ldots, 1 \leq i_k \leq \dim(V_k)$,

[3] If $T : V \to W$ is a linear map between vector spaces, its dual map is the map $T^* : W^* \to V^*$ defined by $T^*(\omega) = \omega \circ T$, for $\omega \in W^*$.

and the $e^i_{(j)}$ are basis vectors of the dual spaces V_j^* (for the cotangent spaces of a manifold, those are the dx^i). For instance, a basis for the bilinear maps from $\mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ is given by the set:

$$e^1 \otimes e^1, \quad e^1 \otimes e^2, \quad e^2 \otimes e^1, \quad e^2 \otimes e^2, \tag{3.28}$$

where e^1 and e^2 are the standard basis of $(\mathbb{R}^2)^*$. These covectors are sometimes called the *contravariant basis* and represented by the *row* vectors $(1,0)$ and $(0,1)$: this is possible because there is an identification between vectors and covectors as elements of (finite dimensional) dual vector spaces, but one should keep in mind that the e^i are linear functionals. With these identifications, we have indeed $e^1(\boldsymbol{e}_1) = (1,0)(1,0)^T = 1$, $e^1(\boldsymbol{e}_2) = (1,0)(0,1)^T = 0$, etc., so that $e^i(\boldsymbol{e}_j) = \delta^i_j$, and the action of $e^i \otimes e^j$ on two column vectors $\boldsymbol{v}, \boldsymbol{w}$ can be thought of as the product $v^T A w$, where A is a matrix with 1 at the position (i,j) and zeros everywhere else.

A construction similar to (3.28) exists for vector spaces: the **tensor product of vector spaces** $V_1 \otimes \ldots \otimes V_k$. Its formal definition is more subtle and is not presented here, however, there is a canonical identification between the tensor product of dual spaces $V_1^* \otimes \ldots \otimes V_k^*$ and the set of multilinear maps $L(V_1, \ldots, V_k; \mathbb{R})$, thus allowing to use either notation to denote such maps.

We are now able to introduce the main concepts of this section. Let V be a finite-dimensional vector space, let k be a positive integer, and let $T^k(V^*)$ denote the k-fold tensor product $V^* \otimes \ldots \otimes V^*$. A **covariant k-tensor T on the vector space** V a is an element of $T^k(V^*)$, i.e., a real-valued multilinear function of k vectors:

$$T : \underbrace{V \times \ldots \times V}_{k \text{ times}} \to \mathbb{R}. \tag{3.29}$$

For example, 0-tensors are functions that take 0 argument and return a real number, hence simply real numbers. Covariant 1-tensors are linear maps from V to $\mathbb{R}$, i.e., linear functionals or covectors. In anisotropic mesh adaptation, we are mostly concerned with covariant 2-tensors, which are the bilinear forms (i.e., the inner products) on V. Although we will mostly focus on covariant 2-tensors, we also define **contravariant k-tensors** on V, which are elements of the k-fold tensor product space $T^k(V) = V \otimes \ldots \otimes V$. With the exception of 1-tensors, which are elements of V and thus simply vectors, the interpretation of a contravariant tensor is far less obvious than for covariant tensors. This is probably because, similarly to $V^* \otimes \ldots \otimes V^*$, the tensor product $V \otimes \ldots \otimes V$ can be identified with the set $L(V_1^*, \ldots, V_k^*; \mathbb{R})$ of multilinear maps taking *linear functionals* as arguments, instead of vectors. Covariant k-tensors are sometimes written as $(0,k)$-tensors, and contravariant l-tensors

as $(l,0)$-tensors. Tensors need not be only covariant or contravariant: a tensor that is k-covariant and l-contravariant is called a **mixed tensor**, also written as a (l,k)-tensor. The Riemann curvature tensor, introduced in Section 3.5, is for example a $(1,3)$ mixed tensor, which can also be written as a 4-covariant tensor. The integer k, l or $k+l$ is the **order** or **rank** of a covariant, contravariant or mixed tensor, respectively. A covariant tensor is **symmetric** if its output remains the same whenever two input vectors are interchanged[4]. The **symmetric product** of two covectors ω, η is used to write Riemannian metrics: it is the symmetric covariant 2-tensor defined by $\omega\eta \triangleq \frac{1}{2}(\omega \otimes \eta + \eta \otimes \omega)$, without any operators between the covectors.

Covariant tensors on the cotangent bundle

Going back to smooth manifolds, we define covariant tensors by setting $V = T_{\boldsymbol{p}}M$ in the previous definitions. A **covariant k-tensor $T_{\boldsymbol{p}}$ on a manifold M at $\boldsymbol{p}$** is an element of the tensor product of cotangent spaces $T^k(T^*_{\boldsymbol{p}}M) = T^*_{\boldsymbol{p}}M \otimes \ldots \otimes T^*_{\boldsymbol{p}}M$, i.e., a multilinear map:

$$T_{\boldsymbol{p}} : T_{\boldsymbol{p}}M \times \ldots \times T_{\boldsymbol{p}}M \to \mathbb{R}. \tag{3.30}$$

Since the $dx^i|_{\boldsymbol{p}}$ form a basis of the cotangent space $T^*_{\boldsymbol{p}}M$, we can write $T_{\boldsymbol{p}}$ as the linear combination of k^2 terms:

$$T_{\boldsymbol{p}} = T_{i_1 \ldots i_k}|_{\boldsymbol{p}}\, dx^{i_1}|_{\boldsymbol{p}} \otimes \ldots \otimes dx^{i_k}|_{\boldsymbol{p}}, \tag{3.31}$$

where $T_{i_1 \ldots i_k}|_{\boldsymbol{p}}$ is the $k \times k$ matrix of the components functions of $T_{\boldsymbol{p}}$. In this thesis, we will use both the notations $T_{i_1 \ldots i_k}|_{\boldsymbol{p}}$ and $[T_{\boldsymbol{p}}]$ to denote the matrix of components of $T_{\boldsymbol{p}}$. The former is used whenever index notation is required, and the latter is used to write relations between matrices, such as matrix products. Note that we do not use brackets to write matrices that are not associated with a tensor, such as Jacobian matrices. Dropping the reference to $\boldsymbol{p}$, a **smooth covariant k-tensor field** T writes:

$$T = T_{i_1 \ldots i_k}\, dx^{i_1} \otimes \ldots \otimes dx^{i_k}, \tag{3.32}$$

where both the coordinates x^i and the components $T_{i_1 \ldots i_k}$ are smooth functions. It is a smooth map that assigns a covariant k-tensor to each point of the manifold. Since 0-tensors are real numbers, smooth 0-tensor fields on M are simply smooth functions on M. A smooth tensor

[4]Symmetric k-tensors with $k > 2$ are thus fully symmetric, e.g., the coefficients of a symmetric covariant 4-tensor T satisfy $T_{ijkl} = T_{jikl} = T_{jilk} = T_{klij} = T_{ikjl} = \ldots$. In continuum mechanics, such a tensor has both the so-called *minor* and *major* symmetries.

field is symmetric if it assigns a symmetric tensor everywhere.

Pullback of covariant tensors

Lastly, we introduce the pullback of tensor fields, a construction that brings tensors back from the image of a smooth map to its domain. It is used extensively in the following chapters, to compare e.g. Riemannian metrics on different manifolds. Let M and N be smooth manifolds, let $F : M \to N$ be a smooth map, and let $T \in T^k(T^*N)$ be a covariant k-tensor field on N. The **pullback of** T **by** F is the covariant k-tensor field $F^*T \in T^k(T^*M)$ defined on M by the pointwise relation:

$$(F^*T)_{\boldsymbol{p}}(\boldsymbol{v}_1, \dots, \boldsymbol{v}_k) \triangleq T_{F(\boldsymbol{p})}(dF_{\boldsymbol{p}}(\boldsymbol{v}_1), \dots, dF_{\boldsymbol{p}}(\boldsymbol{v}_k)), \tag{3.33}$$

where the $\boldsymbol{v}_i \in T_{\boldsymbol{p}}M$ are tangent vectors at $\boldsymbol{p}$. In words, to evaluate the pullback tensor on M, we just need to push its arguments on the image manifold N with the differential $dF_{\boldsymbol{p}}$, then evaluate the original tensor defined on N. If y^i are coordinates for N, the pullback tensor writes ([46], Corollary 12.28):

$$F^*T = F^*(T_{i_1 \dots i_k}\, dy^{i_1} \otimes \dots \otimes dy^{i_k}) = (T_{i_1 \dots i_k} \circ F)\, d(y^{i_1} \circ F) \otimes \dots \otimes d(y^{i_k} \circ F). \tag{3.34}$$

This expression looks complicated, but the computation is actually quite simple. Both the tensor components $T_{i_1 \dots i_k}$ and the basis covectors dy^i are composed with F, hence given the map, one has to view the tensor on N written as a function of the coordinates of M. Alternatively, observing that $y^i \circ F = F^i$ is the i-th component of F, the differential writes in coordinates:

$$d(y^i \circ F) = dF^i = \frac{\partial F^i}{\partial x^j}\, dx^j, \tag{3.35}$$

where x^j are coordinates on M, the domain manifold. Thus, the pullback of a covariant 2-tensor writes in coordinates:

$$\begin{aligned} F^*T = (T_{ij} \circ F)\, d(y^i \circ F) \otimes d(y^i \circ F) &= (T_{ij} \circ F) \frac{\partial F^i}{\partial x^k}\, dx^k \otimes \frac{\partial F^j}{\partial x^\ell}\, dx^\ell \\ &= (T_{ij} \circ F) \frac{\partial F^i}{\partial x^k} \frac{\partial F^j}{\partial x^\ell}\, dx^k \otimes dx^\ell \\ &= (F^*T)_{k\ell}\, dx^k \otimes dx^\ell. \end{aligned} \tag{3.36}$$

The matrix of components $[F^*T]$ is given by:

$$(F^*T)_{k\ell} = (T_{ij} \circ F) \frac{\partial F^i}{\partial x^k} \frac{\partial F^j}{\partial x^\ell} = \left(\frac{\partial F^k}{\partial x^i} \right)^T (T_{ij} \circ F) \frac{\partial F^j}{\partial x^\ell}, \tag{3.37}$$

or, denoting by $J_F = \partial F^i/\partial x^j$ the Jacobian matrix of F, the matrix representations of T and its pullback are related[5] by:

$$[F^*T] = J_F^T\,[T \circ F]\,J_F = J_F^T\,[T]\,J_F. \tag{3.39}$$

Example: Pullback from Cartesian to polar coordinates ([46], Ex. 12.29)

To illustrate these notions, consider the right half-plane described in Cartesian coordinates by (x, y) with $x > 0$, and in polar coordinates by (r, θ) with $r > 0, |\theta| < \pi/2$. Let M denote the polar representation of the half-plane, and N the Cartesian representation, so that a map $F : M \to N$ is given by $(x, y) = F(r, \theta) = (r\cos\theta, r\sin\theta)$. We consider the tensor field $T = x^{-2}\,dy \otimes dy$ defined on N, with x^{-2} as the only component function. Viewed as a function of (r, θ), this tensor writes $F^*T = x^{-2}(r, \theta)\,dy(r, \theta) \otimes dy(r, \theta)$. Substituting the transformation, we have as in (3.36):

$$\begin{aligned} F^*T &= (r\cos\theta)^{-2}\,d(r\sin\theta) \otimes d(r\sin\theta) \\ &= (r\cos\theta)^{-2}\,(\sin\theta\,dr + r\cos\theta\,d\theta) \otimes (\sin\theta\,dr + r\cos\theta\,d\theta) \\ &= (r\cos\theta)^{-2}\left[\sin^2\theta\,dr \otimes dr + r\sin\theta\cos\theta\,(d\theta \otimes dr + dr \otimes d\theta) + r^2\cos^2\theta\,d\theta \otimes d\theta\right] \\ &= r^{-2}\tan^2\theta\,dr \otimes dr + r^{-1}\tan\theta\,(d\theta \otimes dr + dr \otimes d\theta) + d\theta \otimes d\theta, \end{aligned} \tag{3.40}$$

where we used the fact that the differentials act on functions according to Leibniz's rule as in usual calculus, i.e., $d(fg) = df\,g + f\,dg$. The pullback tensor is thus obtained by writing down the original tensor and replacing each occurrence of the image variables by the domain variables. In terms of components, the tensor T and its pullback are represented by the matrices:

$$[T] = \begin{pmatrix} 0 & 0 \\ 0 & x^{-2} \end{pmatrix}, \quad [F^*T] = \begin{pmatrix} r^{-2}\tan^2\theta & r^{-1}\tan\theta \\ r^{-1}\tan\theta & 1 \end{pmatrix}. \tag{3.41}$$

Letting r^j denote the coordinates (r, θ), the Jacobian matrix J_F is:

$$(J_F)_{ij} = \frac{\partial x^i}{\partial r^j} = \begin{pmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{pmatrix}, \tag{3.42}$$

[5] In continuum mechanics, for example, the second Piola-Kirchhoff stress tensor S, defined on the *reference configuration*, is the pullback of the Kirchhoff stress tensor τ, which itself is the scaled Cauchy stress tensor σ. Those tensors are contravariant 2-tensors, and the pullback writes with a slightly different, but analog, relation:

$$[S] = F^{-1}\,[\tau]\,F^{-T}, \tag{3.38}$$

where F is the Jacobian matrix of the map between configurations and is usually called the *gradient of the transformation*.

one checks that $[F^*T] = J_F^T[T \circ F]J_F = J_F^T[T(r,\theta)]J_F$.

3.2 Riemannian metrics

We now introduce the most important tensor field for anisotropic mesh adaptation:

Definition 3.1. *A **Riemannian metric** g on a smooth manifold M is a smooth symmetric covariant 2-tensor field on M that is positive-definite at each point. When M is equipped with a Riemannian metric, it is called a **Riemannian manifold**, and is noted (M, g).*

From Einstein's work in general relativity, it is customary in mathematics and physics to use the letter g (for *gravity*) to denote Riemannian metrics [58]. In the mesh adaptation literature, which we will follow in this thesis, it is usually the letter $\mathcal{M}$ that is used. Therefore, we write a Riemannian manifold as $(M, \mathcal{M})$. Although $\mathcal{M}$ is a tensor *field*, we sometimes abuse the notation and simply write $\mathcal{M}$ to denote the tensor at a point when the point is clear from context. We also sometimes write $\mathcal{M}(\boldsymbol{p})$ to denote $\mathcal{M}_{\boldsymbol{p}}$, the tensor field evaluated at $\boldsymbol{p}$, or, on the contrary, to put the emphasis on the field, but the meaning should always be clear. Moreover, we often drop the "Riemannian" denomination, and just call $\mathcal{M}$ a metric.

As a covariant 2-tensor, a Riemannian metric writes in local coordinates:

$$\mathcal{M} = \mathcal{M}_{ij}\, dx^i \otimes dx^j, \tag{3.43}$$

where $\mathcal{M}_{ij}$ is a symmetric and positive-definite matrix field. The symmetry of $\mathcal{M}_{ij}$, together with the symmetric product of covectors $dx^i dx^j \triangleq \frac{1}{2}(dx^i \otimes dx^j + dx^j \otimes dx^i)$ introduced earlier, allow to write the metric under the equivalent form:

$$\mathcal{M} = \mathcal{M}_{ij}\, dx^i dx^j. \tag{3.44}$$

As a multilinear map, the metric tensor is completely determined by its action on the n basis vectors of the tangent space. Since the differentials dx^i are dual to the basis tangent vectors $\partial/\partial x^i$, we can write:

$$\mathcal{M}\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right) = \left(\mathcal{M}_{k\ell}\, dx^k dx^\ell\right)\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right) = \mathcal{M}_{k\ell} \underbrace{dx^k\left(\frac{\partial}{\partial x^i}\right)}_{\delta_i^k} \underbrace{dx^\ell\left(\frac{\partial}{\partial x^j}\right)}_{\delta_j^\ell} = \mathcal{M}_{ij}. \tag{3.45}$$

At each point, the symmetric tensor $\mathcal{M}_{\boldsymbol{p}}$ takes two tangent vectors $\boldsymbol{u}, \boldsymbol{v} \in T_{\boldsymbol{p}}M$ and returns a positive real number, and thus defines an inner product, noted $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathcal{M}_{\boldsymbol{p}}} \triangleq \mathcal{M}_{\boldsymbol{p}}(\boldsymbol{u}, \boldsymbol{v})$. The

simplest Riemannian metric is the **Euclidean metric** $\overline{\mathcal{M}}$ of $\mathbb{R}^n$, whose component matrix is the identity:

$$\overline{\mathcal{M}} = \delta_{ij}\, dx^i dx^j = \sum_{i=1}^{n} dx^i dx^i \triangleq \sum_{i=1}^{n} (dx^i)^2. \tag{3.46}$$

It is associated to the canonical inner product[6] of $\mathbb{R}^n$, formally:

$$\begin{aligned} \langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\overline{\mathcal{M}}} = \overline{\mathcal{M}}(\boldsymbol{u}, \boldsymbol{v}) &= dx^1(\boldsymbol{u}) dx^1(\boldsymbol{v}) + \ldots + dx^n(\boldsymbol{u}) dx^n(\boldsymbol{v}) \\ &= u^1 v^1 + \ldots + u^n v^n \\ &= \boldsymbol{u} \cdot \boldsymbol{v}, \end{aligned} \tag{3.47}$$

where we used the fact that $dx^i(u) = u^i$, i.e., the differentials dx^i select the i-th component of their vector argument. In the following, this inner product is simply noted $\langle ., . \rangle$, without explicit reference to the Euclidean metric. For a general metric, the inner product can be computed from the matrix $[\mathcal{M}_{\boldsymbol{p}}]$:

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathcal{M}_{\boldsymbol{p}}} = \mathcal{M}_{\boldsymbol{p}}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T [\mathcal{M}_{\boldsymbol{p}}] \boldsymbol{v} = \langle \boldsymbol{u}, [\mathcal{M}_{\boldsymbol{p}}] \boldsymbol{v} \rangle. \tag{3.48}$$

3.2.1 Induced metric

Any smooth manifold M can be equipped with uncountably many Riemannian metrics, so there is not a natural way of choosing a metric for a given abstract manifold. If M is viewed as a submanifold of $(\mathbb{R}^m, \overline{\mathcal{M}})$ with $m \geq n$ (i.e., extrinsically), then the ambient Euclidean metric of $\mathbb{R}^m$ can be pulled back to define a Riemannian metric on M. This pullback metric is called the **induced metric** on M: it is simply the Euclidean metric restricted to pairs of vectors on the tangent spaces of M. It should be stressed that this is one way of equipping M with a metric, among many others. In differential geometry of surfaces, the induced metric is generally used to measure lengths and areas on surfaces in $\mathbb{R}^3$. Consider for instance the parameters $(u, v) \in U \subseteq \mathbb{R}^2$ and the surface M described by the graph of a function $f(u, v)$, characterized by the map $F : \mathbb{R}^2 \to \mathbb{R}^3 : (u, v) \mapsto (u, v, f(u, v)) = (x, y, z)$. This surface can be equipped with the induced metric, which we also denote $\overline{\mathcal{M}}$. It is given on M by $\overline{\mathcal{M}} = dx^2 + dy^2 + dz^2$, and its pullback to the parameter space is obtained by simply

[6] On $\mathbb{R}^n$, the tangent spaces $T_{\boldsymbol{p}}\mathbb{R}^n$ can all be identified to $\mathbb{R}^n$ itself after translation, thus tangent vectors can be identified to any point of $\mathbb{R}^n$. Since the Euclidean metric is constant, we can take the inner product of any two vectors even when they do not originate from the same point, by considering a copy of either fixed at the origin of the other, and thus living in the same tangent space.

substituting the dependence in u, v of x, y, z:

$$\begin{aligned} F^*\overline{\mathcal{M}} = F^*(dx^2 + dy^2 + dz^2) &= d(u)^2 + d(v)^2 + d(f(u,v))^2 \\ &= du^2 + dv^2 + (f_u du + f_v dv)^2 \\ &= du^2 + dv^2 + f_u^2 du^2 + f_v^2 dv^2 + 2 f_u f_v du\, dv \\ &= \left(1 + f_u^2\right) du^2 + 2 f_u f_v\, du\, dv + \left(1 + f_v^2\right) dv^2 \\ &= (F^*\overline{\mathcal{M}})_{ij}\, du^i du^j \end{aligned} \tag{3.49}$$

with $f_u = \partial_u f$, etc. The associated component matrix is:

$$[F^*\overline{\mathcal{M}}] = \begin{pmatrix} 1 + f_u^2 & f_u f_v \\ f_u f_v & 1 + f_v^2 \end{pmatrix}. \tag{3.50}$$

It is also known as the **first fundamental form**, noted $\mathbf{I}(u, v)$ and obtained as the inner product of the vectors $\partial f/\partial u^i \cdot \partial f/\partial u^j$. Using (3.39), the Jacobian matrix of F writes:

$$J_F = \frac{\partial F^i}{\partial x^j} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ f_u & f_v \end{pmatrix} \text{ so that } [F^*\overline{\mathcal{M}}] = \begin{pmatrix} 1 & 0 & f_u \\ 0 & 1 & f_v \end{pmatrix} \begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ f_u & f_v \end{pmatrix} = J_F^T J_F. \tag{3.51}$$

The induced metric is the natural size prescription of a surface, viewed on the parameter space (u, v). In Chapter 6, we discuss the use of this metric in the context of curvilinear mesh adaptation, following ideas from [9, 21, 23, 29, 41].

3.2.2 Riemannian manifolds for anisotropic mesh adaptation

In metric-based mesh adaptation, the Riemannian manifolds of interest are generally the reference space $(\widehat{\Omega} \subseteq \mathbb{R}^n, \overline{\mathcal{M}})$ endowed with the Euclidean metric, the physical space $(\Omega \subset \mathbb{R}^n, \mathcal{M})$ equipped with the Riemannian metric of interest, typically derived from the minimization of an error functional, and the physical space $(\Omega \subset \mathbb{R}^n, \overline{\mathcal{M}})$ equipped with the Euclidean metric. The mesh is created in the latter: it lives on a bounded subset of $\mathbb{R}^n$, which is equipped with the Euclidean metric by CAD softwares and PDE solvers. However, the manifold $(\Omega \subset \mathbb{R}^n, \mathcal{M})$ guides the mesh generation. In 2D, the parameter space $U = (u, v)$ introduced above describes, together with the induced metric $F^*\overline{\mathcal{M}}$, the surface M in $\mathbb{R}^3$, and a mesh created on U and adapted to $F^*\overline{\mathcal{M}}$ reflects the shape and orientation of the surface M. This has been explored in recent work [30, 34, 35, 59] to produce anisotropic and curvilinear meshes that are *isometrically embedded* in $\mathbb{R}^n$ with $n \geq 3$, then projected back to the physical space. The idea and hypothesis behind these methods is that "a uniform isotropic mesh in a higher-dimensional space corresponds to an anisotropic mesh in a

lower-dimensional space" [35]. In Caplan et al. [30, 59], U is equipped with the Riemannian metric $[\mathcal{M}] = J_F^T J_F$ with $F : \mathbb{R}^2 \to \mathbb{R}^3 : \boldsymbol{x} \mapsto (\boldsymbol{x}, f(\boldsymbol{x}))$, which is precisely the metric induced by the graph of $f(\boldsymbol{x})$. At the end of this chapter, we discuss an open question with which we are concerned, namely, instead of the graph of f, which is intrinsic to the function and is not related to the interpolation error a priori, (when) is there a function g such that the metric induced by the graph of g yields metric-conforming triangulations which control and/or equidistribute the interpolation error?

3.2.3 Lengths and distances

The main utility of Riemannian metrics is to measure lengths on manifolds. On the tangent space, a Riemannian metric determines the length of tangent vectors and the angle formed by two vectors. On the manifold, two points are not connected by a vector, but by a curve. We define the length of a curve, and how curves induce a distance function between two points.

Definition 3.2. *Let $\boldsymbol{u}, \boldsymbol{v} \in T_{\boldsymbol{p}}M$ be two tangent vectors at $\boldsymbol{p}$.*

- *The **norm** or **length** of $\boldsymbol{u}$ is defined by:*

$$\|\boldsymbol{u}\|_{\mathcal{M}_{\boldsymbol{p}}} = \sqrt{\langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\mathcal{M}_{\boldsymbol{p}}}} = \sqrt{\boldsymbol{u}^T [\mathcal{M}_{\boldsymbol{p}}] \boldsymbol{u}}. \tag{3.52}$$

- *The **angle** formed by $\boldsymbol{u}$ and $\boldsymbol{v}$ is the unique real number $\theta \in [0, \pi]$ such that:*

$$\cos\theta = \frac{\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathcal{M}_{\boldsymbol{p}}}}{\|\boldsymbol{u}\|_{\mathcal{M}_{\boldsymbol{p}}} \|\boldsymbol{v}\|_{\mathcal{M}_{\boldsymbol{p}}}} \tag{3.53}$$

A metric prescribes the length of curves as follows:

Definition 3.3. *The **length** of a curve γ parameterized by $t \in [a, b]$ is:*

$$\ell_{\mathcal{M}}(\gamma) = \int_a^b \|\gamma'(t)\|_{\mathcal{M}} \, dt = \int_a^b \sqrt{\langle \gamma'(t), \gamma'(t) \rangle_{\mathcal{M}}} \, dt. \tag{3.54}$$

Similarly, the **energy** of a curve is defined by:

$$E_{\mathcal{M}}(\gamma) = \frac{1}{2} \int_a^b \|\gamma'(t)\|_{\mathcal{M}}^2 \, dt = \frac{1}{2} \int_a^b \langle \gamma'(t), \gamma'(t) \rangle_{\mathcal{M}} \, dt. \tag{3.55}$$

An important property of the length of a curve is that it is independent of its parameterization. This is not true for the energy, however. Evaluating lengths is not trivial, as in

general the metric varies along the curve. In the meshing process, lengths are computed often to evaluate quality indicators, decide if an edge is unit, or decide if two vertices are too close to one another, see Chapters 4 and 7. To allow for fast computations, lengths are computed with a quadrature with a low number of Gauss points, typically 3 for the edges of $\mathcal{P}^2$ triangles. In practice, the metric is known only at the discrete mesh vertices so that metric interpolation, discussed in Section 3.6.1, is required. When performing anisotropic adaptation with linear meshes, all curves are straight lines, and a line between $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ can be parameterized by $\gamma \equiv \boldsymbol{x}(t) = \boldsymbol{x}_0 + t(\boldsymbol{x}_1 - \boldsymbol{x}_0)$ for $t \in [0, 1]$. Its length with respect to the metric is:

$$\begin{aligned} \ell_{\mathcal{M}}(\gamma) = \int_0^1 \|\boldsymbol{x}_1 - \boldsymbol{x}_0\|_{\mathcal{M}} \, dt &= \int_0^1 \sqrt{\langle \boldsymbol{x}_1 - \boldsymbol{x}_0, \boldsymbol{x}_1 - \boldsymbol{x}_0 \rangle_{\mathcal{M}}} \, dt \\ &= \int_0^1 \sqrt{(\boldsymbol{x}_1 - \boldsymbol{x}_0)^T [\mathcal{M}] \left(\boldsymbol{x}_0 + t(\boldsymbol{x}_1 - \boldsymbol{x}_0)\right) (\boldsymbol{x}_1 - \boldsymbol{x}_0)} \, dt \end{aligned} \tag{3.56}$$

When the manifold is $\mathbb{R}^n$ and the metric is constant, simplifications arise. For general curves, the metric in the inner product of (3.54) needs to be evaluated only once, and for straight lines, the integral reduces to the single computation:

$$\ell_{\mathcal{M}}(\gamma) = \int_0^1 \sqrt{(\boldsymbol{x}_1 - \boldsymbol{x}_0)^T [\mathcal{M}] (\boldsymbol{x}_1 - \boldsymbol{x}_0)} \, dt = \sqrt{(\boldsymbol{x}_1 - \boldsymbol{x}_0)^T [\mathcal{M}] (\boldsymbol{x}_1 - \boldsymbol{x}_0)}. \tag{3.57}$$

If M is connected, which is assumed from now on, any two points can be connected by a curve and the metric induces a distance function between them:

Definition 3.4. *The **distance** between two points $\boldsymbol{p}$ and $\boldsymbol{q}$ is the infimum of the lengths of all curve segments joining $\boldsymbol{p}$ and $\boldsymbol{q}$, i.e.:*

$$d_{\mathcal{M}}(\boldsymbol{p}, \boldsymbol{q}) = \inf_{\gamma} \ell_{\mathcal{M}}(\gamma), \tag{3.58}$$

where $\gamma(a) = \boldsymbol{p}$ and $\gamma(b) = \boldsymbol{q}$.

On a Riemannian manifold, a curve of minimum length between two points is called a *geodesic*: we postpone the presentation of geodesics and the discussion of the distance function to Section 3.4, because it requires first defining the coordinate-free acceleration of curves. There is in general no closed-form expression for the geodesic between two points, so the definition above does not provide an explicit way of computing distances on arbitrary manifolds. On $(\mathbb{R}^n, \mathcal{M})$ with constant metric, however, as for the usual Euclidean space, straight lines are the shortest path between two points. As the tangent spaces of $\mathbb{R}^n$ can be identified to $\mathbb{R}^n$ itself, vectors can be added or subtracted, and the distance function is given explicitly by

the usual expression:

$$d_{\mathcal{M}}(\boldsymbol{p},\boldsymbol{q}) = \|\boldsymbol{p}-\boldsymbol{q}\|_{\mathcal{M}}. \tag{3.59}$$

3.2.4 Eigenvalues and eigenvectors

Riemannian metrics are used for anisotropic mesh adaptation because they encode the target sizes $h_i > 0$ and directions $\boldsymbol{v}_i$ of the adaptation process. Since the matrix representing the metric is SPD, its eigenvectors are orthogonal and it can always be diagonalized by:

$$[\mathcal{M}] = P\Lambda P^T = \begin{pmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2\end{pmatrix}\begin{pmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{pmatrix}\begin{pmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\end{pmatrix} = \begin{pmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2\end{pmatrix}\begin{pmatrix}h_1^{-2} & \\ & h_2^{-2}\end{pmatrix}\begin{pmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\end{pmatrix} \tag{3.60}$$

with P orthogonal. In linear algebra, the matrices $[\mathcal{M}]$ and Λ are called *similar*, as there exists $Q = P^T = P^{-1}$ such that both matrices are related by $[\mathcal{M}] = Q^{-1}\Lambda Q$. Similar matrices share several invariants, including their characteristic polynomial, hence their eigenvalues and the quantities derived from them, such as the determinant and the trace. The components of covariant 2-tensors, such as Riemannian metrics, do *not* transform as similar matrices, however, as shown below.

Transformation of a covariant 2-tensor

Consider on M the change from coordinates x^i to new coordinates $\widetilde{x}^j$, with Jacobian matrix $J_{ij} = \partial\widetilde{x}^i/\partial x^j$. From the chain rule, the coordinate basis vectors transform as:

$$\frac{\partial}{\partial\widetilde{x}^j} = \frac{\partial x^1}{\partial\widetilde{x}^j}\frac{\partial}{\partial x^1} + \ldots + \frac{\partial x^n}{\partial\widetilde{x}^j}\frac{\partial}{\partial x^n} = \frac{\partial x^i}{\partial\widetilde{x}^j}\frac{\partial}{\partial x^i} = J^{-1}_{ij}\frac{\partial}{\partial x^i}. \tag{3.61}$$

The components of $\mathcal{M}$ in the new coordinates are determined by the new basis:

$$\begin{aligned}\widetilde{\mathcal{M}}_{ij} = \mathcal{M}\left(\frac{\partial}{\partial\widetilde{x}^i},\frac{\partial}{\partial\widetilde{x}^j}\right) &= \mathcal{M}\left(\frac{\partial x^k}{\partial\widetilde{x}^i}\frac{\partial}{\partial x^k},\frac{\partial x^\ell}{\partial\widetilde{x}^j}\frac{\partial}{\partial x^\ell}\right) = \frac{\partial x^k}{\partial\widetilde{x}^i}\frac{\partial x^\ell}{\partial\widetilde{x}^j}\mathcal{M}\left(\frac{\partial}{\partial x^k},\frac{\partial}{\partial x^\ell}\right)\\ &= \frac{\partial x^k}{\partial\widetilde{x}^i}\frac{\partial x^\ell}{\partial\widetilde{x}^j}\,\mathcal{M}_{k\ell} = \left(\frac{\partial x^i}{\partial\widetilde{x}^k}\right)^T\mathcal{M}_{k\ell}\,\frac{\partial x^\ell}{\partial\widetilde{x}^j}.\end{aligned} \tag{3.62}$$

In matrix form, this writes $[\widetilde{\mathcal{M}}] = J^{-T}[\mathcal{M}]J^{-1}$. Note that from the expression of the pullback tensor (3.39), the components in the old coordinates are indeed recovered by $[\mathcal{M}] = [F^*\widetilde{\mathcal{M}}] = J^T[\widetilde{\mathcal{M}}]J = J^TJ^{-T}[\mathcal{M}]J^{-1}J = [\mathcal{M}]$, where $F : M \to M : x^i \mapsto \widetilde{x}^j$ is the change of coordinate map. Thus, the components matrix in both coordinate systems obey a relation of the form:

$$[\widetilde{\mathcal{M}}] = Q^T[\mathcal{M}]Q, \tag{3.63}$$

where $Q^T \neq Q^{-1}$ in general: $[\widetilde{\mathcal{M}}]$ and $[\mathcal{M}]$ are not similar, but *congruent* matrices. Their invariants are characterized by Sylvester's law of inertia, which states that congruent matrices only share the same numbers of positive, zero or negative eigenvalues. In particular, the characteristic polynomials of both matrices, and thus their eigenvalues, trace and determinant, differ. For instance, the determinant of $[\widetilde{\mathcal{M}}]$ is $\det[\widetilde{\mathcal{M}}] = (\det J^{-1})^2 \det[\mathcal{M}] \neq \det[\mathcal{M}]$. The number of eigenvalues of each sign in the matrix of a Riemannian metric is called its **signature**. Since Riemannian metrics are positive-definite, this only ensures that all their eigenvalues are positive. In other words, the eigenvalues and eigenvectors of the matrix representing the metric bear no intrinsic significance, since they are not preserved by change of coordinates. As we discuss below however, setting the components of the tensor in one basis completely determines the tensor, and the eigenvalues and eigenvectors can be used to this end. Before that, we show that, while the eigenvalues do not characterize a covariant tensor, they characterize *mixed* tensors.

Mixed tensors

Eigenvectors and eigenvalues are, in fact, only well-defined for mixed $(1,1)$-tensors, which map vectors to vectors[7]. Such a tensor writes:

$$T = T^i{}_j \, \frac{\partial}{\partial x^i} \otimes dx^j, \tag{3.64}$$

and takes as input both a covector and a tangent vector. Unlike coordinate vectors, the differentials $d\tilde{x}^i$ transform according to the usual calculus relation for differentials:

$$d\tilde{x}^i = \frac{\partial \tilde{x}^i}{\partial x^k} \, dx^k, \tag{3.65}$$

Then T transforms as:

$$\begin{aligned} \widetilde{T}^i{}_j = T\left(d\tilde{x}^i, \frac{\partial}{\partial \tilde{x}^j}\right) = T\left(\frac{\partial \tilde{x}^i}{\partial x^k} \, dx^k, \frac{\partial x^\ell}{\partial \tilde{x}^j} \frac{\partial}{\partial x^\ell}\right) = \frac{\partial \tilde{x}^i}{\partial x^k} \frac{\partial x^\ell}{\partial \tilde{x}^j} T\left(dx^k, \frac{\partial}{\partial x^\ell}\right) \\ = \frac{\partial \tilde{x}^i}{\partial x^k} \frac{\partial x^\ell}{\partial \tilde{x}^j} \, T^k{}_\ell = \frac{\partial \tilde{x}^i}{\partial x^k} \, T^k{}_\ell \, \frac{\partial x^\ell}{\partial \tilde{x}^j}, \end{aligned} \tag{3.66}$$

[7]Mixed $(1,1)$-tensors are real-valued multilinear maps $T : (\omega, v) \mapsto T(\omega, v)$ that take a covector ω and a tangent vector v as input and return a real number. However, they can be also seen as maps $T : v \mapsto T(., v)$ from vectors to vectors, by observing that $T(., v) : \omega \mapsto T(\omega, v)$ is indeed a vector, because it takes a covector and returns a real number. This amounts to *contracting* only one index of T. Note that to properly apply the vector to a covector, we actually identify the vector v with an element of the double dual $(T_{\boldsymbol{p}}M)^{**}$ to the tangent space, which is the tangent space itself, as it is a vector space of finite dimension.

In matrix form, this writes $[\widetilde{T}] = J[T]J^{-1}$, which describes *similar* matrices. It follows that the matrices $[\widetilde{T}]$ and $[T]$ share the same eigenvalues and eigenvectors.

Although eigenvalues are defined only for mixed tensors, a Riemannian metric is completely determined by its matrix in a given system of coordinates, so it *does* make sense to view the eigenvalues of this matrix as the principal sizes when it is associated with Cartesian coordinates. As a simple example, consider the metric $\mathcal{M}$ on $\mathbb{R}^2$ described in Cartesian coordinates $x^i = (x, y)$ by the diagonal matrix $\operatorname{diag}(\lambda_1, \lambda_2)$, and let $F : \mathbb{R}^2 \to \mathbb{R}^2$ be the change to polar coordinates $r^j = (r, \theta)$. Note that we consider here the change from Cartesian to polar, unlike the example in Section 3.1.3 from polar to Cartesian, thus the inverse Jacobian matrix is the Jacobian from Section 3.1.3:

$$J^{-1}_{ij} = \frac{\partial x^i}{\partial r^j} = \begin{pmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{pmatrix}, \tag{3.67}$$

In Cartesian coordinates, the eigenvalues $[\mathcal{M}]$ are λ_1, λ_2, and when $\lambda_1 \neq \lambda_2$, its eigenvectors are the basis vectors $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$. In polar coordinates, the matrix for $\mathcal{M}$ is:

$$\begin{aligned} [\mathcal{M}]_{\text{polar}} = J^{-T}[\mathcal{M}]_{\text{cart}}J^{-1} &= \begin{pmatrix} \cos\theta & \sin\theta \\ -r\sin\theta & r\cos\theta \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{pmatrix} \\ &= \begin{pmatrix} \lambda_1\cos^2\theta + \lambda_2\sin^2\theta & r\cos\theta\sin\theta\,(\lambda_1 - \lambda_2) \\ r\cos\theta\sin\theta\,(\lambda_1 - \lambda_2) & r^2(\lambda_2\cos^2\theta + \lambda_1\sin^2\theta) \end{pmatrix}. \end{aligned} \tag{3.68}$$

whose eigenvalues differ from those of $[\mathcal{M}]_{\text{cart}}$[8]. However, they represent the same metric tensor and measure the same lengths. For instance, consider in polar coordinates the radial straight curve $\gamma(t) = (c + r_0 t, \theta_0)$ for $c > 0$ and $t \in [0, 1]$. It is a *geodesic*, hence locally the shortest path between its endpoints, see Section 3.4, and its velocity vector is $\gamma'(t) = (r_0, 0)$. Its length is:

$$\ell_{\mathcal{M}}(\gamma) = \int_0^1 \sqrt{\langle \gamma'(t), \gamma'(t) \rangle_{\mathcal{M}}}\, dt = \sqrt{r_0^2(\lambda_1\cos^2\theta_0 + \lambda_2\sin^2\theta_0)}. \tag{3.70}$$

In particular, if $\theta_0 = 0$, the curve is aligned with $\boldsymbol{e}_x$ and its length is $r_0\sqrt{\lambda_1} = r_0/h_1$, thus a segment of Euclidean length $r_0 = h_1$ is of unit length with respect to $\mathcal{M}$, as prescribed

[8] In particular, in the case $\lambda_1 = \lambda_2 = 1$ of the Euclidean metric:

$$[\mathcal{M}]_{\text{cart}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{and} \quad [\mathcal{M}]_{\text{polar}} = \begin{pmatrix} 1 & 0 \\ 0 & r^2 \end{pmatrix}, \tag{3.69}$$

that is, the line elements write $ds^2 = dx^2 + dy^2$ and $ds^2 = dr^2 + r^2\, d\theta^2$, respectively.

by the first eigenvalue of $[\mathcal{M}]$ in Cartesian coordinates. One checks that the lengths prescribed by either $[\mathcal{M}]_{\text{polar}}$ or $[\mathcal{M}]_{\text{cart}}$ match for the other directions. In other words, even though the eigenvalues and eigenvectors of the metric are not invariant, we can use them to uniquely determine the tensor in all coordinates systems. On $\mathbb{R}^n$ with Cartesian coordinates in particular, the eigenvalues are associated to lengths and the eigenvectors are associated to directions, and thus have a clear physical interpretation.

Matrix operations

From the decomposition $[\mathcal{M}] = P\Lambda P^T$, we define several standard matrix operations that are used throughout this thesis: matrix power, absolute value, exponential and logarithm. For $\alpha \in \mathbb{R}$, we denote by $[\mathcal{M}]^\alpha$ the symmetric matrix obtained by elevating to the power α the eigenvalues of $[\mathcal{M}]$:

$$[\mathcal{M}]^\alpha \triangleq P\Lambda^\alpha P^T = P\,\mathrm{diag}(\lambda_1^\alpha, \lambda_2^\alpha)\,P^T. \tag{3.71}$$

The absolute value of $[\mathcal{M}]$, the exponential and the logarithm are defined similarly by:

$$\begin{aligned} |[\mathcal{M}]| &\triangleq P\,\mathrm{diag}(|\lambda_1|, |\lambda_2|)\,P^T, \\ \exp[\mathcal{M}] &\triangleq P\,\mathrm{diag}(\exp\lambda_1, \exp\lambda_2)\,P^T, \\ \ln[\mathcal{M}] &\triangleq P\,\mathrm{diag}(\ln\lambda_1, \ln\lambda_2)\,P^T. \end{aligned} \tag{3.72}$$

As it only makes sense to apply these operations on the matrix representing a Riemannian metric in a given coordinate system and not on the tensor itself, we can omit the brackets and write $\mathcal{M}^\alpha, |\mathcal{M}|, \exp\mathcal{M}$ and $\ln\mathcal{M}$ without confusion. Similarly, the determinant of the matrix representing $\mathcal{M}$ is simply noted $\det\mathcal{M}$.

3.2.5 Riemannian isometries

Next, we introduce an important family of maps between Riemannian manifolds.

Definition 3.5. *Let $(M, \mathcal{M})$ and $(\widetilde{M}, \widetilde{\mathcal{M}})$ be two Riemannian manifolds. A diffeomorphism $F : M \to \widetilde{M}$ is called a **(Riemannian) isometry** if $F^*\widetilde{\mathcal{M}} = \mathcal{M}$.*

In other words, an isometry is a diffeomorphism that preserves the metric, so that for all $\boldsymbol{p} \in M$ and for any two tangent vectors $u, v \in T_{\boldsymbol{p}}M$:

$$\mathcal{M}_{\boldsymbol{p}}(u, v) = \widetilde{\mathcal{M}}_{F(\boldsymbol{p})}(dF_{\boldsymbol{p}}u, dF_{\boldsymbol{p}}v). \tag{3.73}$$

Isometries are the *isomorphisms* between Riemannian manifolds, that is, the structure-preserving maps whose inverse is also structure-preserving. The term *structure-preserving* depends on the context: an isomorphism is a bijective map with additional properties depending on the sets put in relation. For instance, an isomorphism between sets without additional structure is just a bijection. If the sets are given a topology, isomorphisms are the continuous bijections with continuous inverse, that is, *homeomorphisms*, which preserve the topology. In particular, isomorphisms between topological manifolds are homeomorphisms. Smooth manifolds are given an additional smooth structure, which is preserved by smooth (i.e., C^∞) maps. Isomorphisms between smooth manifolds are thus *diffeomorphisms*, i.e., smooth homeomorphisms with smooth inverse. Lastly, Riemannian manifolds are smooth manifolds equipped with a metric tensor field: the corresponding isomorphisms are the metric-preserving diffeomorphisms, hence the Riemannian isometries defined above. In mathematics, Riemannian geometry is precisely the branch of geometry that studies the properties of manifolds that are invariant under isometries, a very important of which is *curvature*, presented in Section 3.5. The isomorphisms between each class of manifolds are summarized in (3.74) below, where X is a topology on the set M and $\mathcal{A}$ is the largest smooth atlas for M, i.e., its smooth structure. Smooth and Riemannian manifolds inherit the additional structure of each preceding class of manifolds. For instance, Riemannian manifolds are in particular both smooth and topological manifolds, thus $(M, \mathcal{M})$ is a short notation for $(M, X, \mathcal{A}, \mathcal{M})$.

$$\begin{aligned} \text{Topological manifolds:} \quad & (M, X) && \xrightarrow{\text{Homeo}} && (\widetilde{M}, \widetilde{X}) \\ \text{Smooth manifolds:} \quad & (M, \mathcal{A}) && \xrightarrow{\text{Diffeo}} && (\widetilde{M}, \widetilde{\mathcal{A}}) \\ \text{Riemannian manifolds:} \quad & (M, \mathcal{M}) && \xrightarrow{\text{Isometry}} && (\widetilde{M}, \widetilde{\mathcal{M}}) \end{aligned} \tag{3.74}$$

Two objects that are isomorphic are indistinguishable from each other, in the sense of the structures of interest. In fact, (3.74) alone motivates the use of Riemannian isometries in Chapter 4 to define unit elements and unit meshes. Indeed, the goal of anisotropic mesh adaptation is to generate triangulations that are essentially indistinguishable from a regular tiling of the plane: viewed as triangulations on Riemannian manifolds, they need to be isomorphic, thus isometric.

The composition of isometries is also an isometry, as well as the inverse of an isometry. Riemannian isometries are, in particular, *metric* isometries, i.e., isometries between metric spaces, and thus preserve distances and angles ([56], Prop. 2.51). Moreover, isometries send geodesic curves on M to geodesic curves on $\widetilde{M}$, see Section 3.4. Riemannian isometries play

a key role in Chapter 4, where unit elements of the continuous mesh framework are defined as isometric transformations of a reference equilateral triangle $K_\triangle$. This is motivated by the discussion above, and by the observation that the current definition of unit elements is based on transformations $F_{\triangle\to K}$ whose Jacobian matrices are of the form $J_{\triangle\to K} = \mathcal{M}^{-1/2}R$, with R a rotation matrix. The following proposition shows that this is a necessary condition for $F_{\triangle\to K}$ to be an isometry:

Proposition 3.1. *Let $(M, \mathcal{M})$ be a Riemannian manifold. If $F : (\mathbb{R}^2, \overline{\mathcal{M}}) \to (M, \mathcal{M})$ is a Riemannian isometry, then its differential in coordinates is the Jacobian matrix:*

$$J_F = \frac{\partial F^i}{\partial y^j} = (\mathcal{M}_{ij}^{-1/2} \circ F)O, \tag{3.75}$$

for some orthogonal matrix O, where y^j are coordinates for $\mathbb{R}^2$.

Proof. If F is an isometry, then $F^*\mathcal{M} = \overline{\mathcal{M}}$, i.e., it pulls back the metric $\mathcal{M}$ to the Euclidean metric on $\mathbb{R}^2$. The matrix representing $\overline{\mathcal{M}}$ is the identity matrix, and using (3.39) to compute the pullback matrix, we write:

$$[F^*\mathcal{M}] = J_F^T[\mathcal{M}]J_F = [\overline{\mathcal{M}}] = I. \tag{3.76}$$

This is satisfied when the Jacobian is of the form $J_F = \mathcal{M}^{-1/2}O = P\Lambda^{-1/2}P^TO$, with O any orthogonal matrix (i.e., $O^TO = OO^T = I$). Indeed, since P is also orthogonal:

$$J_F^T[\mathcal{M}]J_F = O^TP\Lambda^{-1/2}P^TP\Lambda P^TP\Lambda^{-1/2}P^TO = I, \tag{3.77}$$

thus $J_F = \mathcal{M}^{-1/2}O = (\mathcal{M}^{-1/2} \circ F)O$ as claimed. □

Orthogonal matrices include rotations, reflections and their compositions, however reflections have negative determinant, so the transformation $F_{\triangle\to K}$ inverts the triangle K, making it invalid. Thus, we only consider the orthogonal matrices $R \in \mathrm{SO}_2$ associated to rotations, so that the Jacobian matrices of interest are of the form:

$$J_F = \mathcal{M}^{-1/2}R, \tag{3.78}$$

and from there:

$$\det J_F = (\det \mathcal{M})^{-1/2}, \tag{3.79}$$

which is used hereafter to write down integrals.

The relation (3.78) is a necessary, but not sufficient condition for F to be an isometry. To be an isometry, F must be a diffeomorphism, thus infinitely differentiable and with commuting mixed derivatives of all orders. However, if J_F is set by a Riemannian metric, then the higher-order derivatives of F commute only if the metric has certain symmetries. For instance, suppose $J_F = \mathcal{M}^{-1/2}$ for simplicity. Then the Hessian matrix of F in coordinates writes:

$$\frac{\partial^2 F^k}{\partial y^i \partial y^j} = \frac{\partial}{\partial y^j}(\mathcal{M}^{-1/2}_{ki} \circ F) = \frac{\partial \mathcal{M}^{-1/2}_{ki}}{\partial x^m} \underbrace{\frac{\partial F^m}{\partial y^j}}_{=(J_F)_{mj}} = \frac{\partial \mathcal{M}^{-1/2}_{ki}}{\partial x^m} \mathcal{M}^{-1/2}_{mj}, \tag{3.80}$$

and is symmetric iff:

$$\frac{\partial \mathcal{M}^{-1/2}_{ki}}{\partial x^m} \mathcal{M}^{-1/2}_{mj} = \frac{\partial \mathcal{M}^{-1/2}_{kj}}{\partial x^m} \mathcal{M}^{-1/2}_{mi} \quad \text{for all} \;\; i, j, k. \tag{3.81}$$

This symmetry condition on $\mathcal{M}$ was also obtained by Rochery [33] when investigating high-order unit elements. Since F is smooth, one should check all derivatives, possibly by induction. For third derivatives, we have:

$$\frac{\partial^3 F^l}{\partial y^i \partial y^j \partial y^k} = \left(\partial_{mn} \mathcal{M}^{-1/2}_{li} \mathcal{M}^{-1/2}_{mj} + \partial_m \mathcal{M}^{-1/2}_{li} \partial_n \mathcal{M}^{-1/2}_{mj} \right) \mathcal{M}^{-1/2}_{nk} \tag{3.82}$$

and symmetry must be enforced for all components l and all permutations of i, j, k, and so on for higher derivatives. We did not obtain the set of constraints on $\mathcal{M}$ to ensure that F is smooth, but we conjecture that it is equivalent to require that the *scalar curvature* of $\mathcal{M}$ vanishes, see Section 3.5, because then the manifold is isometric to the Euclidean space (i.e., *flat*, see also Section 3.5).

Example: Isometries of the Euclidean plane

The isometries of Euclidean geometry, translations, rotations, reflections and compositions thereof, are Riemannian isometries, sending (geodesic) straight lines to straight lines. For instance, consider two subsets $U, V \subset \mathbb{R}^2$ each equipped with the Euclidean metric $\overline{\mathcal{M}}$, represented in coordinates by the identity matrix $I = \delta_{ij}$.

- Translations by a vector $\boldsymbol{p}$ in $\mathbb{R}^2$ are characterized by $F : U \to V : \boldsymbol{u} \mapsto \boldsymbol{v} = \boldsymbol{u} + \boldsymbol{p}$. The Jacobian matrix is the identity, thus the pullback relation $[F^* \overline{\mathcal{M}}] = J_F^T [\overline{\mathcal{M}}] J_F$ given by (3.39), which reads here $[F^* \mathcal{M}] = III = I$, holds.
- Rotations are characterized by $F : U \to V : \boldsymbol{u} \mapsto \boldsymbol{v} = R\boldsymbol{u}$. The Jacobian matrix is the

orthogonal matrix $R \in \mathrm{SO}_2$, hence $[F^*\overline{\mathcal{M}}] = R^T I R = R^T R = I$ and (3.39) holds.

- Reflections are characterized by $F : U \to V : \boldsymbol{u} \mapsto \boldsymbol{v} = A\boldsymbol{u}$ where $\det A = -1$ and the eigenvalues of A are $(-1, 1)$ or $(1, -1)$. For instance, with $A = \mathrm{diag}(-1, 1)$, we have $[F^*\overline{\mathcal{M}}] = A^T I A = A^T A = I$ and (3.39) holds again. In this work, we do not consider reflections because they have a negative determinant and yield invalid triangles.

3.2.6 Integration and area

So far, we have only discussed how Riemannian metrics are used to measure lengths and distances. We conclude this section by showing how they affect the computation of integrals, and thus areas and volumes. The natural objects to integrate on manifolds are the antisymmetric (or alternating) covariant tensor fields, also called *differential forms*. By convention, 1-tensors are both symmetric and antisymmetric, thus, as antisymmetric covariant 1-tensors, covectors are also called *differential 1-forms*. The only family of differential forms we present is the product of a smooth function f and the **Riemannian volume form** $dV_{\mathcal{M}}$, which encodes how a metric $\mathcal{M}$ modifies the computation of elementary volumes. As in usual calculus, integrating the volume form on a compact Riemannian manifold $(M, \mathcal{M})$ yields its **volume**:

$$\mathrm{Vol}(M) = \int_M dV_{\mathcal{M}}. \tag{3.83}$$

The key property we use to compute integrals is the following, and involves once again the pullback: let N and M be two smooth n-manifolds and let $F : N \to M$ be a map that preserves the *orientation*. If ω is a differential n-form on M, then:

$$\int_{M=F(N)} \omega = \int_N F^*\omega, \tag{3.84}$$

where $F^*\omega$ is the pullback of the form ω, for which we refer to [46] for a complete definition. For practical computations, we take $N \subset \mathbb{R}^n$ as a domain of integration (i.e., a parameterization of M). The **integral of f over** M is given by:

$$\int_M f \, dV_{\mathcal{M}} = \int_N F^*(f \, dV_{\mathcal{M}}) = \int_N \left(f\sqrt{\det \mathcal{M}} \circ F\right) (\det J_F) \, dx^1 \dots dx^n. \tag{3.85}$$

In the finite element methods, when integrating a function f over a triangle $K \subset M$, the domain of integration N is usually the reference simplex $\widehat{K} \subset \mathbb{R}^2$ with coordinates $\boldsymbol{\xi} = (\xi, \eta)$, which are a parametrization for $K = F(\boldsymbol{\xi})$. Then, the **integral of f over the triangle** K

is computed by the change of variable:

$$\int_K f\, dV_{\mathcal{M}} = \int_{\widehat{K}} (f\sqrt{\det \mathcal{M}} \circ F)\, (\det J_F)\, d\boldsymbol{\xi}. \tag{3.86}$$

Alternatively, if the domain of integration is an equilateral triangle $K_\triangle \subset \mathbb{R}^2$ with coordinates $\boldsymbol{y} = (y^1, y^2)$ and with $F : K_\triangle \to K$, we write:

$$\int_K f\, dV_{\mathcal{M}} = \int_{K_\triangle} (f\sqrt{\det \mathcal{M}} \circ F)\, (\det J_F)\, d\boldsymbol{y}. \tag{3.87}$$

Moreover, if F is a Riemannian isometry in either (3.86) or (3.87), then the integral is conserved in the sense that e.g.:

$$\int_K f\, dV_{\mathcal{M}} = \int_{K_\triangle} (f \circ F)\, d\boldsymbol{y}, \tag{3.88}$$

because $\det J_F = (\sqrt{\det \mathcal{M}})^{-1}$. These relations describe how integrals are computed in the rest of this thesis.

Example: Area of a triangle on a manifold

To illustrate, we compute the **area** of a (possibly curved) triangle K on a manifold $(M, \mathcal{M})$, setting $f = 1$. Using (3.86) and the reference triangle $\widehat{K}$ as domain of integration:

$$|K| \triangleq |K|_{\mathcal{M}} \triangleq \mathrm{Vol}(K) = \int_K 1\, dV_{\mathcal{M}} = \int_{\widehat{K}} \sqrt{\det \mathcal{M}}\, (\det J_F)\, d\boldsymbol{\xi}. \tag{3.89}$$

If the metric is constant, it is shown in Chapter 4 that K is a linear triangle, and we have:

$$|K| = \sqrt{\det \mathcal{M}} \int_{\widehat{K}} \det J_F\, d\boldsymbol{\xi} = \sqrt{\det \mathcal{M}}\, |K|_{\overline{\mathcal{M}}}, \tag{3.90}$$

where $|K|_{\overline{\mathcal{M}}}$ is the area of K with respect to the Euclidean metric. If $F : \widehat{K} \to K$ is a Riemannian isometry, then the area is conserved. For instance, with $K = F(\widehat{K}) = \mathcal{M}^{-1/2}\widehat{K}$:

$$|K| = \int_{\widehat{K}} \sqrt{\det \mathcal{M}}\, (\det \mathcal{M}^{-1/2})\, d\boldsymbol{\xi} = \int_{\widehat{K}} 1\, d\boldsymbol{\xi} = |\widehat{K}|_{\overline{\mathcal{M}}}. \tag{3.91}$$

Other isometries of interest are obtained by first composing with isometries in the plane, i.e., rotations and translations. Translations do not affect the Jacobian matrix, and rotations have unit determinant, hence the result above holds. In the following, we mostly consider isometries between $K_\triangle$ and K, instead of $\widehat{K}$ and K: in other words, K is an equilateral

triangle on the manifold M. The area of those triangles matches the area of $K_\triangle$:

$$|K| = |K_\triangle|_{\overline{\mathcal{M}}}. \tag{3.92}$$

3.3 Connections and the Levi-Civita connection

Geodesics, the topic of Section 3.4, are curves without acceleration and are the curves of interest when creating curved elements on Riemannian manifolds, whereas curvature, the topic of Section 3.5, characterizes which manifolds are isometric by measuring how second derivatives fail to commute. Both of these require giving a precise sense to derivatives on manifold, particularly derivatives of vector fields. The motivating problem concerns the definition of a coordinate-free acceleration for curves on a manifold M. From Section 3.1.1, we know that the velocity of a curve γ at any t is a well-defined tangent vector $\gamma'(t) \in T_{\gamma(t)}M$, whose representation in any system of coordinates x^i reads:

$$\gamma'(t) = \frac{d\gamma^i}{dt}\frac{\partial}{\partial x^i}\bigg|_{\gamma(t)}. \tag{3.93}$$

The following simple example from Lee shows that acceleration cannot be written in a similar fashion by simply substituting the first derivatives of γ^i for second derivatives. Consider the unit circle in $\mathbb{R}^2$, described by $(\cos t, \sin t)$ in Cartesian coordinates and by $(1, t)$ in polar. Its velocity writes in each coordinate system:

$$\begin{aligned}\gamma'(t) &= x'(t)\partial_x|_{\gamma(t)} + y'(t)\partial_y|_{\gamma(t)} = -\sin t\,\partial_x|_{\gamma(t)} + \cos t\,\partial_y|_{\gamma(t)} \\ &= r'(t)\partial_r|_{\gamma(t)} + \theta'(t)\partial_\theta|_{\gamma(t)} = \partial_\theta|_{\gamma(t)}.\end{aligned} \tag{3.94}$$

For the acceleration, however, the results in Cartesian and polar coordinates differ:

$$\gamma''(t) = -\cos t\,\partial_x|_{\gamma(t)} - \sin t\,\partial_y|_{\gamma(t)} \neq \theta''(t)\partial_\theta|_{\gamma(t)} = 0. \tag{3.95}$$

This is because the acceleration is the limit of the difference between vectors $\gamma'(t+h) \in T_{\gamma(t+h)}M$ and $\gamma'(t) \in T_{\gamma(t)}M$. These vectors belong to different vector spaces, and we cannot, in general, add or subtract them. A **connection** on a smooth manifold M is a tool introduced precisely to differentiate vector fields along curves, by providing a way to *connect* vectors living in neighbouring tangent spaces. More precisely, a connection is a map

$$\nabla : \mathfrak{X}(M) \times \mathfrak{X}(M) \to \mathfrak{X}(M) : (X, Y) \mapsto \nabla_X Y \tag{3.96}$$

that satisfies the following properties:

1. *Linearity over $C^\infty(M)$ in X:* for $f, g \in C^\infty(M)$ and $X_1, X_2 \in \mathfrak{X}(M)$,
$$\nabla_{fX_1+gX_2} Y = f\nabla_{X_1} Y + g\nabla_{X_2} Y, \tag{3.97}$$
2. *Linearity over $\mathbb{R}$ in Y:* for $a, b \in \mathbb{R}$ and $Y_1, Y_2 \in \mathfrak{X}(M)$,
$$\nabla_X(aY_1 + bY_2) = a\nabla_X Y_1 + b\nabla_X Y_2, \tag{3.98}$$
3. *Leibniz's rule for Y:* for $f \in C^\infty(M)$,
$$\nabla_X(fY) = (Xf)Y + f\nabla_X Y \tag{3.99}$$

The vector field $\nabla_X Y$ is called the **covariant derivative of Y in the direction X**. Since $\nabla_X Y$ is a vector field, it can be expanded in terms of its components in a local frame (E_i) (see Section 3.1.1):
$$\nabla_{E_i} E_j = \Gamma_{ij}^k E_k. \tag{3.100}$$

The n^3 smooth real-valued functions Γ_{ij}^k are called the **connection coefficients of ∇**, with the name *Christoffel symbols* usually reserved for the coefficients of the *Levi-Civita connection*, introduced hereafter. If $\Gamma_{ij}^k = \Gamma_{ji}^k$, then we say that the connection ∇ is **symmetric**. The Γ_{ij}^k completely determine the connection, as it writes in a local frame:
$$\nabla_X Y = \left(X(Y^k) + X^i Y^j \Gamma_{ij}^k\right) E_k = \left(X^i \frac{\partial Y^k}{\partial x^i} + X^i Y^j \Gamma_{ij}^k\right) E_k. \tag{3.101}$$

In other words, the covariant derivative is the usual directional derivative, supplemented by a term involving the Γ_{ij}^k reflecting the curvilinear coordinates. The result is a coordinate-free derivative operator for vector fields. In $\mathbb{R}^n$, the basis vectors fields $E_i = \boldsymbol{e}_i$ are orthogonal, so by (3.100) the connection coefficients vanish everywhere. It follows that the simplest example of connection is the usual *nabla* operator in $\mathbb{R}^n$, noted $\overline{\nabla}$ and called the **Euclidean connection**. The directional derivative in $\mathbb{R}^n$ of a vector field Y along X is defined by:
$$\overline{\nabla}_X Y \triangleq \left(X^i \frac{\partial Y^j}{\partial x^i}\right) E_j, \tag{3.102}$$

and indeed matches the coordinate description of the covariant derivative of Y along X. The components of this vector field are the result of the usual directional derivative obtained e.g. in continuum mechanics by contracting the *gradient tensor of Y* with the vector X:
$$\nabla Y \cdot X = \frac{\partial Y^j}{\partial x^i} X^i \tag{3.103}$$

Covariant derivative of tensors of arbitrary order

Although it was introduced to differentiate vector fields, a connection actually induces a coordinate-free derivative operator for tensors of all orders on M. For instance, the covariant derivative for 0-th order tensors, i.e., smooth scalar functions, writes $\nabla_X f = Xf = X^i \partial_i f$ and is the usual directional derivative for a scalar field. A connection ∇ is *not* a (mixed) tensor, because it is not linear in Y over smooth function, i.e., $\nabla_X(fY) \neq f\nabla_X Y$ for a smooth function f, but instead it satisfies Leibniz's rule. A covariant derivative $\nabla_X Y$, however, *is* a tensor field, and covariant derivatives in all directions can be encoded in a tensor field of one (covariant) order higher than its second argument. This tensor is noted ∇Y and is called the **total covariant derivative** of Y. It is the tensor that was informally introduced in (3.103) above. For instance, the total covariant derivative of a vector field Y, i.e., a $(1,0)$-tensor field, is a $(1,1)$-tensor field which writes in coordinates:

$$\nabla Y = Y^i_{\;;j} \, \frac{\partial}{\partial x^i} \otimes dx^j, \tag{3.104}$$

where the semicolon indicates covariant differentiation with respect to the j-th coordinate:

$$Y^i_{\;;j} \triangleq \frac{\partial Y^i}{\partial x^j} + Y^k \Gamma^i_{jk}. \tag{3.105}$$

The Levi-Civita connection

So far, we have presented connections and their properties without the need of a Riemannian metric. It so happens that the Euclidean connection satisfies Leibniz's product rule with respect to the Euclidean metric. That is, for three vector fields $X, Y, Z \in \mathfrak{X}(\mathbb{R}^n)$, the inner product $\langle Y, Z \rangle$ is a smooth real-valued function on $\mathbb{R}^n$ and, remarkably, one shows that:

$$\overline{\nabla}_X \langle Y, Z \rangle = X \langle Y, Z \rangle = \left\langle \overline{\nabla}_X Y, Z \right\rangle + \left\langle Y, \overline{\nabla}_X Z \right\rangle. \tag{3.106}$$

More generally, on a Riemannian manifold $(M, \mathcal{M})$, we say that a connection ∇ is **compatible with the metric** $\mathcal{M}$ if it satisfies the product rule above for all vector fields $X, Y, Z \in \mathfrak{X}(M)$ with respect to the inner product $\mathcal{M}$. A connection compatible with the metric is also called a **metric connection**. Proposition 5.5 in [56] shows that a connection ∇ is compatible with $\mathcal{M}$ iff the total covariant derivative of the metric vanishes, i.e., $\nabla \mathcal{M} = 0$.

In coordinates, this writes:

$$\begin{aligned} &\frac{\partial \mathcal{M}_{ij}}{\partial x^k} - \Gamma^\ell_{ki}\mathcal{M}_{\ell j} - \Gamma^\ell_{kj}\mathcal{M}_{\ell i} = 0 \\ &\qquad\qquad\qquad\qquad \Big\downarrow \\ &\Gamma^\ell_{ki}\mathcal{M}_{\ell j} + \Gamma^\ell_{kj}\mathcal{M}_{\ell i} = \frac{\partial \mathcal{M}_{ij}}{\partial x^k}. \end{aligned} \tag{3.107}$$

Since the coefficients Γ^k_{ij} completely determine a connection, any choice of n^3 smooth functions determines a connection on a Riemannian manifold. Although there is no natural way of equipping a smooth manifold with a metric, an important result is that there is a canonical connection for a given Riemannian manifold. The *fundamental theorem of Riemannian geometry* states that for any Riemannian manifold $(M, \mathcal{M})$, there is a unique connection ∇ that is both symmetric and compatible with the metric $\mathcal{M}$. This connection is known as the **Levi-Civita connection of $\mathcal{M}$**, and its coefficients Γ^k_{ij} are called the **Christoffel symbols of $\mathcal{M}$**. Their expression in coordinates is derived from (3.107):

$$\Gamma^k_{ij} = \frac{1}{2}\mathcal{M}^{-1}_{k\ell}\left(\frac{\partial \mathcal{M}_{j\ell}}{\partial x^i} + \frac{\partial \mathcal{M}_{i\ell}}{\partial x^j} - \frac{\partial \mathcal{M}_{ij}}{\partial x^\ell}\right) = \frac{1}{2}\mathcal{M}^{-1}_{k\ell}\left(\partial_i \mathcal{M}_{j\ell} + \partial_j \mathcal{M}_{i\ell} - \partial_\ell \mathcal{M}_{ij}\right). \tag{3.108}$$

It is clear from this expression that they are symmetric (in i, j).

3.4 Geodesics

Geodesics are the generalization of straight lines to Riemannian manifolds. In the Euclidean space, straight lines are characterized by two properties: (i) they are curves with zero acceleration, and (ii) they are the unique shortest path between their extremities. Geodesics are defined by generalizing the first property: in fact, connections were introduced precisely to this end, as a way of defining a coordinate-free acceleration for curves. As generalized straight lines, geodesics play a crucial role in the generation of curvilinear meshes: in Chapter 7, our proposed meshing algorithm creates triangles with geodesic edges. In the following, we examine the properties of geodesics as curves with zero acceleration. In particular, the length-minimizing property of straight lines applies locally to geodesics. Lastly, we show how geodesics can be computed in practice.

Let $I \subset \mathbb{R}$ be an interval, $\gamma : I \to M$ be a smooth curve on a smooth manifold M and let $\dot{\gamma}(t) \triangleq \gamma'(t)$. Some technicalities aside, if V is a vector field along the curve γ in the sense that $V(t) \in T_{\gamma(t)}M$ for all $t \in I$, then the **covariant derivative of V along the curve γ**

is defined by:

$$D_t V(t) \triangleq \nabla_{\dot{\gamma}(t)} V(t) \overset{(3.101)}{=} \left(\dot{V}^k(t) + \dot{\gamma}^i(t) V^j(t) \Gamma_{ij}^k(\gamma(t)) \right) E_k. \tag{3.109}$$

Then, the **acceleration of** γ is given by $D_t\dot{\gamma}$, and a curve is geodesic w.r.t. a connection ∇ if it has zero acceleration, i.e.:

$$D_t\dot{\gamma} = \nabla_{\dot{\gamma}(t)}\dot{\gamma}(t) = 0. \tag{3.110}$$

For a given metric $\mathcal{M}$, the only connection with respect to which we will consider the geodesics is its Levi-Civita connection. Since it is unique, we can talk about *geodesics with respect to* $\mathcal{M}$ without confusion. An important application of covariant derivatives along curves concerns vector fields which do not vary along these curves, that is, vector fields satisfying $D_t V(t) = 0$. Such fields are said to be *parallel* along γ: they are the closest approximation to constant vector fields on manifolds. The associated construction is called *parallel transport*: it allows to extend tangent vectors to neighbouring tangent spaces in such a way that their components do not vary. In fact, parallel transport is precisely how connections *connect* nearby vectors, as mentioned at the beginning of Section 3.3, and how vectors are parallel transported completely defines the connection. From (3.110), geodesics can also be defined as curves whose velocity vector is parallel to itself, or parallel transported along the curve. Denoting the component functions of the curve by $\gamma(t) = (x^1(t), \dots, x^n(t))$, (3.110) writes in coordinates:

$$\frac{d^2x^k}{dt^2} + \frac{dx^i}{dt}\frac{dx^j}{dt}\Gamma_{ij}^k(x(t)) = 0. \tag{3.111}$$

This ODE is known as the **geodesic equation**. Since we are only interested in geodesic curves for the Levi-Civita connection, the Γ_{ij}^k are the Christoffel symbols given by (3.108). When the metric, and thus the Christoffel symbols, is known, the geodesic equation is a system of n nonlinear ODEs in the component functions $x^i(t)$, where n is the dimension of the manifold. As such, ODE theory guarantees the existence of a unique geodesic curve satisfying the initial conditions $\gamma(t_0) = \boldsymbol{p} \in M$ and $\gamma'(t_0) = v_{\boldsymbol{p}} \in T_{\boldsymbol{p}}M$. When generating curved meshes, the initial velocity is not known, rather, we wish to connect two vertices $\boldsymbol{p}$ and $\boldsymbol{q}$ with a geodesic curve segment. We thus distinguish geodesics with prescribed initial point and velocity from geodesics between prescribed endpoints. The former always exist, at least locally, and the latter exist when M is a complete and connected manifold, which is assumed in this work. Computing the Christoffel symbols in practice also requires interpolating the derivatives of the metric components, which is done using the spectral decomposition proposed by Aparicio-Estrems et al. [26], described in Section 3.6.1.

Among all curves between two prescribed endpoints, the length-minimizing ones are geodesic curves. The converse is true only locally, that is, geodesics are *locally* length-minimizing curves. For instance, the geodesics on a sphere are the great circles, and two points can be connected by either the longer or the shorter segment of a great circle, but only the shorter segment minimizes the length between the two points. When generating curved meshes, the endpoints are known and we approach geodesics by polynomial curves. Since the shortest path between the endpoints is necessarily a geodesic, minimizing the length over the parameters of the polynomial curves yields an approximation of the geodesic between these points.

Both the geodesic character of a curve and parallel transport are notions that are preserved by Riemannian isometries. In particular ([56], Cor. 5.14):

Proposition 3.2. *Let $(M, \mathcal{M})$ and $(\widetilde{M}, \widetilde{\mathcal{M}})$ be two Riemannian manifolds and let $F : M \to \widetilde{M}$ be an isometry. If γ is a geodesic in M, then $F \circ \gamma$ is a geodesic in $\widetilde{M}$.*

The exponential map

To each pair $(\boldsymbol{p}, \boldsymbol{v})$ of a starting point and tangent vector corresponds a unique geodesic $\gamma_{\boldsymbol{v}}(t)$. It follows that there is an identification between tangent vectors of $T_{\boldsymbol{p}}M$ and neighbouring points of $\boldsymbol{p}$ on the manifold, called the **exponential map**, and defined by:

$$\exp_{\boldsymbol{p}} : T_{\boldsymbol{p}}M \to M : \boldsymbol{v} \mapsto \exp_{\boldsymbol{p}}(\boldsymbol{v}) = \gamma_{\boldsymbol{v}}(1), \tag{3.112}$$

whenever $\gamma_v(1)$ is defined. The exponential map takes a tangent vector and assigns it to a point on the manifold by following $\gamma_v(t)$. It is used in the next section to give a geometric interpretation of varying metrics on manifolds, by examining their (curved) unit sphere.

Computing geodesics

Determining the geodesics either with initial data or between prescribed endpoints is not a trivial task, and there is no closed-form expression for either curve for arbitrary Riemannian metrics. In practice, the geodesics are determined as follows:

- Geodesics with prescribed initial data are obtained by numerically integrating (3.111). Thus, we solve the initial-value problem:

$$\frac{d^2x^k}{dt^2} + \frac{dx^i}{dt}\frac{dx^j}{dt}\Gamma^k_{ij}(x(t)) = 0, \qquad x(t_0) = \boldsymbol{p}, \qquad \frac{dx}{dt}(t_0) = v_{\boldsymbol{p}}. \tag{3.113}$$

The second-order ODE can be split into two first-order ODEs, which are integrated using e.g. an explicit 4th order Runge-Kutta method:

$$\frac{dx^k}{dt} = v^k, \qquad \frac{dv^k}{dt} = -\Gamma^k_{ij} v^i v^j. \tag{3.114}$$

- The geodesics between two points that we consider in this thesis are either the true geodesics, with unknown parameterization $\gamma(t)$, or geodesic polynomial curves $\mathcal{C}(t)$, that is, $\mathcal{P}^k$ curve segments of minimal length between these points. In our applications, we only consider $\mathcal{P}^2$ approximation of geodesics.

  The true geodesics between two points are used for visualization, for instance, to illustrate that geodesic edges for the Euclidean metric are mapped to geodesics by an isometry. In this case, we solve the boundary value problem:

$$\frac{d^2x^k}{dt^2} + \frac{dx^i}{dt}\frac{dx^j}{dt}\Gamma^k_{ij}(x(t)) = 0, \qquad x(t_0) = \boldsymbol{p}, \qquad x(t_1) = \boldsymbol{q}. \tag{3.115}$$

  In this work, we simply use the shooting method to determine the initial velocity such that $\gamma(t)$ passes through $\boldsymbol{q}$. In practice, we use a bisection on both the angle of the initial velocity vector and the integration interval so that the curve passes at a prescribed tolerance from $\boldsymbol{q}$. The angle is first determined using a signed distance function from the endpoint to the curve. Then, the interval is determined similarly with a second signed distance function. Other methods, such as the curve-shortening flow [60], can be applied to compute geodesics numerically.

  On the other hand, polynomial approximations $\mathcal{C}(t)$ are obtained by minimizing the Riemannian length between the prescribed endpoints over all $\mathcal{P}^k$ curve segments:

$$\mathcal{C}(t) = \arg\min \ell_{\mathcal{M}}(\boldsymbol{p}, \boldsymbol{q}). \tag{3.116}$$

  In this thesis, the minimization is performed using the unconstrained minimization routines of the Ceres-Solver library [54] (quasi-Newton L-BFGS method).

3.4.1 Geometric representation of a metric

With a way of computing distances on Riemannian manifolds, we can give a graphic representation of a metric tensor. First, we define the **tangential ball** of radius r at $\boldsymbol{p} \in M$ by the Euclidean ball centered at zero in the tangent space defined by:

$$B_r(0) \triangleq \left\{ \boldsymbol{v} \in T_{\boldsymbol{p}}M \mid \|\boldsymbol{v}\|_{\mathcal{M}_{\boldsymbol{p}}} \leq r \right\}, \tag{3.117}$$

and the associated **tangential sphere** by $\partial B_r(0)$. The reference to the point $\boldsymbol{p}$ is not written explicitly, but appears in the definition instead. Then, the **geodesic ball** at $\boldsymbol{p}$ associated to $B_r(0)$ is its image on the manifold by the exponential map:

$$\mathcal{B}_r(\boldsymbol{p}) \triangleq \exp_{\boldsymbol{p}}\left(B_r(0)\right). \tag{3.118}$$

The r in subscript is not without significance: in fact, a geodesic ball $\mathcal{B}_r(\boldsymbol{p})$ or sphere $\partial\mathcal{B}_r(\boldsymbol{p})$ is also a metric ball or sphere of the same radius ([56], Cor. 6.13), which justifies the names of geodesic ball or sphere *of radius* r at $\boldsymbol{p}$. In particular, the **unit geodesic ball** at $\boldsymbol{p}$, or simply the **unit ball** at $\boldsymbol{p}$, is $\mathcal{B}_1(\boldsymbol{p})$, and the associated **unit sphere** is $\partial\mathcal{B}_1(\boldsymbol{p})$. The unit ball at $\boldsymbol{p}$ also writes:

$$\mathcal{B}_1(\boldsymbol{p}) = \{\boldsymbol{q} \in M \mid d_{\mathcal{M}}(\boldsymbol{p},\boldsymbol{q}) \leq 1\}. \tag{3.119}$$

The tangential unit ball $B_1(0)$ illustrates how the metric locally distorts the lengths, and is usually used as its geometric representation when working with linear elements. Consider for instance $U \subseteq \mathbb{R}^2$ and the manifolds $(U, \overline{\mathcal{M}})$ and $(V = F(U), \mathcal{M})$ with constant metric $\mathcal{M}$, with F a linear isometry given by $F : U \to V : \boldsymbol{p} \mapsto \mathcal{M}^{-1/2}\boldsymbol{p}$. Tangent vectors are transformed by the differential dF, which is a linear isometry between tangent spaces:

$$dF_{\boldsymbol{p}} : T_{\boldsymbol{p}}U \to T_{F(\boldsymbol{p})}V : \boldsymbol{u} \mapsto \mathcal{M}^{-1/2}\boldsymbol{u}. \tag{3.120}$$

Indeed, the norm of the tangent vectors is preserved:

$$\|\mathcal{M}^{-1/2}\boldsymbol{u}\|_{\mathcal{M}} = \sqrt{\boldsymbol{u}^T\mathcal{M}^{-1/2}[\mathcal{M}]\mathcal{M}^{-1/2}\boldsymbol{u}} = \sqrt{\boldsymbol{u}^T\boldsymbol{u}} = \|\boldsymbol{u}\|. \tag{3.121}$$

Thus, the unit tangential ball $B_1^U(0)$ at $\boldsymbol{p}$ on U is mapped to $B_1^V(0)$ at $F(\boldsymbol{p})$ on V by $\mathcal{M}^{-1/2}$ and $B_1^V(0)$ also writes:

$$B_1^V(0) = \left\{\mathcal{M}^{-1/2}\boldsymbol{u} \mid \boldsymbol{u} \in B_1^U(0)\right\}. \tag{3.122}$$

This is illustrated in Fig. 3.7. Using the diagonalized form $[\mathcal{M}] = P\Lambda P^T$, the points on $\partial B_1^V(0)$ satisfy:

$$\|\boldsymbol{u}\|_{\mathcal{M}}^2 = \boldsymbol{u}^T[\mathcal{M}]\boldsymbol{u} = \sum_{i=1}^{2}\langle\boldsymbol{u},\boldsymbol{v}_i\rangle^2\lambda_i = \sum_{i=1}^{2}\left(\frac{\langle\boldsymbol{u},\boldsymbol{v}_i\rangle}{h_i}\right)^2 = 1. \tag{3.123}$$

The last equality is the equation of an ellipse (in $\mathbb{R}^3$, an ellipsoid) of axes $\boldsymbol{v}_i$, the eigenvectors of $[\mathcal{M}]$, and lengths $h_i = \lambda_i^{-1/2}$. The goal of mesh adaptation is to create elements with unit edges, which are, in a sense, inscribed inside these ellipses. The unit balls $B_1(0)$ on the

tangent spaces are thus a convenient way of representing the local shape and size of the mesh elements after adaptation.

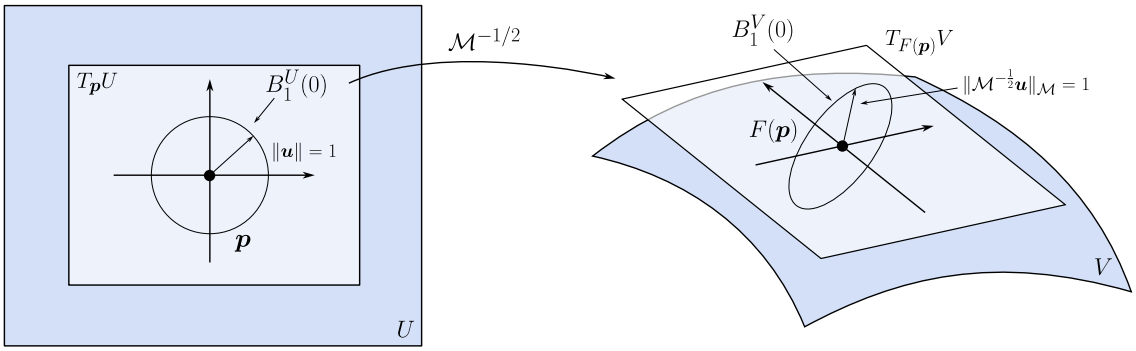


Figure 3.7 Unit tangential ball $B_1^U(0)$ and its image on the tangent space of V.

To generate curved elements, variations of the metric are accounted for as we travel on the manifold and the metric is represented by its geodesic unit sphere. Consider now $(M, \mathcal{M})$ with arbitrary metric. The discussion above applies to the tangent space at each $\boldsymbol{p} \in M$ by replacing $\mathcal{M}$ by $\mathcal{M}_{\boldsymbol{p}}$. From (3.119), a point $\boldsymbol{q}$ lies on the unit sphere at $\boldsymbol{p}$ if $d_{\mathcal{M}}(\boldsymbol{p}, \boldsymbol{q}) = 1$, that is, if it is the extremity of a geodesic of unit length starting at $\boldsymbol{p}$. Moreover, the unit ball is defined *solely* from geodesics. In particular, leaving $\boldsymbol{p}$ along any other parameterization, such as a straight line $\boldsymbol{p} + t(\boldsymbol{q} - \boldsymbol{p})$ of unit length in $\mathcal{M}$, like we did in the tangent space, does not yield a point on the unit sphere of $\boldsymbol{p}$. In fact, this construction is only possible on $\mathbb{R}^n$ because it is also a vector space, i.e., any point between $\boldsymbol{p}$ and $\boldsymbol{q}$ lies on the manifold and the addition of points on the manifold is well-defined, which is not the case in general, such as on a sphere. This is illustrated in Fig. 3.8 for the metric defined hereafter, where both the unit sphere $\partial\mathcal{B}_1(\boldsymbol{p})$ and the locus of the points obtained by leaving $\boldsymbol{p}$ along a straight line of unit length are shown.

Example: Unit geodesic spheres

To illustrate the difference between the unit balls on the tangent spaces and on the manifold, we consider the Riemannian metric obtained by modifying the principal sizes of the induced metric on the graph of the function $f(x, y) = x^2 + y^2$. The induced metric pulled back on the parameter space $(x, y) \subseteq \mathbb{R}^2$, is:

$$[\mathcal{M}] = \begin{pmatrix} 1 + f_x & f_x f_y \\ f_x f_y & 1 + f_y \end{pmatrix} = \frac{1}{f_x^2 + f_y^2} \begin{pmatrix} -f_y & f_x \\ f_x & f_y \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 + f_x^2 + f_y^2 \end{pmatrix} \begin{pmatrix} -f_y & f_x \\ f_x & f_y \end{pmatrix}. \tag{3.124}$$

and the manifold of interest if $(\mathbb{R}^2, \mathcal{M})$. The eigenvectors of $[\mathcal{M}]$ are, in reversed order, the

unit directions of the gradient and its orthogonal vector, i.e., the direction of the isoline of f at (x, y). We set the eigenvalues to $\lambda_i = h_i^{-2} > 0$ to control the size along those two principal directions. For this choice of f, the modified metric reads:

$$[\mathcal{M}] = \frac{1}{x^2+y^2}\begin{pmatrix} -y & x \\ x & y \end{pmatrix}\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}\begin{pmatrix} -y & x \\ x & y \end{pmatrix} = \frac{1}{x^2+y^2}\begin{pmatrix} \lambda_1 y^2 + \lambda_2 x^2 & xy(\lambda_2 - \lambda_1) \\ xy(\lambda_2 - \lambda_1) & \lambda_1 x^2 + \lambda_2 y^2 \end{pmatrix}, \tag{3.125}$$

This metric is well-defined on $\mathbb{R}^2\backslash\{0\}$, and the center of the unit spheres is set to $\boldsymbol{p} = (0.5, 0)$. The tangential unit sphere at $\boldsymbol{p}$ is computed from either (3.117) or (3.122). It is the ellipse whose axes are aligned with the gradient and direction of isoline of f at $\boldsymbol{p}$, Fig. 3.8 in dashed line, and whose sizes are h_i. The geodesic unit sphere at $\boldsymbol{p}$, in black in Fig. 3.8, is computed from (3.119), and requires computing the geodesics of unit length originating from $\boldsymbol{p}$. The "unit sphere" formed by the straight lines of unit length originating from $\boldsymbol{p}$ is also shown in red, for comparison. Once again, this construction is only possible on the parameter space $(x, y) \subseteq \mathbb{R}^2$ because it is also a vector space: it does not make sense on the actual manifold defined by the graph of f, since point addition is not defined. For each direction, we use a bisection to seek the point $\boldsymbol{q}$ such that the straight line $\boldsymbol{pq}$ is unit length:

$$\ell_{\mathcal{M}}(\boldsymbol{pq}) = \int_0^1 \sqrt{(\boldsymbol{pq})^T\,[\mathcal{M}](\boldsymbol{p} + t\boldsymbol{pq})\,(\boldsymbol{pq})}\,dt = 1. \tag{3.126}$$

This unit ball lies completely inside the true unit ball of $\mathcal{M}$, since for each endpoint $\boldsymbol{q}$ of a straight line, there exists a shorter geodesic curve joining $\boldsymbol{p}$ and $\boldsymbol{q}$.

3.5 Curvature

Riemannian geometry is the context of anisotropic mesh adaptation, as we use Riemannian metrics to prescribe sizes and orientation. Riemannian geometry studies the equivalence between Riemannian manifolds; in other words, it addresses the question of whether two Riemannian manifolds are (locally or globally) equivalent, see the discussion in Section 3.2.5. The equivalence of two manifolds is asserted through local invariants, which are preserved by isomorphisms: if both manifolds share these invariants, then they are said to be equivalent. The local invariant that classifies Riemannian manifolds is **curvature**. As its name suggests, Riemannian manifolds without curvature are said to be **flat**. They are indistinguishable from (i.e., isometric to) the Euclidean space. As we aim at creating meshes isometric to a regular tiling of the plane, we wish to know when such an isometric mesh can be created on a given manifold. This boils down to knowing if the manifold is flat.

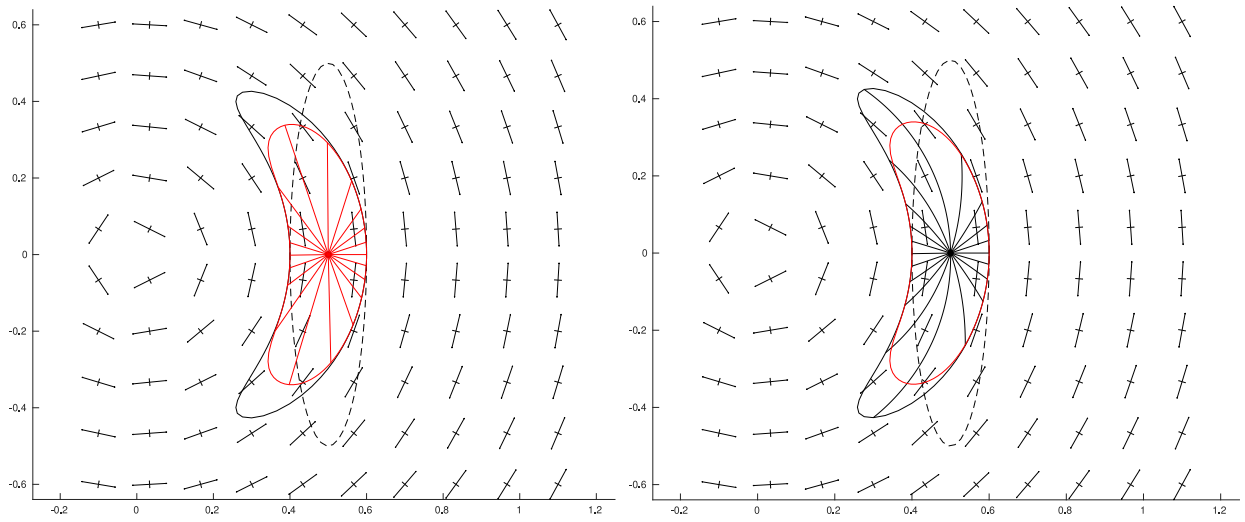


Figure 3.8 Unit spheres for $\mathcal{M}$ centered at $\boldsymbol{p}$ and the parameter space with the field of principal directions scaled by the principal lengths. On the left, the curve in red is obtained by leaving $\boldsymbol{p}$ along straight lines until $\ell_{\mathcal{M}} = 1$. On the right, the geodesic unit sphere (i.e., the true unit sphere) is obtained by leaving $\boldsymbol{p}$ along geodesics, by integrating the geodesic equation. The dashed ellipse is the tangential unit sphere at $\boldsymbol{p}$ for the metric $\mathcal{M}_{\boldsymbol{p}}$.

For the sake of brevity, an intuitive presentation of the notion of curvature, as done in Lee [56], is presented in Appendix A.2.2. The main idea is that one can evaluate how a manifold curves by measuring the impossibility to uniquely extend a vector by parallel transport. This yields a condition on the Levi-Civita connection (and thus on the metric) of the manifold: it is said to satisfy the **flatness criterion** if for any three vector fields X, Y, Z on M, we have:

$$\nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z = \nabla_{[X,Y]} Z, \tag{3.127}$$

where $[X, Y]$ is the Lie bracket of X and Y, see Section 3.1.1. In particular, this criterion is satisfied by the Euclidean connection, as well as by the Levi-Civita connection on any flat manifold ([56], Prop. 7.2), as these are isometric to $\mathbb{R}^n$ and the Levi-Civita connection is preserved by isometries. The flatness criterion can be handily encoded in the form of a map

$$R : \mathfrak{X}(M) \times \mathfrak{X}(M) \times \mathfrak{X}(M) \to \mathfrak{X}(M) : (X, Y, Z) \mapsto R(X, Y)Z, \tag{3.128}$$

called the **curvature endomorphism** and defined by:

$$R(X, Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X,Y]} Z. \tag{3.129}$$

For any vector field Z, $R(X,Y)Z$ is a vector field that measures how its mixed covariant derivatives in the directions X and Y fail to commute. Because the different by-products of Leibniz's rule cancel each other when expanding the curvature endomorphism, the map R is actually multilinear over smooth function, and thus defines a 4-covariant tensor $Rm = R_{ijk\ell}\, dx^i \otimes dx^j \otimes dx^k \otimes dx^\ell$ called the **(Riemann) curvature tensor**[9], whose expression in coordinates is:

$$R_{ijkl} = \mathcal{M}_{lm}\left(\partial_i \Gamma^m_{jk} - \partial_j \Gamma^m_{ik} + \Gamma^p_{jk}\Gamma^m_{ip} - \Gamma^p_{ik}\Gamma^m_{jp}\right). \tag{3.130}$$

The curvature tensor contains all the information regarding the curvature of an abstract manifold. As a 4-tensor, Rm has n^4 components, but it satisfies several symmetry and skew-symmetry relations which significantly reduce its number of independent components. Denoting n the dimension of the manifold, the curvature tensor has $n^2(n^2-1)/12$ independent components. In particular, it has 0, 1 and 6 independent components for 1-, 2- and 3-manifolds, respectively. Thus, the curvature tensor vanishes for 1-dimensional manifolds. A very important property of the curvature tensor is that it is a local invariant under isometries ([56], Prop. 7.6), and thus measures how a manifold locally fails to be flat. This is summarized in the following key result ([56], Th. 7.10):

Theorem 1. *A Riemannian manifold is flat if and only if $Rm = 0$.*

As a consequence, **all 1-manifolds are flat**, since their curvature tensor always vanishes. In Chapter 4, we define unit meshes as triangulations isometric to any regular tiling on $\mathbb{R}^2$. Clearly, they cannot exist if the target manifold is not flat, so we need only verify that $Rm = 0$ to assert the flatness of the manifold, which, for 2- and 3-manifolds, amounts to verify that 1 or 6 independent components vanish. These are stored in two standard contractions of Rm: the **Ricci curvature tensor** Rc and the **scalar curvature** S. The Ricci tensor is the contraction of the curvature tensor on its first and last indices, and the scalar curvature is the trace of the Ricci tensor, defined by $\mathrm{tr}_{\mathcal{M}}\, A \triangleq \mathcal{M}^{-1}_{ij} A_{ij}$ or $\mathrm{tr}_{\mathcal{M}}\, A \triangleq \mathcal{M}^{-1}_{il} A_{ijkl}$:

$$Rc \triangleq \mathrm{tr}_{\mathcal{M}}\, Rm, \qquad R_{ij} = \mathcal{M}^{-1}_{km} R_{kijm}, \tag{3.131}$$

and

$$S \triangleq \mathrm{tr}_{\mathcal{M}}\, Rc = \mathcal{M}^{-1}_{ij} R_{ij} = \mathcal{M}^{-1}_{ij}\left(\partial_k \Gamma^k_{ij} - \partial_i \Gamma^k_{kj} + \Gamma^p_{ij}\Gamma^k_{kp} - \Gamma^p_{kj}\Gamma^k_{ip}\right). \tag{3.132}$$

For 2-manifolds, the scalar curvature completely determines the curvature 4-tensor, thus also the Ricci curvature tensor. This means that to check for the flatness of a 2-manifold, it is sufficient to verify that $S = 0$ everywhere. As we show hereafter, it is equivalent to

[9] The Riemann tensor field is noted Rm, but its coordinates write with the letter R, similarly to the Ricci tensor field.

measuring the Gaussian curvature of the manifold. For 3-manifolds, the Ricci curvature tensor is symmetric and thus has 6 components in three dimensions, hence it completely determines Rm.

When studying the differential geometry of surfaces in $\mathbb{R}^3$, curvature is characterized by the *second fundamental form* $\mathbf{II}(X, Y)$. It is used, for instance, in Chapter 8 to derive an isotropic size field from the curvature of the CAD model. The second fundamental form is defined as the orthogonal component (with respect to the surface) of the ambient connection of $\mathbb{R}^3$. In fact, if M is an embedded submanifold of $\widetilde{M}$ with ∇ and $\widetilde{\nabla}$ their respective Levi-Civita connection, the following tangential-orthogonal decomposition, called the **Gauss formula**, holds on M for vector fields X, Y:

$$\widetilde{\nabla}_X Y = \nabla_X Y + \mathbf{II}(X, Y). \tag{3.133}$$

From the second fundamental form is derived the *shape operator*, whose eigenvalues are the **principal curvatures** κ_1, κ_2, from which are defined the **mean curvature** $H = (\kappa_1 + \kappa_2)/2$ and the **Gaussian curvature** $K_g = \kappa_1\kappa_2$. The second fundamental form is an extrinsic operator, because its definition depends on the embedding chosen to represent the surface M in the ambient space. As a consequence, the principal curvatures are not local invariants of the surface, and neither is the mean curvature. It is a remarkable fact, however, that Gaussian curvature *is* invariant under local isometries, thus is an intrinsic curvature measure of the manifold that can be obtained from the metric only. This is summarized in Gauss's celebrated **Theorema Egregium**:

Theorem 2. *Let M be an embedded surface of $\mathbb{R}^3$. The Gaussian curvature K_g of M is invariant under local isometries. Moreover, $K_g = S/2$.*

It follows that for a 2-manifold $(M, \mathcal{M})$, the curvature measures introduced above are completely determined by the Gaussian curvature. In particular:

$$S = 2K_g, \qquad Rc = K_g\mathcal{M}, \qquad Rm = \frac{K_g}{2}\mathcal{M} \owedge \mathcal{M}, \tag{3.134}$$

where the operator $\owedge$ is called the *Kulkarni-Nomizu* product: it takes two covariant 2-tensors and returns a covariant 4-tensor with the same symmetries as Rm, see e.g. [56].

Example: Curvature of developable surfaces

We illustrate the computation of the curvature on a particular class of 2-manifolds, called **developable surfaces**. These surfaces, such as cylinders and cones, can be obtained by bending and folding the plane without tearing it, and are isometric to the Euclidean plane. Clearly, most surfaces are *not* developable, however, we show in Chapter 4 that truly unit meshes can be generated on developable surfaces. For $(u, v) \in \mathbb{R}^2$, we consider the developable surface obtained by "extruding" the graph of a single variable function $f(u)$ along the axis of the other variable. We consider the developable surface D described by the map $F : \mathbb{R}^2 \to \mathbb{R}^3 : (u, v) \mapsto (u, v, f(u))$. As a surface embedded in $\mathbb{R}^3$, D inherits the Euclidean metric of $\mathbb{R}^3$. From (4.64), this metric writes in (u, v) coordinates:

$$[\mathcal{M}] = [F^* \overline{\mathcal{M}}_{\mathbb{R}^3}] = \begin{pmatrix} 1 + (f'(u))^2 & 0 \\ 0 & 1 \end{pmatrix}. \tag{3.135}$$

From (3.108), the only nonzero Christoffel symbol of $\mathcal{M}$ is:

$$\Gamma^1_{11} = \frac{f'(u) f''(u)}{1 + (f'(u))^2}. \tag{3.136}$$

An important property of developable surfaces is that their Gaussian curvature vanishes everywhere, a result that can also be shown by examining their scalar curvature $S = 2K$. A direct computation using (3.132) shows that $S = 0 = K$, and we conclude that the manifold is flat by examining only its metric.

3.6 Operations on Riemannian metrics

This concludes the theoretical notions used in rest of the thesis. In a more practical point of view, we briefly review the standard operations on metrics for mesh adaptation that are used in this thesis. These are applied on the mesh in the physical space, with metrics stored at the mesh vertices. In the following, we simply write $\mathcal{M}$ to denote the component matrix of the metric.

3.6.1 Metric interpolation and derivatives

In the anisotropic mesh adaptation pipeline, metrics are stored only at mesh vertices and need to be interpolated for various operations. Linear and geometric interpolation schemes for the eigenvalues or the metrics themselves [1, 61] are known to give poor results, because the interpolated metrics tend to become isotropic or do not preserve the volume. We consider

instead the log-Euclidean interpolation scheme, initially described in [62] for linear elements, and recently extended to high-order elements in [17, 26]. Given a set of weights w_i that sum to one and a set of metrics $\mathcal{M}_i$, the log-Euclidean interpolation metric is defined by:

$$\mathcal{M} = \exp\left(\sum_i w_i \ln \mathcal{M}_i\right), \tag{3.137}$$

with the matrix exp and ln operations described in Section 3.2.4. On $\mathcal{P}^1$ triangles, the interpolation metric is computed in the reference space, using the Lagrange basis $\phi_i(\boldsymbol{\xi})$ that sums to one and coincide with the barycentric coordinates. The weights w_i need not be positive, which allows the straightforward extension to $\mathcal{P}^2$ triangles in the reference space using the quadratic Lagrange basis. For both linear and quadratic triangles, the interpolated metric writes:

$$\mathcal{M}(\boldsymbol{x}(\boldsymbol{\xi})) = \exp\left(\sum_{i=1}^{N} \phi_i(\boldsymbol{\xi}) \ln \mathcal{M}(\boldsymbol{X}_i)\right), \tag{3.138}$$

with $N = 3$ or 6. This defines a continuous metric field from metrics known only at the vertices.

Background mesh interpolation

During the mesh generation, edges are curved with the goal to either minimize their length in the metric space, or a measure of distortion with respect to an ideal triangle, see Chapter 7. Both objective functions require repeatedly interpolating in the metric field. This is done on a background mesh, where the metrics are stored and kept unmodified during the whole meshing process. The background mesh is the last computed anisotropic mesh, on which the metrics are computed from an a posteriori error estimator. Metric interpolation at a point $\boldsymbol{p}$ of the new mesh consists in (i) localizing $\boldsymbol{p}$ in the background mesh, then (ii) interpolating the metric at the localized point using the log-Euclidean scheme. Following the notation in [26], let $\boldsymbol{p}$ be a point in the new mesh and $\hat{\boldsymbol{p}}$ its equivalent in the background mesh, with reference coordinates $\hat{\boldsymbol{\xi}}$. The interpolation metric $\mathcal{M}(\boldsymbol{p})$ is computed at $\hat{\boldsymbol{\xi}}$ as follows:

$$\mathcal{M}(\boldsymbol{p}(\hat{\boldsymbol{\xi}})) = \exp\left(\sum_{i=1}^{N} \phi_i(\hat{\boldsymbol{\xi}}) \ln \mathcal{M}(\hat{\boldsymbol{X}}_i)\right). \tag{3.139}$$

In this expression, the $\hat{\boldsymbol{X}}_i$ are the vertices of the triangle of the background mesh in which lies $\boldsymbol{p}$. Point localization is handled with an RTree [63], a tree structure storing the bounding boxes of the triangles in the background mesh. The queried point $\hat{\boldsymbol{p}}$ is given a small box and we detect the overlap between this box and all the boxes the RTree, as illustrated in

Fig. 3.9. For all overlapping boxes, we test if the point is actually in the triangle by inverting the reference-to-physical transformation $\hat{\boldsymbol{\xi}} = F_K^{-1}(\hat{\boldsymbol{p}})$. For a linear background mesh, the reference coordinates are given by:

$$\begin{pmatrix} \hat{\xi} \\ \hat{\eta} \end{pmatrix} = \begin{pmatrix} X_1 - X_0 & X_2 - X_0 \\ Y_1 - Y_0 & Y_2 - Y_0 \end{pmatrix}^{-1} \begin{pmatrix} p_x - X_0 \\ p_y - Y_0 \end{pmatrix} \tag{3.140}$$

and $\hat{\boldsymbol{p}}$ lies inside the background triangle if $\hat{\boldsymbol{\xi}}$ lies inside the reference triangle, i.e., if $\hat{\xi}, \hat{\eta} \geq 0$ and $\hat{\xi} + \hat{\eta} \leq 1$. For a quadratic background mesh, we first test if the triangle is in fact linear, in which case the coordinates above are returned. If the triangle is nonlinear, we invert the nonlinear mapping $\hat{\boldsymbol{p}} - F_K(\hat{\boldsymbol{\xi}}) = \mathbf{0}$ using Newton-Raphson's method and the linear solution (3.140) as initial guess. Currently, curved meshes are generated using the last straight anisotropic mesh as background mesh, so that only $\mathcal{P}^1$ localization and metric interpolation is perfomed in practice. In the future, however, it is expected to iterate over curved meshes until convergence of the mesh/solution pair, using the $\mathcal{P}^2$ interpolation scheme.

Derivatives of the interpolated metric

The derivatives of the metric w.r.t. the front mesh coordinates are required to integrate along the geodesics (for the Christoffel symbols) or minimize the metric-weighted length or distortion to curve the edges. The gradient of the metric are computed from the derivatives of its eigenvalues and eigenvectors, following Aparicio-Estrems et al. [26]. As we use a Hessian-free L-BFGS method for the minimization, only the gradient needs to be computed. It is

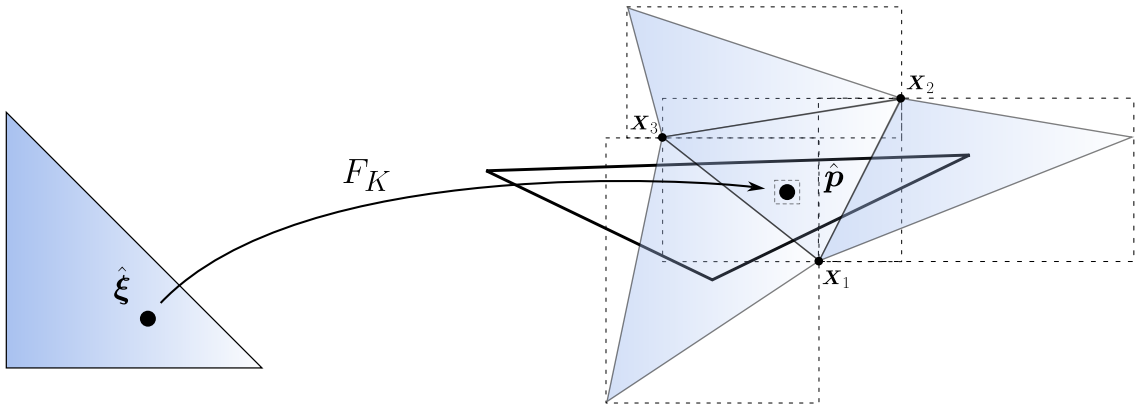


Figure 3.9 Localization in a linear background mesh.

computed from the diagonalized form $L = P\Lambda P^T$ of the log-matrix $L \triangleq \sum_i \phi_i \ln \mathcal{M}_i$, so that:

$$\begin{aligned}\partial_j \mathcal{M} = \partial_j \exp L &= \partial_j \left(P \exp \Lambda \, P^T \right) \\ &= (\partial_j P) \exp \Lambda P^T + P(\partial_j \exp \Lambda) P^T + P \exp \Lambda (\partial_j P^T).\end{aligned} \tag{3.141}$$

The computation of each term is detailed in [26] and in Appendix A.2.3.

3.6.2 Metric intersection

The intersection $\mathcal{M}_{1\cap 2} = \mathcal{M}_1 \cap \mathcal{M}_2 \in \mathrm{Sym}_2^+$ of two metrics prescribes the largest possible sizes while respecting the direction constraints given by each metric. It is generally used to provide a unique metric from several metric fields derived from different indicators (velocity components, pressure, temperature, etc.). In this work, it is only used to grade a metric, as discussed in the next Section. Different schemes exist [64], and we consider intersection based on the simultaneous reduction, which diagonalizes two metrics in the same basis. The simultaneous reduction $\mathcal{N} \triangleq \mathcal{M}_1^{-1}\mathcal{M}_2$ is a positive-definite but not symmetric matrix in general. The metrics $\mathcal{M}_1$ and $\mathcal{M}_2$ are simultaneously diagonalized by P^{-1}, where $P = (\boldsymbol{v}_1, \boldsymbol{v}_2)$ is the matrix of the (nonorthogonal) eigenvectors of $\mathcal{N}$:

$$\mathcal{M}_i = P^{-T} \begin{pmatrix} \mu_{i,1} & 0 \\ 0 & \mu_{i,2} \end{pmatrix} P^{-1}, \quad \text{with } \mu_{i,j} = \boldsymbol{v}_j^T \mathcal{M}_i \boldsymbol{v}_j \tag{3.142}$$

As both metrics are diagonal in that basis, the intersection is obtained by taking the most critical sizes:

$$\mathcal{M}_{1\cap 2} = P^{-T} \begin{pmatrix} \max\limits_{i=1,2} \mu_{i,1} & 0 \\ 0 & \max\limits_{i=1,2} \mu_{i,2} \end{pmatrix} P^{-1}. \tag{3.143}$$

The metric intersection does not commute for more than two metrics. Instead, one can determine the intersection $\mathcal{M}_{\cap_i \mathcal{M}_i}$ by solving a constrained optimization problem to determine the largest ellipse contained in the n metrics to intersect, see e.g. [1]. The intersection of two metrics must be computed a large number of times when e.g. smoothing the metric field, see the next section. To speed up the computation and limit round-off imprecision, we first test if the input metrics are equal, diagonal, or multiples of one another up to a prescribed tolerance. Then, the eigenvectors of the simultaneous reduction, as well as their inverse, are computed explicitly from their symbolic expression when necessary.

3.6.3 Metric gradation

The metric field for mesh adaptation typically depends on the derivatives of the fields of interest. These may exhibit large variations from one vertex to another, which translates into large variations of the metric. For most applications, e.g. when sampling vertices according to the metric, computing quality indicators, evaluating or minimizing approximate lengths, a smooth metric field with controlled variation is desirable. Such a metric is said to be **graded**, and limits the geometric size variation on an edge to a parameter $\beta > 1$, in the sense that for any $\boldsymbol{p}, \boldsymbol{q}$, the size $h_{\boldsymbol{q}}$ cannot exceed $h_{\boldsymbol{p}}\beta^{\ell_{\mathcal{M}}(\boldsymbol{pq})}$. We apply the gradation algorithm proposed by Alauzet [65], which we briefly recall here. A metric at a point $\boldsymbol{p}$ can be extended to the whole domain by enlarging its unit sphere with a prescribed factor . This can be done by multiplying its eigenvectors with either a unique factor $\eta_{\boldsymbol{p}}(\boldsymbol{x})$ or different factors $\eta_{\boldsymbol{p},i}(\boldsymbol{x})$, which reflect the prescribed size variation. These spanned metric fields write respectively:

$$\begin{aligned}
&\mathcal{M}_{\boldsymbol{p}}(\boldsymbol{x}) = \eta^2_{\boldsymbol{p}}(\boldsymbol{x})\mathcal{M}_{\boldsymbol{p}}, \quad \text{with} \quad \eta^2_{\boldsymbol{p}}(\boldsymbol{x}) = \Big(1 + \|\boldsymbol{px}\|_{\mathcal{M}_{\boldsymbol{p}}} \ln\beta\Big)^{-2}, \\
&\mathcal{M}_{\boldsymbol{p}}(\boldsymbol{x}) = P \begin{pmatrix} \eta^2_{\boldsymbol{p},1}(\boldsymbol{x}) & 0 \\ 0 & \eta^2_{\boldsymbol{p},2}(\boldsymbol{x}) \end{pmatrix} \Lambda P^T, \quad \text{with} \quad \eta^2_{\boldsymbol{p},i}(\boldsymbol{x}) = \Big(1 + \sqrt{\lambda_i}\|\boldsymbol{px}\| \ln\beta\Big)^{-2}.
\end{aligned} \tag{3.144}$$

The former preserves the ratios of anisotropy, whereas the latter yields isotropic metrics far from $\boldsymbol{p}$. Each point in the background mesh spans such a metric field, representing its size constraint over the domain. To account for all size prescriptions, the metric at $\boldsymbol{p}$ should be intersected with each metric spanned at $\boldsymbol{p}$ from $\boldsymbol{x}$m however, this has an $\mathcal{O}(N_v^2)$ cost. In practice, the edge-based sweep proposed by Alauzet and summarized in Algorithm 1 is used. For a linear background mesh, each edge is added to the set to be treated. For $\mathcal{P}^2$ triangles, we consider the 15 edges and subedges formed by connecting the $\mathcal{P}^2$ vertices, to enforce the desired gradation in all directions on the element.

Gradation in the metric space, obtained by spanning the metrics according to first line of (3.144), is generally favored as it yields smooth meshes without blurring the anisotropic features. It is however dependent on the mesh topology, and small sizes prescribed orthogonally to the mesh edges are known to propagate far in the domain, creating artifact-like structures [65]. To mitigate this, we implemented one of the solution proposed in [65], namely mimicking an exponential size variation away from the anisotropic features (exp-metric), so that the size prescription quickly decreases as we move away from a vertex and its influence is localized. In practice, the intersection metric is stored in a different array and the metrics are spanned from the unmodified metric. The gradation parameter increases by a factor $\alpha > 1$

after each sweep, which simulates a layer-by-layer size decrease and effectively removes the artifacts.

Algorithm 1 Edge-based metric gradation control.

```
correction ← true
while correction is true do
    correction ← false
    M^new ← M                                   ▷ Only for the exp-metric method
    for all edges pq do
        Intersect M_q (M_q^new if exp-metric) with M_p(q) spanned from p
        Intersect M_p (M_p^new if exp-metric) with M_q(p) spanned from q
    end for
    M ← M^new                                   ▷ Only for the exp-metric method
    if any metric was modified then
        correction ← true
        β ← αβ                                  ▷ Only for the exp-metric method
    end if
end while
```

The gradation is illustrated in Fig. 3.10 for the function:

$$f(x,y) = x^3 + x^2 y + \tanh\left(\frac{2x - \sin(5y)}{\delta}\right) \quad \text{on} \quad \Omega = [-2,2] \times [-1,1]. \tag{3.145}$$

with $\delta = 0.02$, proposed in [38]. Here, we reduced the value of δ to show how small sizes propagate. An anisotropic mesh is generated with the goal of minimizing the H^1 seminorm of the interpolation error on Ω. The associated Riemannian metric is presented in Section 5.3. Its unit balls are shown on the bottom left, and feature abrupt variations from one vertex to the other on the background mesh. The metric field after gradation is regularized and both the sizes and the directions vary smoothly.

3.7 Conclusion

This chapter presented the notions of geometry required to work with Riemannian metrics. They were introduced in a formal, but hopefully accessible way. Riemannian metrics and isometries are used throughout this thesis, starting with the next chapter discussing unit elements. Tangent vectors, differentials and pullbacks are used, among others, to interpret the invariants of linear unit elements, see Section 4.1. Curved unit edges are also shown to have geodesic edges, which justifies their introduction. Lastly, the quantitative results on curvature are not used per se, but they give a criterion to identify isometric manifolds. On such manifold, we construct in the next chapter perfectly unit isometric triangulations with

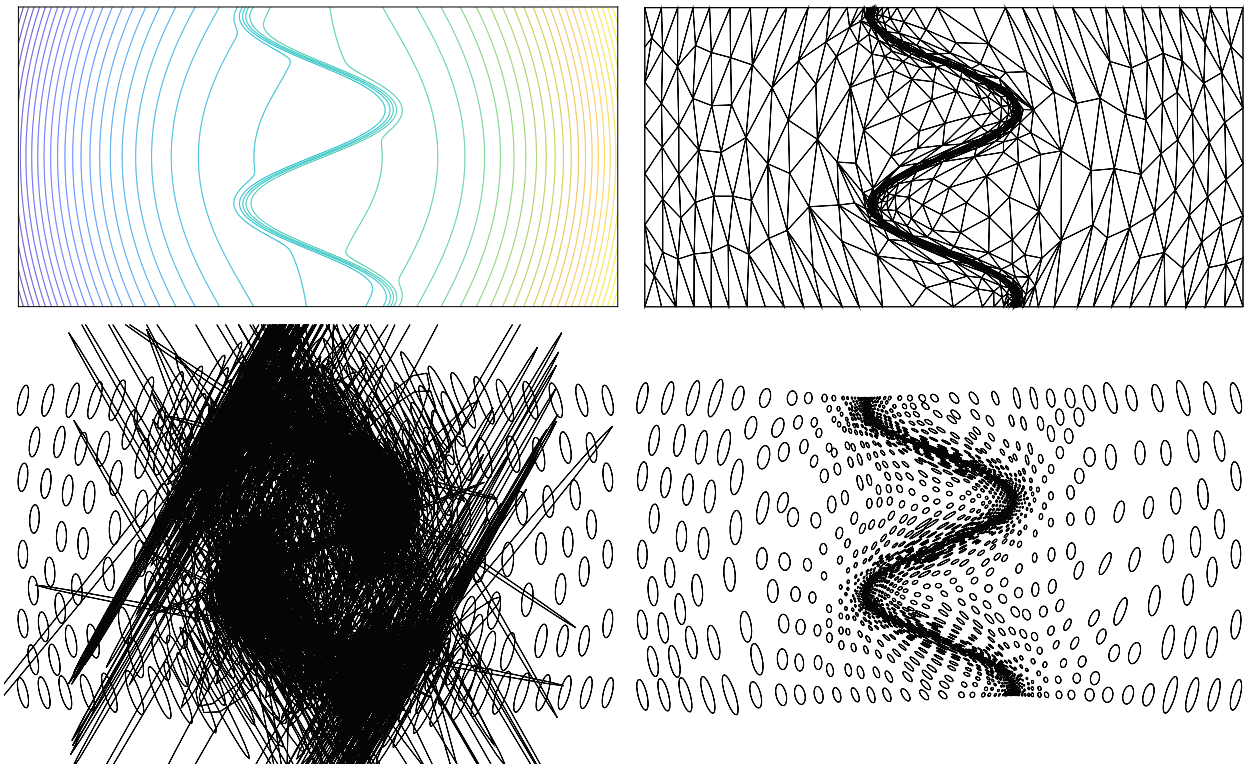


Figure 3.10 Top: Contours of the target function and adapted mesh for the H^1 seminorm without gradation. Bottom: Scaled unit balls associated to the metric field without and with gradation with $\beta = 2$.

respect to a uniform tiling of the plane. Practical operations on discrete metrics, such as interpolation and gradation, were also reviewed. These are extensively used in our meshing pipeline.

Following Section 3.2.2 and with the goal of controlling the interpolation error with respect to a real-valued function f, we conclude this chapter with the following discussion, which could motivate future investigations. Given a function f and its graph, one can create on a parameter space $U \subseteq \mathbb{R}^2$ a triangulation adapted to metric $\mathcal{M}$ induced by the graph. By default, the metric is obtained from the ambient Euclidean metric of $\mathbb{R}^3$, so that the triangulation has lengths prescribed by the Euclidean metrics: for example, if the function f is constant, the associated surface is the graph of a plane and the induced metric is the Euclidean metric of $\mathbb{R}^2$, so that the triangulation on U has edges of quasi-unit length with respect to $\overline{\mathcal{M}}$. The mesh density can be uniformly adjusted, if need be, by changing the measure of distances on the graph, i.e., by modifying the Euclidean metric of $\mathbb{R}^3$ by a constant c: $[\overline{\mathcal{M}}'_{\mathbb{R}^3}] = cI_3$, with I_3 the 3×3 identity matrix. Because the graph is unique and intrinsic to f, however, there is no reason to believe that this triangulation offers any

particular property on the control of the interpolation error. Indeed, the induced metric differs from the optimal metric minimizing the continuous interpolation error, proposed by Alauzet and Loseille [8, 9] and described in Chapter 5, which we sometimes refer to as the *error metric*. Thus, a natural question arises: for two-dimensional applications, (when) is the error metric also induced by the graph of a surface in $\mathbb{R}^3$? A priori, the answer is negative as the error metric is represented by a 2×2 symmetric matrix, thus has 3 independent components, whereas the induced metric only has 2 independent components in f_u, f_v. It is possible to show[10] that for any *uniform* metric on Ω, the Euclidean metric of $\mathbb{R}^3$ can be scaled uniformly so that $\mathcal{M}$ is induced by a graph, however, we do not expect such a graph to expect for nonuniform metrics.

[10]The metric should satisfy $\det \mathcal{M} = (a + c) - 1$ to be induced by a graph. Since the determinant scales with α^n in dimension n and the coefficients scale with α when the Euclidean metric is scaled by α, we can solve $\alpha^n \det \mathcal{M} = \alpha(a + c) - 1$ for the uniform scaling α. This coefficient depends on the metric components, and thus varies from one point to another in general, however, so that there is a unique scaling of $\overline{\mathcal{M}}_{\mathbb{R}^3}$ only if $\mathcal{M}$ is constant.

CHAPTER 4 UNIT AND ISOMETRIC TRIANGULATIONS

This thesis builds on the **continuous mesh theory**, introduced by Loseille [1] and Alauzet and Loseille [8, 9] to provide a continuous point of view to the discrete problem of mesh adaptation. In this framework, a Riemannian metric $\mathcal{M}$ is the continuous counterpart of a mesh and is also called a **continuous mesh**. The properties of a discrete mesh, such as the number of vertices or the elementwise interpolation error, have a continuous counterpart in the continuous mesh framework and can be written in terms of the metric. As a result, differentiation and optimization can be performed with respect to the Riemannian metric, and the intractable problem of finding an optimal discrete triangulation minimizing an error functional can be translated into the well-posed problem of finding an optimal metric tensor field. The fact that the optimal metric minimizing the linear interpolation error on linear simplices has a closed-form expression largely contributed to the success of the continuous mesh framework. As a mesh encodes the size and orientation of its discrete elements, a Riemannian metric describes the geometry of an equivalence class of simplices, called **unit elements**. In essence, unit elements are a way of describing *identical* elements when viewed through the eyes of a metric, sharing invariants such as their edge lengths and area with respect to the metric. In the continuous mesh framework, simplices are said to be unit if their edges have unit length with respect to $\mathcal{M}$. The precise length of 1 is actually of little consequence, as setting another target length simply amounts to scaling the metric tensor, that is, it only affects the mesh density. Instead, the idea is to generate elements whose edges have the same length. Indeed, if the function to approximate (or, rather, its derivatives) is isotropic, the vertices should be disposed uniformly, in order not to reduce the convergence rate. Choosing nonuniform elements amounts to add degrees of freedom in an unjustified direction, thus increasing the mesh complexity without any reduction of the approximation error. If the Riemannian metric accurately represents the objective function, the goal of the continuous mesh framework is to create uniform meshes with respect to this metric tensor, called **unit meshes**. Generating a mesh made exclusively of unit elements is generally impossible, so a relaxed definition is needed in the form of **quasi-unit** elements. In practice, linear anisotropic meshes obtained from metric-based mesh generators are made of quasi-unit elements. Generating quasi-unit linear meshes is now well understood, and anisotropic meshing libraries are fast and robust [66, 67]. For the curvilinear case, several questions remain open, starting with an adequate definition of unit simplices. The original definition of unitness involves only the edges and is not suited for high-order elements, as curved simplices with unit edges may either be invalid, or may not share the same invariants

as linear unit elements. The difficulty lies in the fact that the definition of linear unit elements is based on a property they share, having unit edges, rather than on the transformation they come from. A broader and descriptive definition that incorporates linear elements as a special case is thus required. To keep curved meshes competitive, a constructive definition involving the edges would be preferred. In his thesis, Rochery proposed to define unit high-order elements as transformations whose Jacobian matrix relates to the metric [17]. A distortion indicator can be derived from this definition, which matches the distortion function minimized by Aparicio-Estrems et al. [25, 26], however this considers an expensive global optimization problem, which becomes intractable for large problems. In contrast, unit linear elements are generated by repeatedly creating or collapsing points until all edges are of unit length, a local criterion leading to fast algorithms. A definition of curvilinear unit simplices involving the edges would thus be desirable. In most work on curvilinear meshing, the go-to strategy is to curve the edges to minimize their length with respect to the metric, that is, to approximate the geodesics. However, to our best knowledge, a formal link between unit elements and geodesics is still missing in the literature.

In this chapter, we focus on the two-dimensional case, but the ideas are straightforward to generalize to three dimensions. Sections 4.1 through 4.4 are concerned with the extension of unitness to curvilinear simplices. We start by giving the usual definition and properties of unit triangles, then, as a counterexample, we discuss curved triangles illustrating its shortcomings for high-order elements. Then, as suggested by Rochery [17, 33], we introduce in Section 4.2 a stronger definition of unitness based on the pointwise Jacobian matrix $J_{\triangle \to K}$ of the transformation of an equilateral triangle. This definition implies the original one for both linear and high-order triangles. We then look at the concept of unitness using the tools of Riemannian geometry, observing that the Jacobian matrix $J_{\triangle \to K}$ pulls back the metric $\mathcal{M}$ to the Euclidean metric. This leads us to introduce a new definition of unitness based on **Riemannian isometries**, which englobes the previous definitions. This is the main contribution of this chapter. The properties of **isometric unit elements** are discussed in Section 4.3, and the conditions under which such isometries exist are investigated. With this definition, unit elements are simplices isometric to $K_{\triangle}$. One can then choose another reference simplex not isometric to $K_{\triangle}$, from which a different, nonisometric class of unit elements is defined. We show that unit meshes only exist on Riemannian manifolds isometric to $\mathbb{R}^n$, i.e., flat manifolds, and we construct, through optimization, examples of isometric unit triangulations on such 2-manifolds, that is, on developable surfaces. The presented triangulations are isometric to regular tilings of $\mathbb{R}^2$ made of copies of either $K_{\triangle}$ or $K_{\triangleright} \triangleq \widehat{K}$. On non-flat manifolds, we show examples of quasi-isometric triangulations. The last section of this

chapter is dedicated to pointwise interpolation error estimates on unit elements. Classical estimates on linear simplices are reviewed, then estimates on curvilinear unit simplices are discussed. We detail the error estimate for linear interpolation on quadratic triangles proposed by Rochery, then, as the second contribution of this chapter, we propose a pointwise error estimate for interpolation of arbitrary order on $\mathcal{P}^q$ triangles. This is done by including the reference-to-physical transformation into the existing high-order error estimate for linear elements.

4.1 Linear unit elements

We start by giving the usual definitions of unit and quasi-unit elements, from Frey [61] or Loseille [1], which are sufficient for anisotropic mesh adaptation with linear triangles:

Definition 4.1. *A **linear triangle** K with edges e_i is **unit** with respect to the metric $\mathcal{M}$ if:*

$$\ell_{\mathcal{M}}(e_i) = 1, \qquad \text{for } i = 1, 2, 3, \tag{4.1}$$

*i.e., its edges are of unit length when measured with respect to $\mathcal{M}$. K is **quasi-unit** with respect to $\mathcal{M}$ iff:*

$$\ell_{\mathcal{M}}(\boldsymbol{e}_i) \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right], \qquad \forall\ i = 1, 2, 3. \tag{4.2}$$

Since their definition does not change throughout this chapter, it is useful to define **unit edges** w.r.t. $\mathcal{M}$ as edges e such that $\ell_{\mathcal{M}}(e) = 1$. The definition above states that a triangle is unit if its edges are unit. Any equilateral triangle with unit edges for the Euclidean metric $\overline{\mathcal{M}}$ is a unit triangle with respect to $\overline{\mathcal{M}}$. This is indeed the motivation behind unit elements, as anisotropic adaptation methods aim at producing equilateral triangles deformed by the metric $\mathcal{M}$. In particular, the reference regular triangle $K_{\triangle}$, introduced in Section 2.3, is a unit triangle for the Euclidean metric. Any rotation or translation of an equilateral triangle is still an equilateral triangle, hence is still unit with respect to $\overline{\mathcal{M}}$. Recalling that the linear transformation associated to $\mathcal{M}^{-1/2}$ preserves the lengths, it follows that there are infinitely many unit triangles for a given metric. More specifically, the complete set of linear triangles that are unit with respect to $\mathcal{M}$ is described as follows [1, 17]:

Proposition 4.1. *Let $\mathcal{M}$ be a constant Riemannian metric. The set $\mathcal{K}_{\mathcal{M}}$ of linear triangles unit with respect to $\mathcal{M}$ is:*

$$\mathcal{K}_{\mathcal{M}} = \left\{ K = \mathcal{M}^{-1/2} R K_{\triangle} + \boldsymbol{x} \mid R \in SO_2,\ \boldsymbol{x} \in \mathbb{R}^2 \right\}. \tag{4.3}$$

This makes sense as long as $\mathcal{M}^{-1/2}$, and thus $\mathcal{M}$, is constant, as both transformations F_K and $F_\triangle$ are linear functions of $\boldsymbol{\xi}$. For practical applications, the metric field is not constant, and we consider quasi-unit triangles with respect to the interpolated metric between vertices of the background mesh, on which the metric field is stored. Thus, quasi-unit triangles satisfy the relation above for some interpolated metric $\mathcal{M}_{\boldsymbol{p}}^{-1/2}$. For a given unit triangle, the rotation R is unique and constant. All the operations above are isometries, thus the overall transformation is also an isometry, but they do not commute with each other, hence triangles unit w.r.t. $\mathcal{M}$ are not rotations of one another in general, Fig. 4.1.

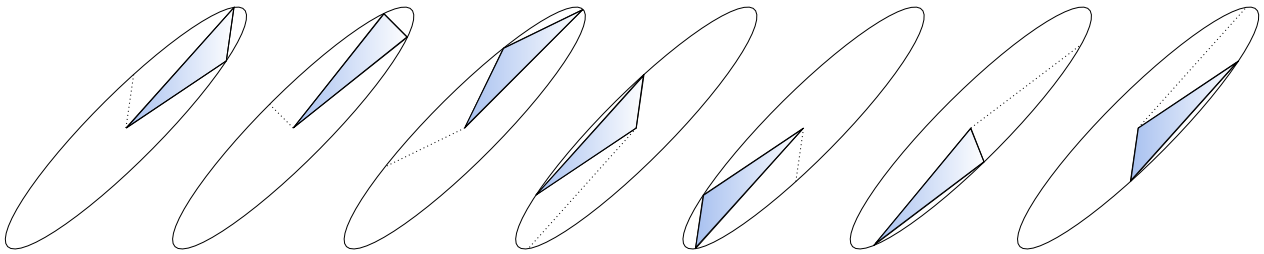


Figure 4.1 Linear unit triangles for different rotations of $K_\triangle$ and unit sphere $\partial\mathcal{B}_1(\boldsymbol{p})$ of a constant metric. The dotted line is a copy of the third edge, also unit with respect to $\mathcal{M}$. Adapted from [1].

A linear triangle is unit in only one metric, and this metric always exists. This is easily seen by writing down the linear system $\ell^2_{\mathcal{M}}(e_i) = 1$ made of $n(n+1)/2$ equations which are linear in the $n(n+1)/2$ independent coefficients of the metric to determine, and whose determinant $|K|_{\overline{\mathcal{M}}}$ never vanishes [1]. A metric thus describes an equivalence class of unit elements, and unit elements each describe the metric in which they are unit. Linear unit elements share the following invariants, ([17], Prop. 2.1.3):

Proposition 4.2. *Other than their edges being unit, i.e., $\ell^2_{\mathcal{M}}(e_i) = e_i^T \mathcal{M} e_i = 1$, linear triangles that are unit w.r.t. the constant metric $\mathcal{M}$ share the following invariants:*

1. *The angles formed by their edges are such that:*

$$\cos\theta(e_i, e_j) = e_i^T \mathcal{M} e_j = \begin{cases} 1/2 \text{ if } e_i \text{ and } e_j \text{ have the same origin,} \\ -1/2 \text{ otherwise,} \end{cases} \tag{4.4}$$

   hence their inner angles are $\pi/3$.

2. *Their area with respect to $\mathcal{M}$ is:*

$$|K| = |K|_{\overline{\mathcal{M}}} \sqrt{\det \mathcal{M}} = \frac{\sqrt{3}}{4}. \tag{4.5}$$

3. *Let H be a symmetric matrix. Then:*

$$\sum_{i=1}^{3} e_i^T H e_i = \frac{3}{2} \operatorname{tr}(\mathcal{M}^{-1/2} H \mathcal{M}^{-1/2}). \tag{4.6}$$

4. *Their exists a rotation matrix R such that the (constant) Jacobian matrix of their reference-to-physical transformation $J_K = J_{\widehat{K}\to K}$ is:*

$$J_K = \mathcal{M}^{-1/2} R J_\triangle, \tag{4.7}$$

where $J_\triangle = J_{\widehat{K}\to K_\triangle}$ is the Jacobian matrix of the transformation from the reference element $\widehat{K}$ to the reference regular element $K_\triangle$.

Proof. See [1, 8, 17].

The first two invariants are quite straightforward, and the second was shown in (3.90). The third plays a key role when writing interpolation error estimator involving only the metric, as the dominant term of the interpolation error for both $\mathcal{P}^1$ and $\mathcal{P}^k$ interpolation can be written with a quadratic form. If H is also positive-definite, which is the case for error estimators, this invariant derives from a stronger result, namely that the isometry $F(\boldsymbol{y}) = \mathcal{M}^{-1/2} R \boldsymbol{y}$ preserves the energy of the edges with respect to the metric $\mathcal{H}$ whose representation in coordinates is H. Indeed, suppose $K_\triangle$ and K are subsets of the Riemannian manifolds $M = (\mathbb{R}^2, \overline{\mathcal{M}})$ and $N = (\mathbb{R}^2, \mathcal{M})$, respectively. Let E_i and e_i denote tangent vectors to the edges of each triangle at vertices $\boldsymbol{y}_i$ with $i = 1, 2, 3$, so that $e_i = dF_{\boldsymbol{y}_i}(E_i)$, or in other words, each e_i is the pushforward of E_i by F at $\boldsymbol{y}_i$, Fig. 4.2. Let $\mathcal{H}$ be a Riemannian metric on N, represented by the symmetric positive-definite matrix H. The pullback tensor of $\mathcal{H}$ at $\boldsymbol{y}_i \in K_\triangle$ is defined according to (3.33) as:

$$(F^*\mathcal{H})_{\boldsymbol{y}_i}(E_i, E_i) = \mathcal{H}(dF_{\boldsymbol{y}_i}(E_i), dF_{\boldsymbol{y}_i}(E_i)) = \mathcal{H}(e_i, e_i). \tag{4.8}$$

As an isometry, F preserves the energy of curves. If the edges are parameterized by $t \in [0, 1]$, this writes:

$$\begin{gathered} E_{F^*\mathcal{H}}(E_i) = E_{\mathcal{H}}(e_i) \\ \downarrow \\ \frac{1}{2}\int_0^1 \langle E_i'(t), E_i'(t)\rangle_{F^*\mathcal{H}_{E_i(t)}}\, dt = \frac{1}{2}\int_0^1 \langle e_i'(t), e_i'(t)\rangle_{\mathcal{H}_{e_i(t)}}\, dt. \end{gathered} \tag{4.9}$$

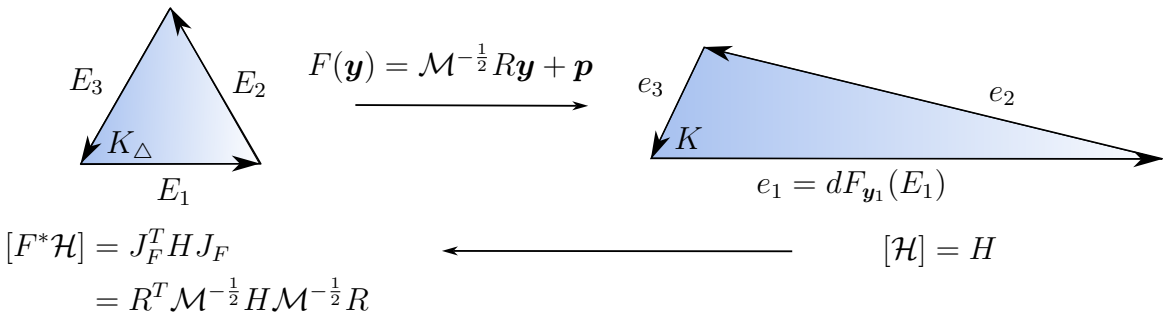


Figure 4.2 The isometry F preserves the energy of the edges of $K_\triangle$.

Since all edges are straight lines, the velocities can be identified to the edges themselves viewed as tangent vectors $E_i \in T_{\boldsymbol{y}_i} M$ and $e_i \in T_{F(\boldsymbol{y}_i)} N$, thus the integrals become:

$$\begin{aligned} \frac{1}{2}\, E_i^T [F^*\mathcal{H}] E_i &= \frac{1}{2}\, e_i^T [\mathcal{H}] e_i \\ &\Big\downarrow \quad [F^*\mathcal{H}] = J_F^T [\mathcal{H}] J_F \\ \frac{1}{2}\, E_i^T R^T \mathcal{M}^{-1/2} H \mathcal{M}^{-1/2} R E_i &= \frac{1}{2}\, e_i^T H e_i \end{aligned} \tag{4.10}$$

With a direct computation on $K_\triangle$, the energy of the boundary ∂K is then:

$$\begin{aligned} E_{\mathcal{H}}(\partial K) &= E_{F^*\mathcal{H}}(\partial K_\triangle) \\ &\Big\downarrow \\ \frac{1}{2} \sum_{i=1}^{3} e_i^T H e_i &= \frac{1}{2} \sum_{i=1}^{3} E_i^T R^T \mathcal{M}^{-1/2} H \mathcal{M}^{-1/2} R E_i \\ &= \frac{3}{4}\, \mathrm{tr}\,(R^T \mathcal{M}^{-1/2} H \mathcal{M}^{-1/2} R) \\ &= \frac{3}{4}\, \mathrm{tr}\,(\mathcal{M}^{-1/2} H \mathcal{M}^{-1/2}), \end{aligned} \tag{4.11}$$

since $\mathrm{tr}(R^T A R) = \mathrm{tr}(A)$. This is equivalent to the third invariant. Because the metric of M is the Euclidean metric, the trace of the matrix $\mathcal{M}^{-\frac{1}{2}} H \mathcal{M}^{-\frac{1}{2}}$ is equal to the **trace of $F^*\mathcal{H}$ with respect to $\overline{\mathcal{M}}$**, defined by:

$$\mathrm{tr}_{\overline{\mathcal{M}}} F^*\mathcal{H} \triangleq (\overline{\mathcal{M}}^{-1})_{ij} (F^*\mathcal{H})_{ij} = (\overline{\mathcal{M}}^{-1})_{ij} (\mathcal{M}^{-\frac{1}{2}} H \mathcal{M}^{-\frac{1}{2}})_{ij} = (\mathcal{M}^{-\frac{1}{2}} H \mathcal{M}^{-\frac{1}{2}})_{ii} \tag{4.12}$$

and such that:

$$\mathrm{tr}_{\overline{\mathcal{M}}}\, F^*\mathcal{H} = \mathrm{tr}_{\mathcal{M}}\, \mathcal{H} = (\mathcal{M}^{-1})_{ij} H_{ij}. \tag{4.13}$$

Lastly, the fourth invariant immediately follows from Proposition 4.1, but is far-reaching nonetheless, as its generalization to higher order elements is a pointwise relation between the derivatives of the reference-to-physical transformation of K and the metric. It serves as the definition of high-order unit triangles in the next section. In Appendix A.3.1, we give additional invariants that may be relevant for high-order error estimation in future developments.

Quasi-unit linear meshes are commonly generated by modifying iteratively an existing mesh to comply with the input Riemannian metric. New vertices are inserted along long edges ($\ell > \sqrt{2}$) and short ($\ell < 1/\sqrt{2}$) edges are collapsed by merging their extremities until all edges are quasi-unit. Then, the mesh may be optimized by swapping edges and relocating vertices to improve some quality measure $\in [0, 1]$, which is 1 on the ideal elements. We do not discuss this further, but we refer for instance to Dobrzynski et al. [68] and the references therein for more details regarding the practical construction of quasi-unit linear simplicial meshes.

4.2 Curvilinear unit elements with prescribed Jacobian matrix

On linear triangles, two vertices are connected by a straight line with a linear parameterization, and a definition based only on the edges is satisfactory. On the other hand, vertices of a curved triangle can be connected by arbitrary smooth curves. An edge-based definition alone is not sufficient, since any nontrivially curved triangle with unit edges for a constant metric has a nonconstant Jacobian matrix and violates (4.7). Moreover, curved triangles can have unit edges but be invalid, or have different areas depending on whether its edges curve inwards or outwards. A more restrictive definition is thus needed for curved simplices. Rochery [17, 33] proposed to consider the (4.7) as the definition of unitness, instead of an invariant; we discuss this choice here. Generalizing (4.7) for nonconstant Jacobian yields:

$$J_K(\xi) = \mathcal{M}^{-1/2}(F_K(\xi)) R J_\triangle. \tag{4.14}$$

As R and $J_\triangle$ are constant matrices, J_K and $\mathcal{M}^{-1/2} \circ F_K$ must be the same function of ξ, for instance, polynomials of degree $q-1$ if we consider $\mathcal{P}^q$ triangles. We first introduce the following general definition of curved unit triangle, assuming a smooth transformation:

Definition 4.2. *A **curved triangle** K is **unit** with respect to the metric $\mathcal{M}$ if there exists a rotation R such that for all $\xi \in \widehat{K}$:*

$$J_K(\xi) = \mathcal{M}^{-1/2}(F_K(\xi))RJ_\triangle, \tag{4.15}$$

and if $\mathcal{M}$ is such that F_K is a smooth function of ξ.

It follows from this definition that unit triangles are curved as soon as the metric is not constant. Moreover, unit triangles are valid everywhere since the matrix of $\mathcal{M}$ is positive-definite. We first show that this definition implies Definition 4.1. This is done in the following:

Proposition 4.3. *If a curved triangle K with edges e_i is unit w.r.t. $\mathcal{M}$ in the sense of Definition 4.2, then:*

$$\ell_{\mathcal{M}}(e_i) = 1, \quad \text{for } i = 1, 2, 3. \tag{4.16}$$

Proof. See Appendix A.3.2. The key idea is that the edges of a unit element $K = F(\xi)$ can be parameterized by $e_i(t) = F(\boldsymbol{\xi}(t))$, whose derivative involves the Jacobian matrix $J_F(\xi)$. Since $\mathcal{M}^{-1/2}$ cancels the effect of the metric in the length computations, we can relate the length of e_i to the length of the edges of $K_\triangle$, which are unit for the Euclidean metric.

In addition to being unit length, the edges of curved unit triangles may be geodesic curves, depending on the metric. To see this, we use the same parameterization $\gamma(t) = (x(\boldsymbol{\xi}(t)), y(\boldsymbol{\xi}(t)))$ for the edges as in the proof hereinabove and substitute the velocity and acceleration into the geodesic equation (3.111):

$$\begin{aligned}
\dot{\gamma}^i(t) &= \frac{\partial x^i}{\partial \xi^j}\frac{d\xi^j}{dt} = J_{K,ij}\frac{d\xi^j}{dt} = \mathcal{M}_{ip}^{-1/2}(RJ_\triangle)_{pj}\frac{d\xi^j}{dt}, \\
\ddot{\gamma}^i(t) &= \frac{d\mathcal{M}_{ip}^{-1/2}}{dt}(RJ_\triangle)_{pj}\frac{d\xi^j}{dt} = \frac{\partial \mathcal{M}_{ip}^{-1/2}}{\partial x^m}\dot{\gamma}^m(t)(RJ_\triangle)_{pj}\frac{d\xi^j}{dt} = \frac{\partial \mathcal{M}_{ip}^{-1/2}}{\partial x^m}\dot{\gamma}^m(t)\mathcal{M}_{pj}^{1/2}\dot{\gamma}^j(t)
\end{aligned} \tag{4.17}$$

After reordering the indices in the acceleration, the geodesic equation writes:

$$\begin{aligned}
\ddot{\gamma}^k + \Gamma_{ij}^k\dot{\gamma}^i\dot{\gamma}^j &= 0 \\
&\downarrow \\
\frac{\partial \mathcal{M}_{kp}^{-1/2}}{\partial x^i}\mathcal{M}_{pj}^{1/2}\dot{\gamma}^i\dot{\gamma}^j + \Gamma_{ij}^k\dot{\gamma}^i\dot{\gamma}^j &= 0,
\end{aligned} \tag{4.18}$$

so that, after identifying and multiplying by $\mathcal{M}_{jn}^{-1/2}$, the edges $\gamma(t)$ are geodesic curves iff

the metric satisfies:

$$\frac{\partial \mathcal{M}_{kn}^{-1/2}}{\partial x^i} + \Gamma_{ij}^k \mathcal{M}_{jn}^{-1/2} = \frac{\partial \mathcal{M}_{kn}^{-1/2}}{\partial x^i} + \frac{1}{2}\mathcal{M}_{k\ell}^{-1}\left(\frac{\partial \mathcal{M}_{j\ell}}{\partial x^i} + \frac{\partial \mathcal{M}_{i\ell}}{\partial x^j} - \frac{\partial \mathcal{M}_{ij}}{\partial x^\ell}\right)\mathcal{M}_{jn}^{-1/2} = 0. \quad (4.19)$$

The derivative of $\mathcal{M}^{-1/2}$ requires some precaution, as it is defined from the diagonalized form $\mathcal{M} = P\Lambda P^T$. From Sebastiani [69], defining $\partial_i P \triangleq \partial P/\partial x^i P^T$, it is given by e.g.:

$$\frac{\partial \mathcal{M}_{kn}^{-1/2}}{\partial x^i} = -\frac{1}{2}\mathcal{M}_{kl}^{-1}\left(\frac{\partial \mathcal{M}}{\partial x^i} + 2\mathcal{M}^{1/2}\partial_i P \mathcal{M}^{1/2} - \mathcal{M}\partial_i P + \partial_i P \mathcal{M}\right)_{lj} \mathcal{M}_{jn}^{-1/2}. \quad (4.20)$$

In particular, it differs from the usual derivative of integer matrix power $-\frac{1}{2}\mathcal{M}^{-1}\partial_i \mathcal{M}\mathcal{M}^{-\frac{1}{2}}$ unless the matrices $\mathcal{M}$ and $\partial_i P$ commute. Thus, the edges are geodesic iff:

$$\frac{\partial \mathcal{M}_{il}}{\partial x^j} - \frac{\partial \mathcal{M}_{ij}}{\partial x^\ell} = \left(2\mathcal{M}^{1/2}\partial_i P \mathcal{M}^{1/2} - \mathcal{M}\partial_i P + \partial_i P \mathcal{M}\right)_{lj}. \quad (4.21)$$

The right-hand side only involves derivatives w.r.t. x^i, so the metric should satisfy a relation on the permutations of i, j, l to include the x^i on the left-hand side as well. We were not able to exhibit a more explicit constraint on the metric, but we conjecture that it implies the symmetry constraint (4.24) already discussed in Chapter 3 and recalled below, and that it is also equivalent to cancelling the scalar curvature of $\mathcal{M}$, in which case the transformation F_K is an isometry and maps geodesics to geodesics, as discussed in Section 4.3. Indeed, we have for example for the metric induced by the developable surface $S = (x, y, f(x))$:

$$\mathcal{M}^{-1/2} = \begin{pmatrix} \frac{1}{\sqrt{1+f_x^2}} & 0 \\ 0 & 1 \end{pmatrix}, \quad \frac{\partial \mathcal{M}^{-1/2}}{\partial x} = \begin{pmatrix} \frac{-f_x f_{xx}}{(1+f_x^2)^{3/2}} & 0 \\ 0 & 0 \end{pmatrix}, \quad \frac{\partial \mathcal{M}^{-1/2}}{\partial y} = [0], \quad (4.22)$$

and the Christoffel symbols:

$$\Gamma_{ij}^k = 0 \quad \text{except} \quad \Gamma_{11}^1 = \frac{f_x f_x x}{1+f_x^2} \quad \Rightarrow \quad \frac{\partial \mathcal{M}_{kn}^{-1/2}}{\partial x^i} + \Gamma_{ij}^k \mathcal{M}_{jn}^{-1/2} = 0, \quad (4.23)$$

thus the geodesic equation is satisfied and edges are geodesic curves for this metric. This also holds for a surface $(x, y, f(x - ay))$, where the line $x = ay$ accounts for any rotation applied before f, but not for general nondevelopable surfaces $f(x, y)$.

We focus now on the special case of $\mathcal{P}^q$ triangles, which have a polynomial boundary. From the discussion in Section 3.2.5, the Hessian of a polynomial transformation F_K is symmetric

iff $\mathcal{M}$ verifies:

$$\frac{\partial \mathcal{M}_{ki}^{-1/2}}{\partial x^m}\mathcal{M}_{mj}^{-1/2} = \frac{\partial \mathcal{M}_{kj}^{-1/2}}{\partial x^m}\mathcal{M}_{mi}^{-1/2} \quad \text{for all } \; i,j,k. \tag{4.24}$$

For quadratic triangles, this is the only constraint on $\mathcal{M}$ to ensure the smoothness of F_K, since higher-order derivatives vanish. We can thus define quadratic unit triangles as follows [33]:

Definition 4.3. *Let $\mathcal{M}$ be a Riemannian metric satisfying the symmetry constraint* (4.24) *and such that $\mathcal{M}^{-1/2} \circ F_K$ is linear in ξ. A **quadratic triangle** K is **unit** with respect to $\mathcal{M}$ if there exists a rotation R such that for all $\xi \in \widehat{K}$:*

$$J_K(\boldsymbol{\xi}) = \mathcal{M}^{-1/2}(F_K(\boldsymbol{\xi}))RJ_{\triangle}. \tag{4.25}$$

$\mathcal{P}^q$ triangles can be defined in the same way, adding constraints on the derivatives of the metric to guarantee the continuity of the q-th derivatives of F_K.

The generation of unit elements in the sense of Definition 4.2 is less straightforward than for linear elements, as the pointwise relation between the Jacobian and the metric must be enforced everywhere on the element, instead of an edge-based criterion. The following two sections detail how unit elements can be generated in practice by degree continuation, that is, curving from a unit linear mesh, following Rochery et al. [33], or by minimization of a global distortion measure, following Aparicio-Estrems et al. [25–27].

4.2.1 Construction by degree continuation

As an alternative to Definition 4.3, Rochery et al. [33] proposed in a recent paper to define unit as a relation between displacements in the reference and physical triangles:

Definition 4.4. *Let $\mathcal{M}$ be a Riemannian metric such that $\mathcal{M}^{-1/2} \circ F_K$ is a polynomial of degree $q-1$ in ξ. A **$\mathcal{P}^q$ triangle** K is **unit at** ξ_0 with respect to $\mathcal{M}$ if there exists a rotation R such that for all $\xi \in \widehat{K}$:*

$$F_K(\xi) - F_K(\xi_0) = \mathcal{M}^{-1/2}(F_K(\xi))RJ_{\triangle}(\xi - \xi_0). \tag{4.26}$$

In the relation above, $J_{\triangle}$ is constant and $J_{\triangle}(\xi - \xi_0)$ is a matrix-vector product. Note that the symmetry constraint on $\mathcal{M}$ is absent from this definition. Definition 4.3 describes a unit element by its Jacobian matrix, while Definition 4.4 describes it by a translation-invariant relation between displacements, thus by its transformation F_K. In the following, we present Rochery's proof of equivalence with the previous definition and detail how they decouple

(4.26) into linear and high-order unitness to devise a local edge curving method. We show that the same decoupling can be obtained from (4.25).

Asymptotic equivalence with the definition in Jacobian

For linear triangles, Definitions 4.3 and 4.4 are equivalent: taking the derivative w.r.t. ξ on both sides, we write:

$$J_K(\xi) = \mathcal{M}^{-1/2}(F_K(\xi))RJ_\triangle + \left.\frac{\partial \mathcal{M}^{-1/2}}{\partial x}\right|_{F_K(\xi)} \frac{\partial F_K}{\partial \xi} RJ_\triangle(\xi - \xi_0). \tag{4.27}$$

When $\mathcal{M}^{-1/2}$ is constant, hence for linear triangles, the second term vanishes and this is Definition 4.3. For a general metric, Rochery shows that both definitions are asymptotically equivalent. To see it, consider embedded Riemannian metrics defined, for $h > 0$ and for some $\mathcal{M}_0$, by $\mathcal{M}_h^{-1/2} \triangleq h\mathcal{M}_0^{-1/2}$ (i.e., $\mathcal{M}_h = h^{-2}\mathcal{M}_0$). The principal sizes prescribed by these metrics are the eigenvalues of $\mathcal{M}_h^{-1/2}$, thus they decrease as $\mathcal{O}(h)$ and so do their unit elements through the transformation $F_{K,h}(\xi)$ defined by (4.26) for $\mathcal{M}^{-1/2} = \mathcal{M}_h^{-1/2}$. As $RJ_\triangle$ and $\xi - \xi_0$ are both $\mathcal{O}(1)$ and because the derivatives of $\mathcal{M}$ in the physical space are independent of h, unit elements in the sense of (4.26) for these embedded metrics satisfy:

$$\begin{aligned} J_{K,h}(\xi) &= \underbrace{\mathcal{M}_h^{-1/2}(F_{K,h}(\xi))}_{\mathcal{O}(h\mathcal{M}_0^{-1/2})} RJ_\triangle + \underbrace{\left.\frac{\partial \mathcal{M}_h^{-1/2}}{\partial x}\right|_{F_{K,h}(\xi)}}_{\mathcal{O}(h\mathcal{M}_0^{-1/2})} \underbrace{\frac{\partial F_{K,h}}{\partial \xi}}_{\mathcal{O}(h)} RJ_\triangle(\xi - \xi_0) \\ &\downarrow \\ &= \mathcal{M}_h^{-1/2}(F_{K,h}(\xi))RJ_\triangle + \mathcal{O}(h\mathcal{M}_h^{-1/2}), \end{aligned} \tag{4.28}$$

thus we recover Definition 4.3 as $h \to 0$. The reference coordinate $\xi_0 \in \widehat{K}$ does not appear at convergence, so the element can be taken to be unit with respect to any ξ_0 and the equivalence between the two definitions still holds.

Decoupling

For $\mathcal{P}^2$ triangles, Definition 4.4 splits unitness into a linear part, involving the metric, and a quadratic part, involving its derivatives, as follows. Since F_K is quadratic in ξ and $\mathcal{M}^{-1/2} \circ F_K$ is linear, the quadratic and linear expansion for each polynomials are exact. They write:

$$F_K^i(\xi) = F_K^i(\xi_0) + J_K(\xi_0)_{ij}(\xi - \xi_0)^j + \frac{1}{2}(\xi - \xi_0)^j \frac{\partial F_K^i}{\partial x^j \partial x^k}(\xi_0)(\xi - \xi_0)^k \tag{4.29}$$

For the metric, we have:

$$\mathcal{M}_{ij}^{-1/2}(F_K(\xi)) = \mathcal{M}_{ij}^{-1/2}(F_K(\xi_0)) + \underbrace{\left.\frac{\partial \mathcal{M}_{ij}^{-1/2}}{\partial x^m}\right|_{F_K(\xi_0)} \left.\frac{\partial F_K^m}{\partial \xi^k}\right|_{\xi_0}}_{\triangleq\ T_{ijk}(\xi_0)} (\xi - \xi_0)^k. \tag{4.30}$$

We let $v \triangleq \xi - \xi_0$, and $N \triangleq RJ_\triangle$ as in [33]. Inserting both (4.29) and (4.30) into (4.26) yields:

$$J_K(\xi_0)_{ij} v^j + \frac{1}{2} v^j v^k \, \partial_{jk} F_K^i = \left[\mathcal{M}_{ij}^{-1/2}(F_K(\xi_0)) + T_{ijk}(\xi_0) v^k\right] N_{jn} v^n \tag{4.31}$$

This relation is polynomial in v, thus in ξ, and can be split into linear and quadratic components, which identify to:

$$\begin{aligned} J_K(\xi_0) \times (\xi - \xi_0) &= \mathcal{M}^{-1/2}(F_K(\xi_0)) N \times (\xi - \xi_0), \\ \frac{1}{2}(\xi - \xi_0)^j \, \partial_{jk} F_K^i(\xi_0)(\xi - \xi_0)^k &= \left[T_{ijk}(\xi_0)(\xi - \xi_0)^k\right] \, N \times (\xi - \xi_0). \end{aligned} \tag{4.32}$$

The linear part is exactly the definition of a linear unit triangle. The coefficients of the quadratic part can be identified with a bit more work (see Appendix in [33]) to finally obtain the following relations on the first and second derivatives of F_K depending only on the metric and its derivatives:

$$\begin{cases} J_K(\xi_0) = \mathcal{M}^{-1/2}(F_K(\xi_0)) N, \\ \partial_{ij} F_K^k(\xi_0) = (N_{sj} N_{ti} + N_{si} N_{tj}) \left.\dfrac{\partial \mathcal{M}_{sk}^{-1/2}}{\partial x^m}\right|_{F_K(\xi_0)} \mathcal{M}_{mt}^{-1/2}\left(F_K(\xi_0)\right). \end{cases} \tag{4.33}$$

Thus, a quadratic triangle is unit at $\xi_0 \in \widehat{K}$ in the sense of Definition 4.4 if the linear part of its reference-to-physical transformation agrees with the metric and the quadratic part agrees with its derivatives. When the metric is constant, the Hessian of the transformation vanishes and the unit triangle is linear as expected. As observed in [33] by applying the same embedded metrics argument as previously, the Jacobian of F_K scales as $\mathcal{O}(h)$ whereas the Hessian scales as $\mathcal{O}(h^2)$, thus the edge curvature decreases faster than the edge size. In other words, **unit quadratic meshes are asymptotically straight**, in agreement with Ciarlet and Raviart [52] and the optimal convergence rate of reference-space Lagrange interpolation is maintained as $h \to 0$.

We also obtain the decoupling (4.33) by a slightly easier path, starting from Definition 4.3. The transpose of (4.25) is $J_K^T = (RJ_\triangle)^T \mathcal{M}^{-1/2} \Leftrightarrow J_{K,ji} = N_{ip}^T \mathcal{M}_{pj}^{-1/2}$, and a Taylor expansion

for J_K^T and $\mathcal{M}^{-1/2}$ yields:

$$\begin{aligned} J_{K,ji} + \frac{\partial J_{K,ji}}{\partial \xi^k}(\xi - \xi_0)^k &= N_{ip}^T \left(\mathcal{M}_{pj}^{-1/2} + \frac{\partial \mathcal{M}_{pj}^{-1/2}}{\partial x^m} \frac{\partial F_K^m}{\partial \xi^k}(\xi - \xi_0)^k \right) \\ &= N_{ip}^T \left(\mathcal{M}_{pj}^{-1/2} + \frac{\partial \mathcal{M}_{pj}^{-1/2}}{\partial x^m} \mathcal{M}_{mq}^{-1/2} N_{qk}(\xi - \xi_0)^k \right), \end{aligned} \tag{4.34}$$

where all quantities are evaluated at ξ_0 or $F_K(\xi_0)$. The condition on the Jacobian at ξ_0 is:

$$J_{K,ji} = N_{ip}^T \mathcal{M}_{pj}^{-1/2} \Leftrightarrow J_K^T = N^T \mathcal{M}^{-1/2} \Leftrightarrow J_K = \mathcal{M}^{-1/2} R J_\triangle, \tag{4.35}$$

and here, the condition on the Hessian is readily identified:

$$\frac{\partial J_{K,ji}}{\partial \xi^k} = \partial_{ik} F_K^j = N_{ip}^T \frac{\partial \mathcal{M}_{pj}^{-1/2}}{\partial x^m} \mathcal{M}_{mq}^{-1/2} N_{qk} = N_{pi} N_{qk} \frac{\partial \mathcal{M}_{pj}^{-1/2}}{\partial x^m} \mathcal{M}_{mq}^{-1/2}. \tag{4.36}$$

The Hessian of each component j is symmetrized by adding the transpose with respect to i, k, thus:

$$\partial_{ik} F_K^j = (N_{pi} N_{qk} + N_{pk} N_{qi}) \frac{\partial \mathcal{M}_{pj}^{-1/2}}{\partial x^m} \mathcal{M}_{mq}^{-1/2}. \tag{4.37}$$

which is the same as (4.33) after renaming the dummy $j \leftrightarrow k, p \to s, q \to t$.

Curving

The decoupling (4.33) yields a fast and local (elementwise) edge curving scheme without the need for optimization. The linear constraint can be satisfied by a standard metric-based mesh generator, so that the linear mesh is unit e.g. at the triangles barycenters. Then, the three independent coefficients of the Hessian yield a linear system for the midnode displacements $\boldsymbol{a}_i$ for each component of F_K and the edges are curved in a single step. Because the first and second derivatives of F_K asymptotically differ by an order of magnitude in h, both relations of (4.33) are satisfied up to a negligible remainder even when the linear constraint is satisfied at the barycenters of the linear mesh, instead of those of the quadratic mesh. This means that a single iteration is sufficient to converge both constraints together. Adjacent elements will typically prescribe different curvatures on their shared edges, so these edges are curved only once. The rotation matrix R is necessary to compute $N = RJ_\triangle$ in (4.33): this is done from the adapted linear mesh, for which the Jacobian matrices on the left-hand side of the first relation are known, as is the metric. For the complete computations, we refer to Rochery's paper [33].

4.2.2 Construction by distortion minimization

To curve the edges, Aparicio-Estrems et al. [25, 26] proposed to minimize a global mesh distortion measure described as follows. Let K be a physical triangle and let $F_{\triangle\to K}$ and $J_{\triangle\to K}$ denote the transformation and Jacobian from $K_\triangle$ to K. In dimension n, the **pointwise distortion** of K with respect to $\mathcal{M}$ is defined for $\boldsymbol{y} \in K_\triangle$ by [25]:

$$\eta(\boldsymbol{y}) = \frac{1}{n} \frac{\operatorname{tr}\left(J_{\triangle\to K}^T(\boldsymbol{y})\,[\mathcal{M}_{F_{\triangle\to K}(\boldsymbol{y})}]\,J_{\triangle\to K}(\boldsymbol{y})\right)}{\left[\det\left(J_{\triangle\to K}^T(\boldsymbol{y})\,[\mathcal{M}_{F_{\triangle\to K}(\boldsymbol{y})}]\,J_{\triangle\to K}(\boldsymbol{y})\right)\right]^{1/n}} \geq 1. \tag{4.38}$$

This is the AM-GM inequality for the (strictly positive) eigenvalues of $[F_{\triangle\to K}^* \mathcal{M}]$, the matrix of the pullback metric on $K_\triangle$:

$$\frac{1}{n} \operatorname{tr}\left[F_{\triangle\to K}^* \mathcal{M}\right] \geq \left(\det[F_{\triangle\to K}^* \mathcal{M}]\right)^{1/n}, \tag{4.39}$$

with equality $\eta(\boldsymbol{y}) = 1$ only if $[F_{\triangle\to K}^* \mathcal{M}] = cI$ for some $c > 0$. The pointwise distortion thus measures if K lies on a **conformally flat** manifold, that is, equipped with a metric that is a positive multiple of the Euclidean metric $c\overline{\mathcal{M}}$. In other words, the distortion measures how the interior angles of K deviate from the angles of $K_\triangle$, but do not account for the edge lengths. In the context of moving mesh PDEs (MMPDE), this indicator reaching a minimum is also known as the *alignment condition*, see e.g. Huang [70], and is used in conjunction with a sizing (or *monitor*) function to perform the mesh adaptation. From the pointwise distortion, the **regularized elementwise distortion** is defined by:

$$\eta_K^2 = \frac{1}{|K_\triangle|_{\overline{\mathcal{M}}}} \int_{K_\triangle} \eta^2(\boldsymbol{y})\, d\boldsymbol{y} = \frac{4}{\sqrt{3}} \int_{K_\triangle} \eta^2(\boldsymbol{y})\, d\boldsymbol{y}. \tag{4.40}$$

A $\mathcal{P}^2$ mesh of minimum distortion can be obtained by minimizing the overall mesh distortion with respect to the positions of the edge vertices $\boldsymbol{x}_i$, as it is done in Aparicio-Estrems et al. [25, 26]:

$$\min \mathcal{F}(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n_{\text{edges}}}) = \min \sum_{K\in\mathcal{T}} \eta_K^2 \tag{4.41}$$

Minimizing the distortion on an element amounts to finding a constant rotation R such that $J_{\triangle\to K} = c\mathcal{M}^{-1/2}R$:

$$\eta(\boldsymbol{y}) = \frac{1}{n} \frac{\operatorname{tr}\left(c^2\, R^T \mathcal{M}^{-1/2}[\mathcal{M}]\mathcal{M}^{-1/2} R\right)}{\left[\det\left(c^2\, R^T \mathcal{M}^{-1/2}[\mathcal{M}]\mathcal{M}^{-1/2} R\right)\right]^{1/n}} = \frac{1}{n} \frac{c^2 \operatorname{tr}\,(I)}{\left[c^{2n}\, \det I\right]^{1/n}} = 1. \tag{4.42}$$

This enforces the Jacobian matrix of Definition 4.2 up to a constant, and does not distinguish between two triangles scaled by a factor c. Thus:

Proposition 4.4. *The following affirmations are equivalent:*

1. *The elementwise distortion η_K^2 on K is minimal and equal to 1;*
2. *There is a constant $c > 0$ such that $c^{-1}K$ is unit w.r.t. $\mathcal{M}$ in the sense of Definition 4.2 (and thus also of Definition 4.1).*

Proof. From the discussion above, the AM-GM inequality reaches a minimum if and only if the argument $[F_{\triangle\to K}^* \mathcal{M}] = J_{\triangle\to K}^T [\mathcal{M}_{F_{\triangle\to K}(\boldsymbol{y})}] J_{\triangle\to K}$ is a positive multiple c of the identity matrix. This is the case when $J_{\triangle\to K} = c\mathcal{M}^{-1/2}R$ for some rotation matrix R, as shown in the proof of Proposition 3.1. If K has such a Jacobian, then $c^{-1}K$ is unit in the sense of Definition 4.2.

In Section 4.4, we construct isometric triangulations by minimization of an energy function based on the length of subtriangles, to enforce the isometries locally. The distortion is minimized as result of this stronger constraint on the elements.

4.3 Curvilinear unit elements as Riemannian isometries

Linear unit elements are triangles whose edges have the same length as a regular equilateral triangle. Moreover, Definition 4.2 states that a curved triangle K is unit if its Jacobian matrix satisfies:

$$J_K = J_F J_\triangle = \mathcal{M}^{-1/2} R J_\triangle, \tag{4.43}$$

Having discussed Riemannian isometries in Section 3.2.5, we are naturally led to the following definition of unit triangle:

Definition 4.5. *Let U and V be subsets of $(\mathbb{R}^n, \overline{\mathcal{M}})$ and $(M, \mathcal{M})$, respectively, and let $K_0 \subseteq U$ be a positively oriented reference linear simplex. A **curved simplex** $K \subset V$ is **unit** with respect to $\mathcal{M}$ and K_0 if it is isometric to K_0, that is, if $K = F(K_0)$ for some orientation-preserving isometry $F : U \to V$.*

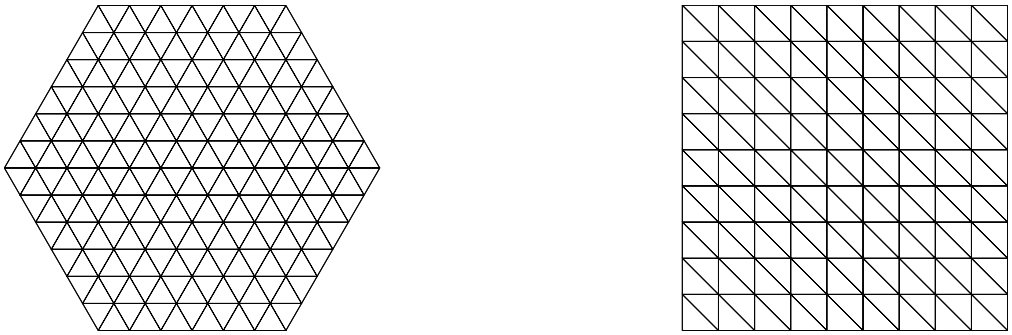
The previous definitions are recovered by setting $K_0 = K_\triangle$ as the reference simplex. Indeed, isometries preserve the lengths, thus any unit simplex has unit edges because $K_\triangle$ is regular, and from Proposition 3.1, the Jacobian matrix of F is $J_F = \mathcal{M}^{-1/2}R$ for some rotation

matrix R, thus $J_K = J_F J_\triangle = \mathcal{M}^{-1/2} R J_\triangle$. Moreover, the invariants shared by linear unit triangles, Proposition 4.2, are direct consequences of Riemannian isometries preserving the inner product and lengths (and thus the angles), the area of compact manifolds, see (3.88), and the energy of curves. We should stress that the isometry F cannot, in general, be described by a closed-form expression. Constant metrics are a notable exception, as F is then linear and given by $F(\boldsymbol{\xi}) = \mathcal{M}^{-1/2} R \boldsymbol{\xi}$, and unit simplices are the image $F(K_0) = \mathcal{M}^{-1/2} R K_0 + \boldsymbol{p}$, as described by Proposition 4.1. Riemannian isometries are a much stronger constraint on the simplices, as the distance preservation is enforced locally, as opposed to the linear definition where only the global edge length is prescribed. As shown hereafter, since the reference simplex K_0 is linear and thus geodesic, this narrows the possible parameterizations of the edges of K to geodesic curves, as they preserve the local distance-minimizing property of the edges of K_0. In the same way that a single metric tensor describes an equivalence class of linear unit elements, a metric field describes an equivalence class of curved unit elements whose equivalence relation is to be isometric. Indeed, if F_1 and F_2 are two distinct isometries whose Jacobian matrix is prescribed by the metric field, their inverse are also isometries, so that two unit elements $K_1 = F_1(R_1 K_0)$ and $K_2 = F_2(R_2 K_0)$ are isometric and related by $K_2 = F_2(R_2 R_1^T F_1^{-1}(K_1))$.

Remark. Here, we chose to define a unit simplex as an isometric transformation of an arbitrary linear simplex, not necessarily equilateral. When the simplex is regular, we recover the usual definition of unitness for linear simplices. We could instead restrict unit simplices to be isometric to a regular simplex only, to convey the idea that the edges of a unit simplex are always of unit length.

To define meshes of isometric simplices, we first consider **simplicial tilings**, or **simplicial tessellations**, of the Euclidean space with a single shape, using both regular and nonregular simplices. These are coverings of $\mathbb{R}^2$ using a single repeated simplex, without any gap or overlap. Tilings made only of regular simplices are called **regular tilings**. All tilings of $\mathbb{R}$ with a single shape are regular. In two dimensions, we consider tilings of either equilateral or right triangles, with only the former being regular. Any two tilings of $\mathbb{R}^2$ with the same shape differ by a rotation: to single out a particular choice, we define the reference tilings $\mathcal{T}_\triangle$ and $\mathcal{T}_\triangleleft$ to be the ones depicted in Fig. 4.3, in which at least one edge of each triangle is aligned with the x-axis.

In each of these reference tilings, two triangles are either a translation of each other, or a rotation of either $\pi/3$ or $\pi/2$ followed by a translation. Although there exists no regular tiling of $\mathbb{R}^3$, we can choose other reference tilings to define unit tetrahedral meshes, see e.g.



$\mathcal{T}_\triangle$ $\mathcal{T}_\triangleleft$

Figure 4.3 Reference tilings of the plane.

Loseille [1]. Then:

Definition 4.6. *A **unit (simplicial) mesh** on an n-manifold is a transformation by an orientation-preserving isometry of a simplicial tiling of $\mathbb{R}^n$.*

Isometries, and thus unit elements and unit meshes, exist iff the manifold M is flat. From Theorem 1, all 1-manifolds are flat, hence isometric to the real line. Thus, leaving aside the problem of the boundaries, a unit mesh exists for any one-dimensional Riemannian metric. 2-manifolds are flat only if the Gaussian curvature K_g vanishes, hence unit meshes exist only for developable surfaces, a very restrictive limitation. If F is an isometry on such 2-manifold and R_0 is any rotation matrix, then for instance a unit mesh is obtained from $\mathcal{T}_\triangle$ by first rotating it by R_0, then applying F. Thus, $F(R_0\mathcal{T}_\triangle)$ is a unit mesh, and as the image of an equilateral tiling, its elements have unit edge length in the deformed space. Since two triangles in $\mathcal{T}_\triangle$ are rotations of 0 or $\pi/3$ of one another, the Jacobian matrices J_K of the triangles in $F(R_0\mathcal{T}_\triangle)$ are given explicitly by $J_K = \mathcal{M}^{-1/2}R_0J_\triangle$ and $J_K = \mathcal{M}^{-1/2}R_0R(\frac{\pi}{3})J_\triangle$. 3-manifolds, which are not explored in this thesis, are flat iff their Ricci curvature tensor vanishes. In the following, we discuss the properties of isometric unit simplices in two dimensions, then we construct examples of isometric and quasi-isometric triangulations on flat and nonflat manifolds.

4.3.1 Properties of isometric unit simplices

The key properties of isometric unit triangles are summarized in the following proposition. They generalize the results presented so far, using the properties of Riemannian isometries.

Proposition 4.5. *Let $K_0 \subset \mathbb{R}^2$ be a positively oriented linear triangle. A unit triangle K on $(M, \mathcal{M})$ with respect to K_0 has the following properties:*

1. *The transformation from the reference simplex $K_\triangleleft$ to K is the composition $F_{\triangleleft\to K} = F(K_0) = F(F_{\triangleleft\to K_0})$. It is not an isometry unless $K_\triangleleft \cong K_0$, that is, unless the linear triangle K_0 is chosen to be isometric to $K_\triangleleft$. Unless the isometry F also happens to be polynomial, which is not the case in general, K is not a $\mathcal{P}^q$ triangle, Fig. 4.4.*

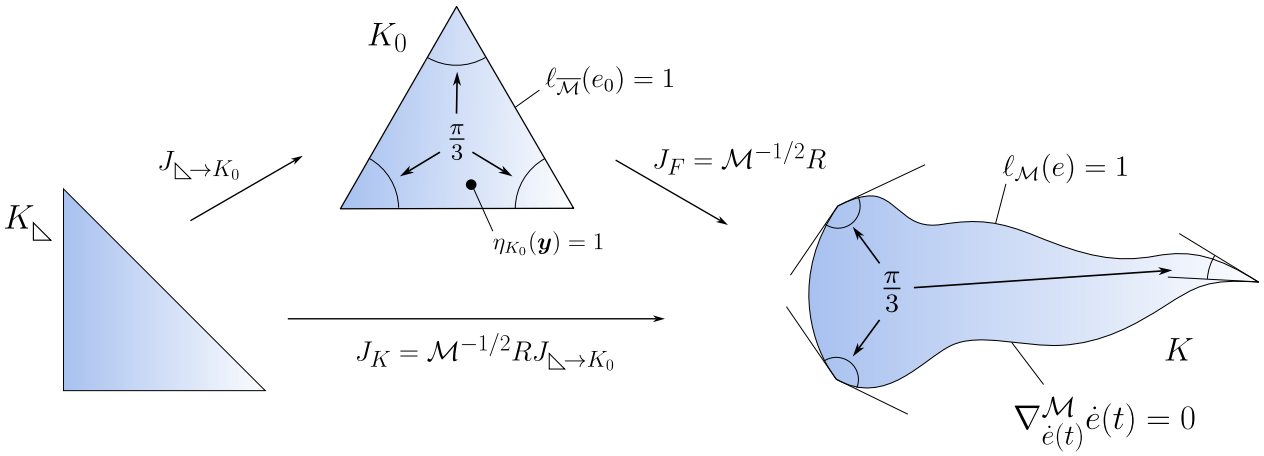


Figure 4.4 An isometric triangle $K = F(K_0)$ with $K_0 = K_\triangle$. The edge lengths and inner angles of K_0 are preserved in K, and the edges $e(t)$ of K are geodesic curves for the Levi-Civita connection of $\mathcal{M}$, noted here $\nabla^{\mathcal{M}}$. The pointwise distortion $\eta_{K_0}(\boldsymbol{y})$, defined on K_0 but characterizing K, is 1 everywhere.

2. *K is positively oriented. In particular, if K is a $\mathcal{P}^q$ triangle, it is valid everywhere.*

3. *The edge lengths and energies, area and interior angles of K_0 are preserved in K.*

4. *K is a geodesic triangle.*

5. *The integral of the enclosed Gaussian curvature in K is zero.*

6. *There is a rotation matrix R such that the Jacobian matrix of $F_{\triangleleft\to K}$ writes:*

$$J_{\triangleleft\to K} = J_{K_0\to K}J_{\triangleleft\to K_0} = J_F J_{\triangleleft\to K_0} = \mathcal{M}^{-1/2}RJ_{\triangleleft\to K_0}. \tag{4.44}$$

7. *Let $\eta_{K_0}(\boldsymbol{y})$ be the pointwise distortion of K with respect to K_0 defined for $\boldsymbol{y} \in K_0$ by:*

$$\eta_{K_0}(\boldsymbol{y}) = \frac{1}{n}\frac{\operatorname{tr}\left(J^T_{K_0\to K}(\boldsymbol{y})\left[\mathcal{M}_{F(\boldsymbol{y})}\right]J_{K_0\to K}(\boldsymbol{y})\right)}{\left[\det\left(J^T_{K_0\to K}(\boldsymbol{y})\left[\mathcal{M}_{F(\boldsymbol{y})}\right]J_{K_0\to K}(\boldsymbol{y})\right)\right]^{1/n}} \geq 1. \tag{4.45}$$

Then $\eta_{K_0}(\boldsymbol{y}) = 1$ for all $\boldsymbol{y} \in K_0$.

Proof.

1. This is straightforward, but it emphasizes the difference between the isometry F, which is not polynomial in general, and the polynomial transformation F_{K^q}, which is usually not an isometry.

2. The reference triangle K_0 is positively oriented. As the isometry F in Definition 4.5 preserves the orientation, K is also positively oriented. A $\mathcal{P}^q$ triangle is valid iff it is positively oriented, thus K is valid if it happens to be a $\mathcal{P}^q$ triangle.

3. This is a direct consequence of each quantity being preserved by isometries.

4. The edges of K_0 are straight lines, which are geodesics in $\mathbb{R}^2$ with the Euclidean metric. Since K is the image of K_0 by an isometry, which by Proposition 3.2 maps geodesics to geodesics, its edges are geodesics in $(M, \mathcal{M})$.

5. This is an immediate application of the Gauss-Bonnet theorem, a major result of differential geometry. For the specific case of geodesic triangles, it relates the enclosed Gaussian curvature K_g to the sum of interior angles α, β, γ as:

$$\int_K K_g \, dA = \alpha + \beta + \gamma - \pi. \tag{4.46}$$

   The right-hand side sums to zero, as the interior angles of K_0 sum to π and they are preserved by isometry.

6. The transformation $F_{\triangle \to K}$ is the composition of the isometry F with linear transformation $F_{K_\triangle \to K_0}$. The Jacobian matrix $J_{\triangle \to K}$ follows from the composition, with $J_{\triangle \to K_0}$ constant, and it was already shown in Proposition 3.1 that $J_F = \mathcal{M}^{-1/2} R$ for some R.

7. The matrix in the definition of $\eta_{K_0}(\boldsymbol{y})$ is the pullback of $\mathcal{M}$ on $\mathbb{R}^2$, which is the identity matrix I as F preserves the metric. The distortion of I is equal to 1, as discussed in Section 4.2.2. □

These properties follow from the local length preservation of Riemannian isometries, which in turn preserves the interior angles, global edge lengths, distortion and geodesic property of curves. Let us first emphasize that as images by isometries, unit simplices are trivially geodesic: this has been exploited in recent work on curvilinear meshing, but never established. In practice, the edges of $\mathcal{P}^2$ triangles are commonly curved by minimizing their length w.r.t. $\mathcal{M}$, approximating the geodesics, but the optimal edge vertex does not have to lie on the

geodesic. This is discussed in Section 4.3.2 further. The third and fourth properties show that if K is unit with respect to $K_0 = K_\triangle$, then it is a geodesic simplex with unit edges. The converse is true only on flat manifolds, for which we have more generally:

Proposition 4.6. *Let $(M, \mathcal{M})$ be a flat manifold and let $K_0 \subset \mathbb{R}^2$ be a linear triangle. A curved simplex $K \subset M$ is isometric to K_0 if and only if it is a geodesic simplex whose homologous edges have the same length as those of K_0.*

Proof. The $\Rightarrow$ direction is simply applying the properties of an isometry, which preserves the lengths and geodesics. To prove the $\Leftarrow$ direction, we proceed as follows. Since M is flat, there exist an isometry $F : \mathbb{R}^2 \to M$ and a triangle K_0' in $\mathbb{R}^2$ such that $K = F(K_0')$. The lengths of the edges of K_0' are the same as those of K, and thus of K_0. By the side-side-side theorem for triangles in the Euclidean space, K_0' and K_0 are isometric, hence K and K_0 are isometric by composing this isometry with F. □

This does not hold on nonflat manifolds, as any nonzero Gaussian curvature prevents the existence of isometries. Indeed, suppose K is a geodesic triangle whose interior angles sum to π and whose homologous edges have the same length as those of a reference linear triangle $K_0 \subset \mathbb{R}^2$. The Gauss-Bonnet formula (4.46) only ensures that the average enclosed curvature in K is zero, however K cannot be isometric to K_0 as soon as $K_g \neq 0$ anywhere in K, because then the underlying manifold itself is not isometric to any subset of $\mathbb{R}^2$. In other words, the metric can behave arbitrarily inside K as long as its curvature sums to zero, but only a flat metric yields an isometry between K and K_0. This is depicted in Fig. 4.5: the bump in the right figure has the same amount of positive and negative Gaussian curvature, thus its integral vanishes but $K_g \neq 0$ prevents the existence of an isometry with the plane.

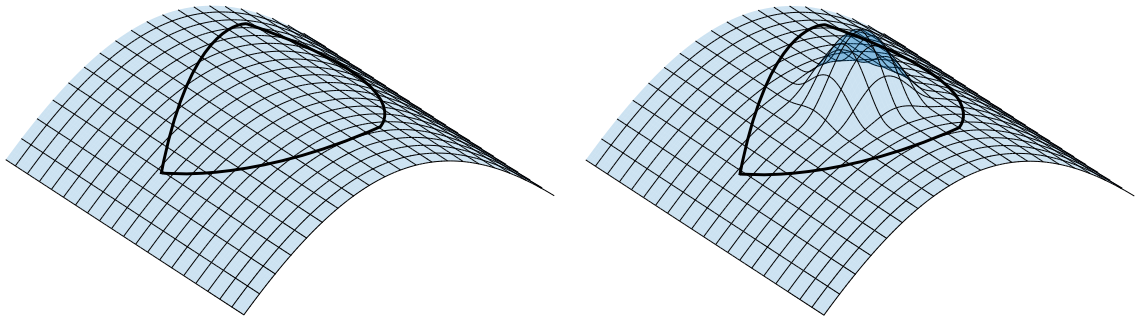


Figure 4.5 Geodesic triangles enclosing a region with vanishing integral of Gaussian curvature. The left triangle is isometric to some $K_0 \subset \mathbb{R}^2$, but not the right one.

From Section 4.1, a metric tensor $\mathcal{M}_{\boldsymbol{p}}$ describes an equivalence class of unit linear elements,

and a linear element is unit for a unique metric tensor. Similarly, a metric *field* with zero scalar curvature describes an equivalence class of isometric triangles and triangulations. In particular, a constant metric field has no curvature and yields the isometric linear triangles of Section 4.1. Because the Jacobian matrix controls $\mathcal{M}^{-1/2}$ everywhere given a rotation R, a curvilinear mesh is isometric to a regular tiling for a unique metric field.

Since any linear simplex K_0 spans an equivalence class of isometric simplices, unit elements can be defined from nonregular simplices, for instance $K_\triangleleft$. This is the idea behind metric-orthogonal mesh generation, proposed by Loseille [42,43] and applied e.g. in Zhang et al. [21] and our previous work [41]. Similarly to Proposition 4.2, triangles isometric to $K_0 = K_\triangleleft$ share the following invariants:

1. The length of their edges are any permutation of $(1, \sqrt{2}, 1)$ and their interior angles are the same permutation of $\left(\frac{\pi}{4}, \frac{\pi}{4}, \frac{\pi}{2}\right)$.
2. Their area is $|K|_{\mathcal{M}} = 1/2$.
3. Let $\mathcal{H}$ be a Riemannian metric represented by the matrix H and let R be the rotation matrix such that $K = F(RK_\triangleleft)$. The energy of the perimeter of K with respect to $\mathcal{H}$ writes:
$$\sum_{i=1}^{3} e_i^T H e_i = 2\,(a - b + c), \tag{4.47}$$
where a, b, c are the coefficients of $R^T \mathcal{M}^{-1/2} H \mathcal{M}^{-1/2} R$. Unlike (4.6) and because $K_\triangleleft$ is not isotropic, this energy depends on the rotation R.
4. The Jacobian matrix of their transformation simply writes $J_{\triangleleft \to K} = \mathcal{M}^{-1/2} R$.

4.3.2 Quasi-unit elements

Definition 4.5 is elegant and provides a broader framework for unit simplices, but is overly restrictive, as flat metrics are an insignificant fraction of the Riemannian metrics at play when performing mesh adaptation for real-life applications. For the practical generation of unit curvilinear meshes, a generalization of quasi-unitness is needed. As ideal elements should be isometric to a reference triangle, a certain number of relaxed properties from Proposition 4.5 should be satisfied by quasi-unit elements. Quasi-unit linear elements, in particular, preserve the relaxed global edge lengths as well as the volume, to avoid the creation of sliver elements whose volume tends to zero. The inner angles or the geodesicity of the edges, which

are local properties, are not controlled. In addition to the edge length and volume, quasi-unit curvilinear elements should preserve relaxed local properties as well, such as the inner angles, the distortion or have geodesic edges. In the following, we propose two definitions of quasi-unit curvilinear elements: a stronger one in terms of the distortion-based quality, and a weaker one in terms of geodesic edges with controlled inner angles.

In Loseille [1,8], the sliverness is controlled by a quality function, which can be generalized to curvilinear simplices by the **elementwise distortion-based quality** defined in dimension n by:

$$Q_{K_0} \triangleq \left(\frac{1}{|K_0|_{\overline{\mathcal{M}}}} \int_{K_0} \eta_{K_0}^n(\boldsymbol{y})\, d\boldsymbol{y} \right)^{-1/n}. \tag{4.48}$$

For a constant metric and $K_0 = K_\triangle$, this reduces to the quality function of Loseille. Indeed, $\det\left(J_{\triangle\to K}^T\left[\mathcal{M}\right] J_{\triangle\to K}\right) = (\det J_{\triangle\to K})^2 \det \mathcal{M}$ and from the preservation of the energy of the linear edges of $K_\triangle$ with respect to $\mathcal{M}$:

$$\sum_{i=1}^{N_e} \ell_{\mathcal{M}}(e_i)^2 = \sum_{i=1}^{N_e} e_i^T \mathcal{M} e_i = \sum_{i=1}^{N_e} E_i^T J_{\triangle\to K}^T \mathcal{M} J_{\triangle\to K}^T e_i = c_n \text{ tr } \left(J_{\triangle\to K}^T\left[\mathcal{M}\right] J_{\triangle\to K}\right), \tag{4.49}$$

with $c_n = 3/2$ in two dimensions and 2 in three dimensions. For linear simplices, the distortion is constant on K_0 and we have, letting here $J = J_{\triangle\to K}$:

$$\int_{K_0} \eta_{K_0}^n\, d\boldsymbol{y} = \eta_{K_0}^n |K_0|_{\overline{\mathcal{M}}} = \frac{\text{tr}^n(J^T[\mathcal{M}]J)}{n^n\,(\det J)^2 \det \mathcal{M}} \frac{|K_0|_{\overline{\mathcal{M}}}^3}{|K_0|_{\overline{\mathcal{M}}}^2} = \frac{\text{tr}^n(J^T[\mathcal{M}]J)}{n^n |K|_{\mathcal{M}}^2} |K_0|_{\overline{\mathcal{M}}}^3, \tag{4.50}$$

thus:

$$Q_{K_0} = n|K_0|_{\overline{\mathcal{M}}}^{-2/n} \frac{|K|_{\mathcal{M}}^{2/n}}{\text{tr}(J^T[\mathcal{M}]J)} = n\, c_n |K_0|_{\overline{\mathcal{M}}}^{-2/n} \frac{|K|_{\mathcal{M}}^{2/n}}{\sum_{i=1}^{N_e} \ell_{\mathcal{M}}(e_i)^2} \in [0,1]. \tag{4.51}$$

For $K_0 = K_\triangle$, this is the quality measure proposed by Loseille, as

$$n\, c_n |K_\triangle|_{\overline{\mathcal{M}}}^{-2/n} = \frac{12}{\sqrt{3}} \quad \text{for} \quad n = 2 \quad \text{and} \quad \frac{36}{\sqrt[3]{3}} \quad \text{for} \quad n = 3. \tag{4.52}$$

Since the distortion measures how an element deviates from a conformal transformation of K_0, this quality controls in particular the inner angles. Quasi-unit simplices can then be extended as follows:

Definition 4.7. *(QU1) A* ***curved simplex*** *K is* ***quasi-unit*** *with respect to $\mathcal{M}$ and K_0 if*

- *its quality Q_{K_0} lies within $[a,1]$ for $0 < a < 1$;*

- *each edge e of K homologous to the edge e_0 of K_0 satisfies:*

$$\frac{1}{\sqrt{2}}\ell_{\overline{\mathcal{M}}}(e_0) \leq \ell_{\mathcal{M}}(e) \leq \sqrt{2}\ell_{\overline{\mathcal{M}}}(e_0). \tag{4.53}$$

This definition controls the distortion, thus the inner angles, and the edge lengths. If the metric is curvature-free, then the edges are geodesics, as discussed in Section 4.2. In [1, 8], the constant a is set to 0.8 to allow for a variety of families of tetrahedra to be considered for three-dimensional meshing. A map F that satisfies (4.53) everywhere in K is called a **quasi-isometry**, so it is natural to use this criterion to define quasi-unitness. Alternatively, quasi-unitness can be tackled through the geodesic property of the edges. By the Gauss-Bonnet formula (4.46), geodesic triangles only preserve the sum of the interior angles when the enclosed Gaussian curvature of the metric cancels out, which is not expected to be the case in general. We propose to preserve individually the relaxed angles as follows[1]:

Definition 4.8. *(QU2) A **curved simplex** K is **quasi-unit** with respect to $\mathcal{M}$ and K_0 if*

- *it is a geodesic simplex;*
- *each angle θ homologous to the angle θ_0 of K_0 satisfies $\theta_0 - c \leq \theta \leq \theta_0 + c$;*
- *its homologous edges with respect to K_0 satisfy* (4.53)*.*

In this definition, c is an angle increment which allows to generate quasi-unit simplices in practice. The triangulations obtained in Chapter 7 indicate that the vast majority of angles lies in $60° \pm 30°$ for $K_0 = K_\triangle$. Imposing the inner angles is weaker than the distortion and being a geodesic is a local property, thus *QU2* is weaker than *QU1* in general. For instance, consider again the geodesic triangle on the right of Fig. 4.5. If the homologous angles are individually preserved, then it is quasi-unit in the sense of *QU2* despite not being unit as it encloses some nonzero curvature. It can be considered a false positive. In comparison, the bump yields strictly positive distortion whose integral does not vanish, so this triangle may not be quasi-unit in the sense of *QU1*, depending on the amplitude of the bump. In

[1]It is worth noticing that unlike QU2, the edges of simplices satisfying *QU1* are not geodesics in general, thus their edge lengths are controlled, but not the Riemannian distance between their vertices:

$$\begin{aligned} QU1:&\ \frac{1}{\sqrt{2}}\ell_{\overline{\mathcal{M}}}(e_0) = d(\boldsymbol{y}_i, \boldsymbol{y}_j) \leq \ell_{\mathcal{M}}(e) \neq d(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq \sqrt{2}\ell_{\overline{\mathcal{M}}}(e_0), \\ QU2:&\ \frac{1}{\sqrt{2}}\ell_{\overline{\mathcal{M}}}(e_0) = d(\boldsymbol{y}_i, \boldsymbol{y}_j) \leq \ell_{\mathcal{M}}(e) = d(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq \sqrt{2}\ell_{\overline{\mathcal{M}}}(e_0). \end{aligned} \tag{4.54}$$

Since the reference metric is Euclidean and the simplex K_0 is straight, the reference edge are always geodesic, so the lengths $\ell_{\overline{\mathcal{M}}}(e_0)$ are always the distances between the reference vertices $\boldsymbol{y}_i, \boldsymbol{y}_j$.

practice, however, *QU1* requires computing integrals and is costlier to enforce; the metric is also usually graded, limiting its variations and curvature, so that *QU2* is expected to be fast to generate while being a satisfying indicator of unitness.

In standard anisotropic mesh adaptation, all straight simplices are also $\mathcal{P}^1$ simplices, that is, all straight lines are represented by a linear parameterization, and there is no distinction between a simplex and a polynomial simplex. For curvilinear simplices, the geodesics do not usually have a polynomial parameterization, so there should be a distinction between quasi-unit simplices and quasi-unit $\mathcal{P}^q$ simplices. The definitions above can be adapted for $\mathcal{P}^q$ simplices as follows:

Definition 4.9. *(QU1p) A $\mathcal{P}^q$* ***simplex*** *K is* ***quasi-unit*** *with respect to $\mathcal{M}$ and K_0 if its quality Q_{K_0} is in $[a, 1]$ and if homologous edges e and e_0 satisfy* (4.53)*.*

Definition 4.10. *(QU2p) A $\mathcal{P}^q$* ***simplex*** *K is* ***quasi-unit*** *with respect to $\mathcal{M}$ and K_0 if*

- *it is valid;*
- *its edges approximate the geodesics, thus are of minimial length;*
- *homologous inner angles θ and θ_0 satisfy $\theta_0 - c \leq \theta \leq \theta_0 + c$;*
- *homologous edges e and e_0 satisfy* (4.53)*.*

QU1p is the immediate polynomial version of *QU1*, whereas *QU2p* approximates the geodesics with polynomial curves. Note that *QU1p* does not require enforcing validity, as it is necessary to have a positive quality. These definitions are used in Chapter 7 to generate quasi-unit curvilinear triangulations.

4.4 Construction of quasi-isometric triangulations

We now illustrate our new definition of unit element and create isometric curvilinear triangulations. Riemannian isometries are a much stronger constraint on the mesh, as distance preservation must be enforced locally, as opposed to the linear case where only the global edge length is prescribed. The local size is controlled by dividing each triangle into a subgrid of N^2 equilateral triangles. In a isometric triangle $K = F(K_0)$, each subtriangle k is also isometric to a subtriangle of K_0, hence its target edge lengths and area with respect to the metric are

$$\ell_{\text{target}} = \frac{1}{N} \quad \text{and} \quad |k|_{\text{target}} = \frac{|K_\triangle|_{\overline{\mathcal{M}}}}{N^2} = \frac{\sqrt{3}}{4N^2} \quad \text{for} \quad K_0 = K_\triangle, \tag{4.55}$$

and

$$\ell_{\text{target}} = \frac{1}{N} \quad \text{or} \quad \frac{\sqrt{2}}{N} \quad \text{and} \quad |k|_{\text{target}} = \frac{|K_{\triangleright}|_{\overline{\mathcal{M}}}}{N^2} = \frac{1}{2N^2} \quad \text{for} \quad K_0 = K_{\triangleright}. \tag{4.56}$$

As a proof of concept, the curved unit mesh is obtained by direct minimization of a cost function which measures the discrepancy between the ideal length and the length of the subedges. This ensures that the optimized triangles are both locally and globally isometric to K_0. We start from a uniform mesh and perform r-adaptation to deform the mesh at a fixed connectivity. Since the topology is unchanged during the optimization, the starting mesh should have the same connectivity as a regular tiling, thus 6 triangles per vertex for all interior vertices. The chosen cost function has the same form as the one in Chapter 2 and is similar to Toulorge et al. [16]:

$$E = \mathcal{E} + \mu \frac{\mathcal{E}_0}{\mathcal{F}_0} \mathcal{F}, \tag{4.57}$$

where the zero subscripts denote quantities evaluated on the initial mesh and μ scales down the penalization term, for instance $\mu = 10^{-2}$. The term $\mathcal{E}$ is an energy associated to the edges. It measures the discrepancy between ℓ_{target} and the metric-weighted length of the edges of each subtriangle. If the chosen reference tiling is $\mathcal{T}_{\triangle}$, the target length is the same for all subedges and the energy writes:

$$\mathcal{E}_{\triangle}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_v}) = \frac{1}{2} \sum_{\text{edges } \boldsymbol{e}} \left(\ell^2_{\mathcal{M}}(\boldsymbol{e}) - \frac{1}{N^2} \right)^2 \tag{4.58}$$

For $\mathcal{T}_{\triangleright}$, the target length depends on whether the edge is mapped from one of the edges adjacent to the right angle $\boldsymbol{e}_{\text{right}}$ or from the hypotenuse $\boldsymbol{e}_{\text{hyp}}$:

$$\mathcal{E}_{\triangleright}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_v}) = \frac{1}{2} \left(\sum_{\boldsymbol{e}_{\text{right}}} \left(\ell^2_{\mathcal{M}}(\boldsymbol{e}_{\text{right}}) - \frac{1}{N^2} \right)^2 + \sum_{\boldsymbol{e}_{\text{hyp}}} \left(\ell^2_{\mathcal{M}}(\boldsymbol{e}_{\text{hyp}}) - \frac{2}{N^2} \right)^2 \right). \tag{4.59}$$

This is summed up into the generic energy:

$$\begin{aligned} \mathcal{E}_{\star}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_v}) &= \frac{1}{2} \sum_{\boldsymbol{e}_{\star}} \left(\ell^2_{\mathcal{M}}(\boldsymbol{e}_{\star}) - \ell^2_{\star,\text{target}} \right)^2 \\ &= \frac{1}{2} \sum_{\boldsymbol{e}_{\star,ij}} \left((\boldsymbol{x}_j - \boldsymbol{x}_i)^T \mathcal{M}_{\boldsymbol{x}_{ij}} (\boldsymbol{x}_j - \boldsymbol{x}_i) - \ell^2_{\star,\text{target}} \right)^2, \end{aligned} \tag{4.60}$$

with $\boldsymbol{x}_{ij} = (\boldsymbol{x}_i + \boldsymbol{x}_j)/2$ and where $\star$ refers to either a regular edge, a right angle adjacent edge or a hypotenuse. The edge length is approximated by evaluating the metric at the middle of the edge, which is satisfactory if the subtriangles are fine enough.

The term $\mathcal{F}$ controls the area of the subtriangles. These should have an optimized metric-weighted area of $|k|_{\text{target}}$, while also remaining valid throughout the optimization. Since the subtriangles are linear, validity is ensured by enforcing that their area remains strictly positive. As in [16], we use a log-barrier [53] to penalize the very small subtriangles and a quadratic function to penalize the very large ones:

$$\mathcal{F}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_v}) = \sum_{K \in \mathcal{T}} \sum_{k \in K} F_\epsilon \left(\frac{|k|_{\mathcal{M}}}{|k|_{\text{target}}} \right), \tag{4.61}$$

where the area of a subtriangle writes:

$$|k|_{\mathcal{M}} = \int_{\widehat{K}} \sqrt{\det \mathcal{M}(\boldsymbol{x}(\boldsymbol{x}_i, \boldsymbol{\xi}))} \, (\det J_F(\boldsymbol{x}_i)) \, d\boldsymbol{\xi}, \tag{4.62}$$

and is evaluated by a 3-points Gauss-Legendre quadrature. The function F_ϵ is defined by:

$$F_\epsilon \left(\frac{|k|_{\mathcal{M}}}{|k|_{\text{target}}} \right) = \left[\ln \left(\frac{|k|_{\mathcal{M}} - \epsilon |k|_{\text{target}}}{|k|_{\text{target}} - \epsilon |k|_{\text{target}}} \right) \right]^2 + \left(\frac{|k|_{\mathcal{M}}}{|k|_{\text{target}}} - 1 \right)^2. \tag{4.63}$$

The behaviour of F_ϵ is shown in Fig. 4.6: it vanishes when $|k|_{\mathcal{M}} = |k|_{\text{target}}$, blows up when $|k|_{\mathcal{M}} \to \epsilon$ and increases as $|k|_{\mathcal{M}} \to \infty$. The parameter ϵ sets the position of the barrier and allows for a dynamic control of $\min |k|_{\mathcal{M}}$ to speed up the optimization. The first log-barrier is set with $\epsilon = 0$ to guarantee that all subtriangles remain valid. This is an adequate first value, as the starting linear mesh is valid. After the first optimization is completed, the ϵ is increased to $\min(0.99, 0.99 \times |k|_{\mathcal{M}}/|k|_{\text{target}})$, pushing the minimum area towards 1 during the next optimization. The log-barrier is moved this way 5 times. The overall cost

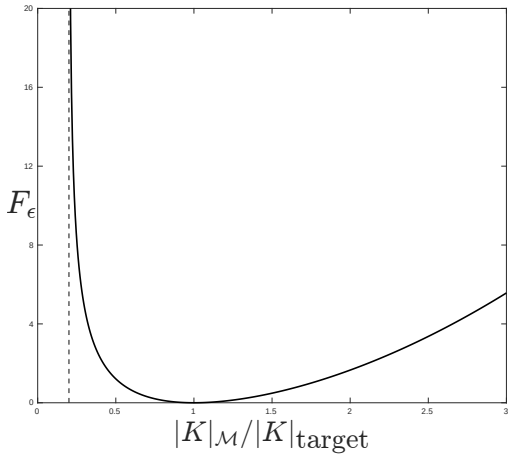


Figure 4.6 Log-barrier function for $\epsilon = 0.2$.

function E is smooth if the metric field is smooth, thus its gradient $\nabla E = \partial E / \partial \boldsymbol{x}_i$ with

respect to the position of the vertices of the subtriangles is available. The cost function is nonlinear and nonconvex, however. In addition, it may have infinitely many global minima, as discussed below. To minimize E, we use the L-BFGS algorithm implemented in the unconstrained minimization library Ceres [54], providing the exact gradient ∇E. This approach does not require the Hessian matrix of E, as it stores and updates instead a low-rank approximation of its inverse. For each of the examples hereafter, we minimize E over the positions $\boldsymbol{x}_i$ of the vertices of the subtriangles. The minimization terminates when either the relative decrease in the cost function, the maximum norm of its gradient or the change in the problem parameters $\|\Delta \boldsymbol{x}_i\|$ passes below the default threshold of Ceres, which are $10^{-6}, 10^{-10}$ and 10^{-8} respectively. To assess the quality of the optimized triangulations, we compare the edge lengths with those of K_0, and measure the elementwise distortion $\eta_{K_0}(\boldsymbol{y})$ given by (4.45), which should be 1 on perfectly isometric triangulations. Since $\eta_{K_0} \in [1, \infty)$, we show instead the distortion-based quality $q_{K_0} = \eta_{K_0}^{-1} \in [0, 1]$, as suggested in Aparicio-Estrems et al. [25, 26]. Lastly, for each choice of Riemannian metric, we superimpose to the curved edges an approximation of the geodesics connecting the mesh vertices.

The Riemannian metrics used in the following examples are all induced by a surface of $\mathbb{R}^3$. The meshes obtained in the parameter space $U \subset \mathbb{R}^2$ appear as if a piece of fabric was laid upon each surface. Because they are the pullback of the Euclidean metric of $\mathbb{R}^3$ on these surfaces, the characteristic lengths are 1 in both the x and y-directions before the metric is pulled once again to the parameter space, in which they exhibit anisotropy induced by the shape of the surface. Thus, in this section, we do not consider the control of mesh size according to the polynomial degree of some interpolant or interpolation error, but, rather, exhibit unit and quasi-unit triangulations for a class of metric that are induced by a graph. The geodesic triangles follow the features of the metric until unit length and are the ideal edge parameterization with potentially multiple inflexion points, whose $\mathcal{P}^q$ approximation may require polynomials of (very) higher order, see for instance the wavy surface S_6 hereafter. In particular, $\mathcal{P}^2$ triangles are not expected to provide a good approximation for all of these examples.

4.4.1 Developable surfaces

We first consider developable surfaces in $\mathbb{R}^3$. As these manifolds have no Gaussian curvature, they are isometric to the Euclidean space and can be covered by triangulations which are isometric to any tiling of the Euclidean space. We consider the three surfaces $S_i = (x, y, f_i(x))$ obtained by extruding the graph of $f_1 = x$, $f_2 = x^2$ and $f_3 = \exp(-10x^2)$ along the y-axis,

Fig. 4.7. The optimization moves the vertices of a starting uniform mesh in $\mathbb{R}^2$ until it matches the projection of a regular tiling of S_i onto the parameter space $(x, y) \in U \subset \mathbb{R}^2$. As a result, the optimized meshes appear distorted in the Euclidean space. On optimal triangulations, the cost function E is exactly zero, and any two of these triangulations differ by the choice of the initial rotation of the regular tiling. There is thus an infinity of global minima for E, as each rotation yields an isometric mesh. The rotation can be imposed by fixing an edge or subedge of the initial mesh, but it requires knowing beforehand the location of a straight edge. In our tests, all vertices are free to move, so that the rotation is decided by the initial conditions given to the optimizer. In particular, different rotations of the initial mesh yield unit triangulations that differ by the rotation applied to the reference tiling before applying the map F.

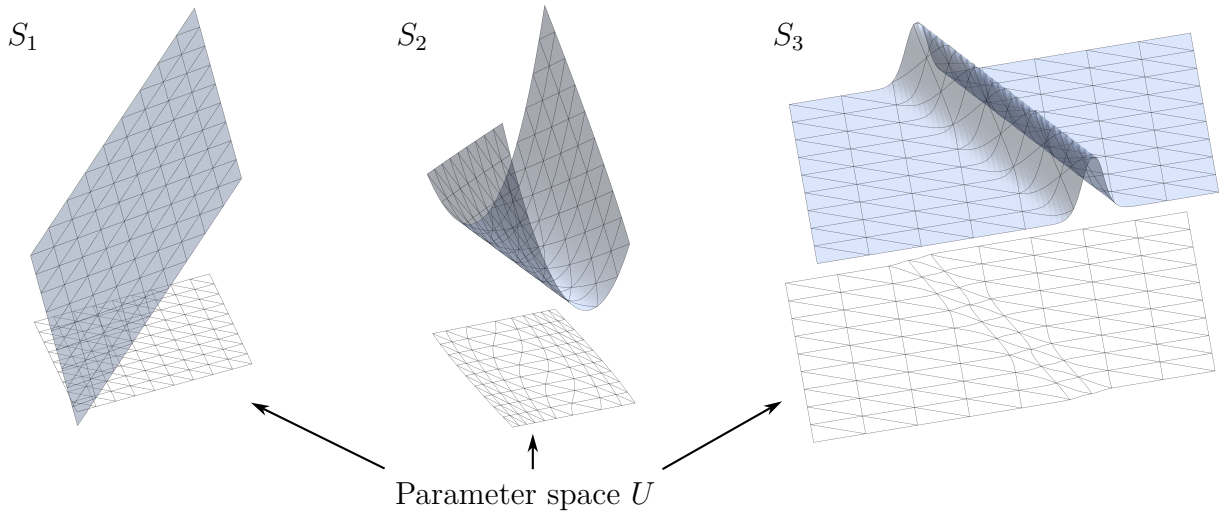


Figure 4.7 Developable surfaces in $\mathbb{R}^3$ and their parameter space $(x, y) \in U$.

The Riemannian metric of interest is the induced metric on each surface, pulled back to the two-dimensional parameter space (x, y) according to (4.64). It simply writes:

$$[\mathcal{M}] = \begin{pmatrix} 1 + f_x^2 & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.64}$$

The optimized triangulations are shown in Figs. 4.8 and 4.9 and are optimized with respect to $\mathcal{T}_\triangle$ and $\mathcal{T}_\diagdown$ respectively. These triangulations are everywhere isometric to their reference tiling, and are thus perfectly unit curvilinear meshes in the sense of all three Definitions 4.1, 4.2 and 4.5. For each surface, optimal triangulations $\mathcal{T}_j$ obtained from three different rotations R_j of the initial mesh are shown. They are not rotations of one another, as each of them

satisfies either $\mathcal{T}_j = F(R_j\mathcal{T}_\triangle)$ or $\mathcal{T}_j = F(R_j\mathcal{T}_\triangleleft)$, where F is the Riemannian isometry between $\mathbb{R}^2$ and the considered surface. Instead, they are related by e.g. $\mathcal{T}_2 = F(R_2R_1^TF^{-1}(\mathcal{T}_1))$.

The metric associated to S_1 is constant, so the unit mesh is linear, as observed on top of Figs. 4.8 and 4.9. For this particular surface, the isometry F can be written explicitly as $F : \mathbb{R}^2 \to U : \boldsymbol{y} \mapsto \mathcal{M}^{-1/2}R\boldsymbol{y}$, whereas it requires solving a nonlinear ODE in f in general. The lengths and areas with respect to this constant metric are represented exactly by the quadratures, so the cost function can be minimized almost to the machine precision (10^{-14} and 10^{-12} with respect to $\mathcal{T}_\triangle$ and $\mathcal{T}_\triangleleft$ respectively). Similarly, the metric of S_3 becomes the identity away from 0 as $\exp \to 0$ and the mesh quickly becomes linear. Whenever the metric varies, the isometric triangulations are curved: they can be viewed as straight grids drawn on the surfaces first, then projected into the parameter space.

The geodesics connecting the mesh vertices are drawn in blue and superimposed to the curved edges. They perfectly match the edges for all triangulations, in agreement with Proposition 4.5 as the elements are geodesic triangles. The edges are the intersections of three geodesic grids that are parallel transported across the surface: this is trivial on linear meshes, but it can be appreciated on the meshes of S_2 and S_3. In these grids, the angles (measured with respect to the local metric) at which two sets of parallel geodesics intersect is either $\pi/3$ when the reference tiling is $\mathcal{T}_\triangle$ or $\pi/2$ and $\pi/4$ when the tiling is isometric to $\mathcal{T}_\triangleleft$. Following Proposition 4.5, the distortion (not shown) is minimal and equal to 1 on all linear subtriangles, and the curved edges are isometric to the edges of the reference triangle, Fig. 4.10. Together, these two criteria ensure that the isometries are enforced globally (edge lengths) and locally (distortion + edge lengths).

Isometric $\mathcal{P}^2$ triangulations

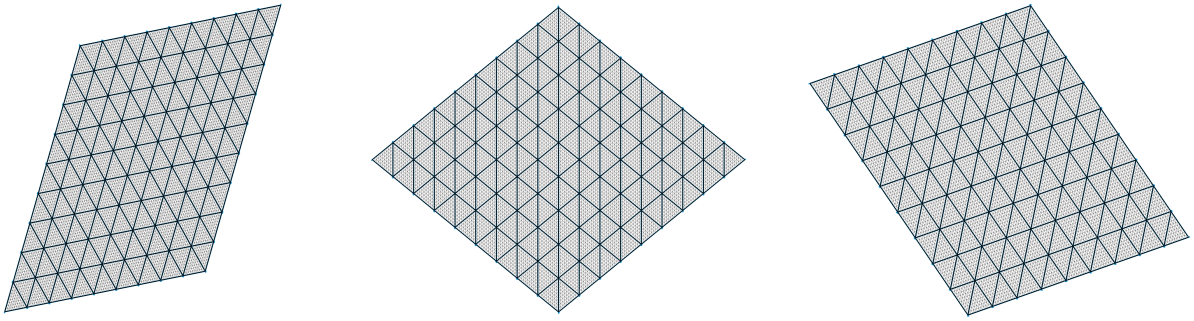
In addition to being isometric, the optimized triangulations are also quadratic if $\mathcal{M}^{-1/2}$ is linear. The metrics

$$\mathcal{M}^{-1/2} = \begin{pmatrix} x & 0 \\ 0 & 1 \end{pmatrix} \quad \to \quad [\mathcal{M}] = \begin{pmatrix} 1+f_x^2 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} x^{-2} & 0 \\ 0 & 1 \end{pmatrix} \tag{4.65}$$

are induced by the surface $S_4 = (x, y, f(x))$, with $f(x)$ defined on $(0, 1)$ by:

$$f(x) = \frac{x\sqrt{x^{-2}-1}\left(\sqrt{x^2-1} - \arctan(\sqrt{x^2-1})\right)}{\sqrt{x^2-1}}, \quad \text{so that} \quad f_x = \sqrt{x^{-2}-1}. \tag{4.66}$$

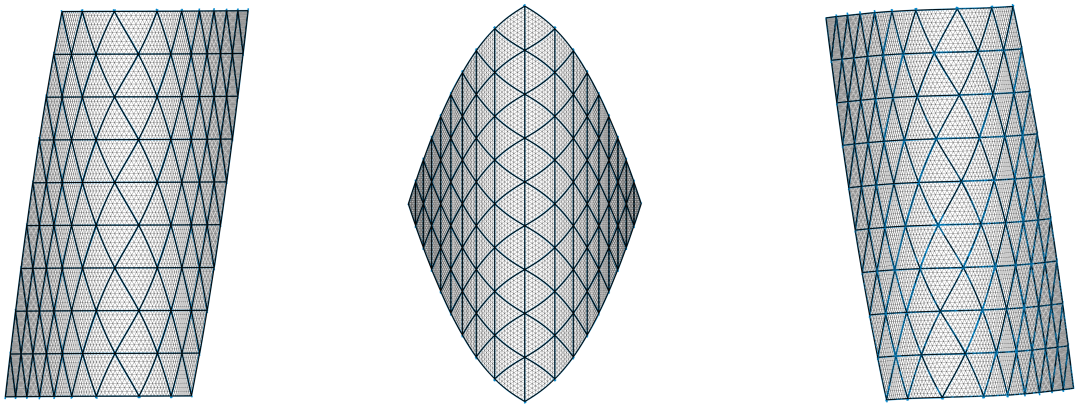
This function is real-valued on $(0, 1)$ despite the roots of x^2-1. To avoid complications outside of $(0, 1)$ where f explodes or is complex-valued, the initial mesh covers $[0.0174, 0.3495]\times[-2, 2]$



x

$E_{\text{initial}} = 3.9862\text{e+}00$ $E_{\text{initial}} = 5.8737\text{e+}00$ $E_{\text{initial}} = 7.8995\text{e+}00$

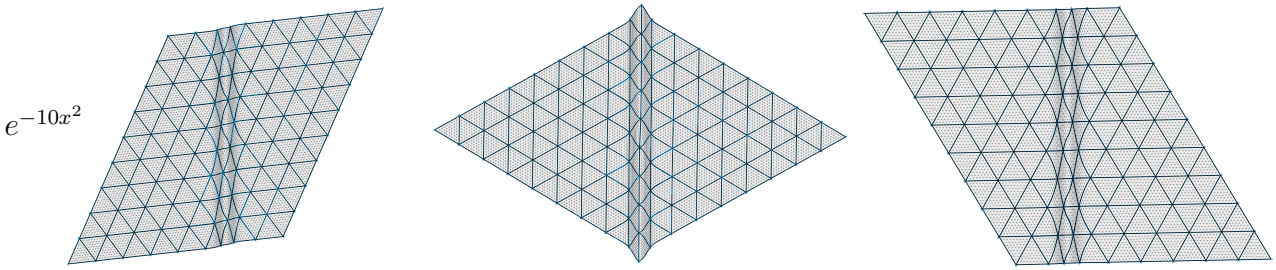
$E_{\text{final}} = 1.3417\text{e-}14$ $E_{\text{final}} = 7.2833\text{e-}14$ $E_{\text{final}} = 1.4437\text{e-}14$



x^2

$E_{\text{initial}} = 3.9044\text{e+}00$ $E_{\text{initial}} = 3.4381\text{e+}03$ $E_{\text{initial}} = 4.6694\text{e+}03$

$E_{\text{final}} = 3.0017\text{e-}08$ $E_{\text{final}} = 1.5992\text{e-}03$ $E_{\text{final}} = 2.1175\text{e-}05$

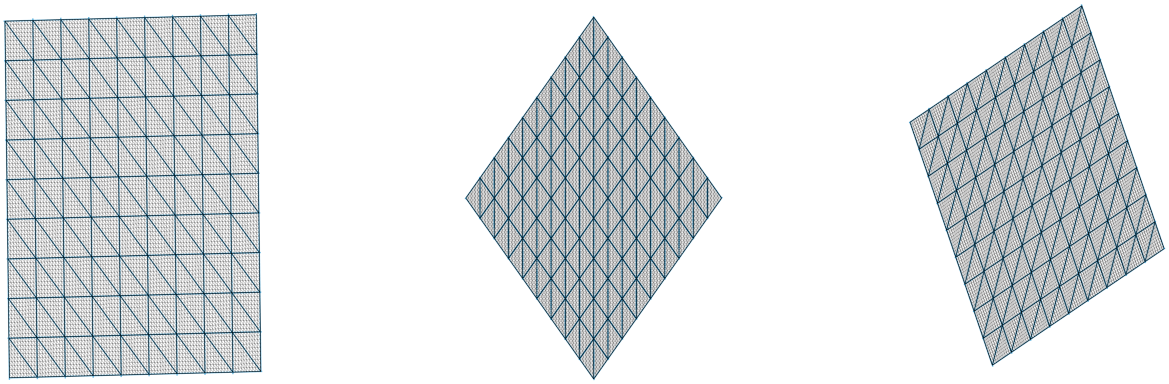


e^{-10x^2}

$E_{\text{initial}} = 3.7692\text{e+}00$ $E_{\text{initial}} = 7.7535\text{e+}00$ $E_{\text{initial}} = 9.0946\text{e+}00$

$E_{\text{final}} = 6.2525\text{e-}05$ $E_{\text{final}} = 2.0653\text{e-}03$ $E_{\text{final}} = 1.2795\text{e-}04$

Figure 4.8 Triangulations with $N = 10$ subdivisions which are isometric to $\mathcal{T}_{\triangle}$.

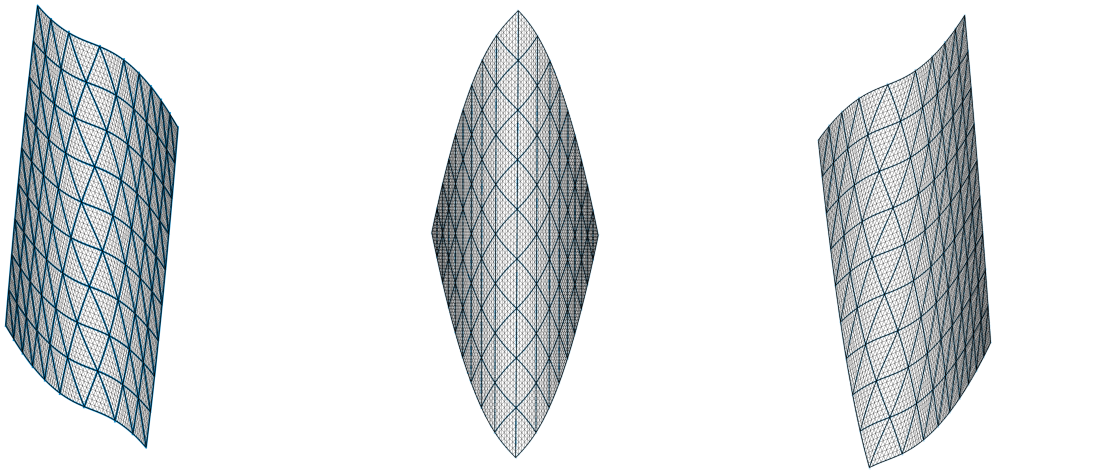


x

$E_{\text{initial}} = 8.3662\text{e+}00$ $E_{\text{final}} = 2.8419\text{e-}12$

$E_{\text{initial}} = 2.7338\text{e+}00$ $E_{\text{final}} = 4.0016\text{e-}12$

$E_{\text{initial}} = 3.8237\text{e+}00$ $E_{\text{final}} = 3.9511\text{e-}12$

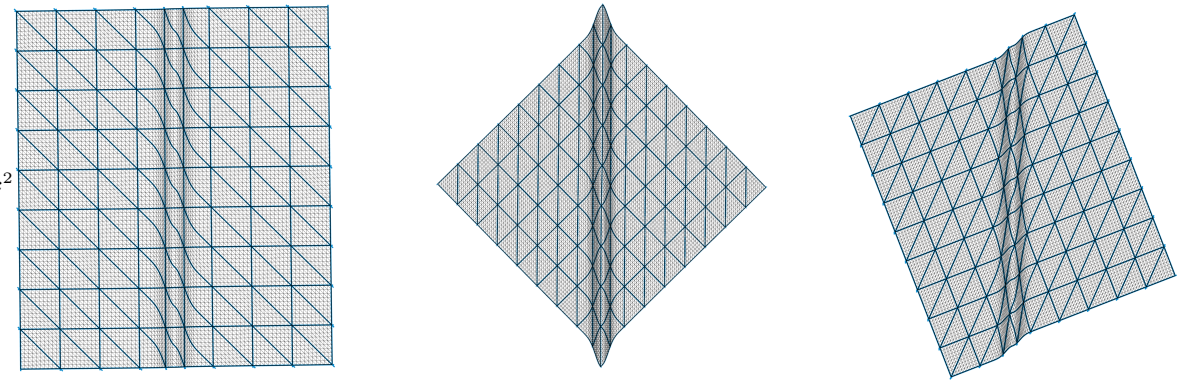


x^2

$E_{\text{initial}} = 1.2873\text{e+}01$ $E_{\text{final}} = 3.7719\text{e-}08$

$E_{\text{initial}} = 5.4106\text{e+}03$ $E_{\text{final}} = 4.2228\text{e-}03$

$E_{\text{initial}} = 7.2785\text{e+}03$ $E_{\text{final}} = 6.2878\text{e-}08$



e^{-10x^2}

$E_{\text{initial}} = 7.9922\text{e+}00$ $E_{\text{final}} = 4.2048\text{e-}05$

$E_{\text{initial}} = 4.6135\text{e+}00$ $E_{\text{final}} = 1.6983\text{e-}03$

$E_{\text{initial}} = 5.5453\text{e+}00$ $E_{\text{final}} = 7.5553\text{e-}05$

Figure 4.9 Triangulations with $N = 8$ subdivisions which are isometric to $\mathcal{T}_{\triangle}$.

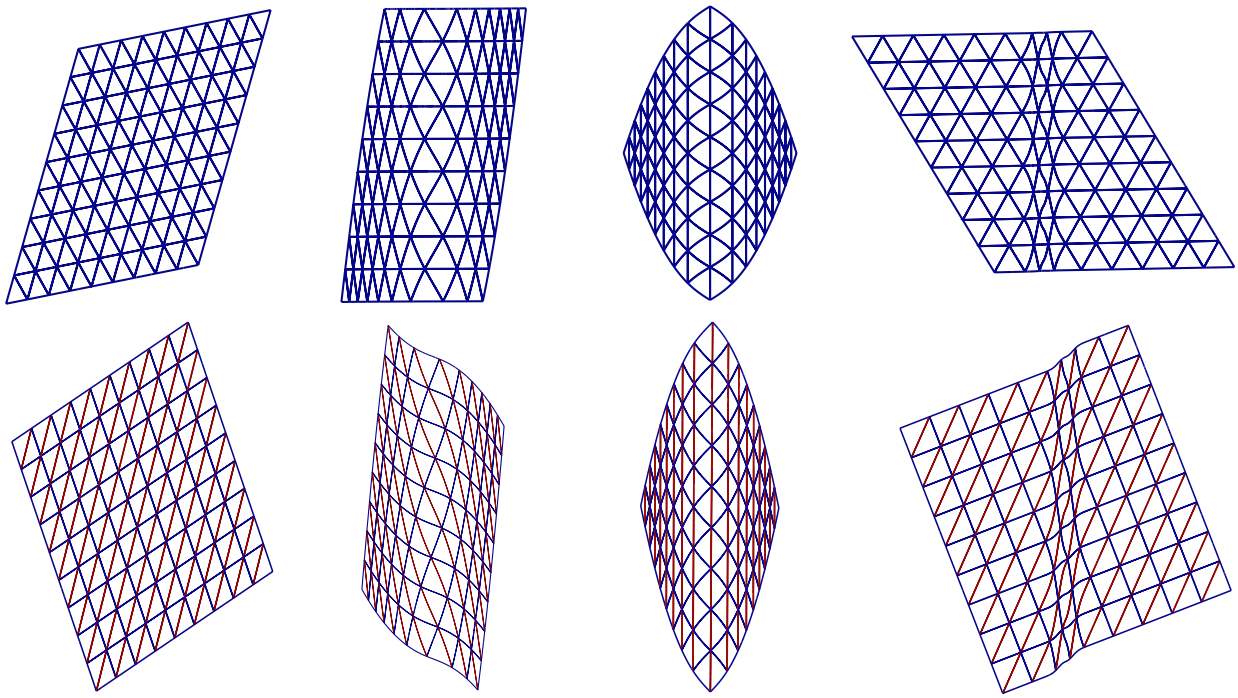


Figure 4.10 Edge length with respect to the metric. On top, all edges are unit. On the bottom, the edges are either unit (blue) or of length $\sqrt{2}$ (red).

when optimizing for $\mathcal{T}_{\triangle}$ and $[0.0174, 0.95]\times[-2, 2]$ for $\mathcal{T}_{\triangleleft}$. This yields respectively 3 and 4 unit edges along x and 4 along y. When optimizing for $\mathcal{T}_{\triangle}$, the vertices can move freely, whereas they slide along the boundary for $\mathcal{T}_{\triangleleft}$, as the initial mesh is close to the optimal triangulation. In that case, the boundary vertices of the initial mesh are obtained by bisection to split the upper and lower boundaries into 4 unit edges. The initial subdivision is uniform and the additional vertices on the boundaries are free to slide to comply with the metric. The optimized triangulations are shown in Fig. 4.11. As a matter of fact, they are the only isometric $\mathcal{P}^2$ triangulations shown in this thesis, i.e., the only perfectly unit $\mathcal{P}^2$ meshes.

4.4.2 Nondevelopable surfaces

Next, we consider isometric triangulations on less trivial surfaces.

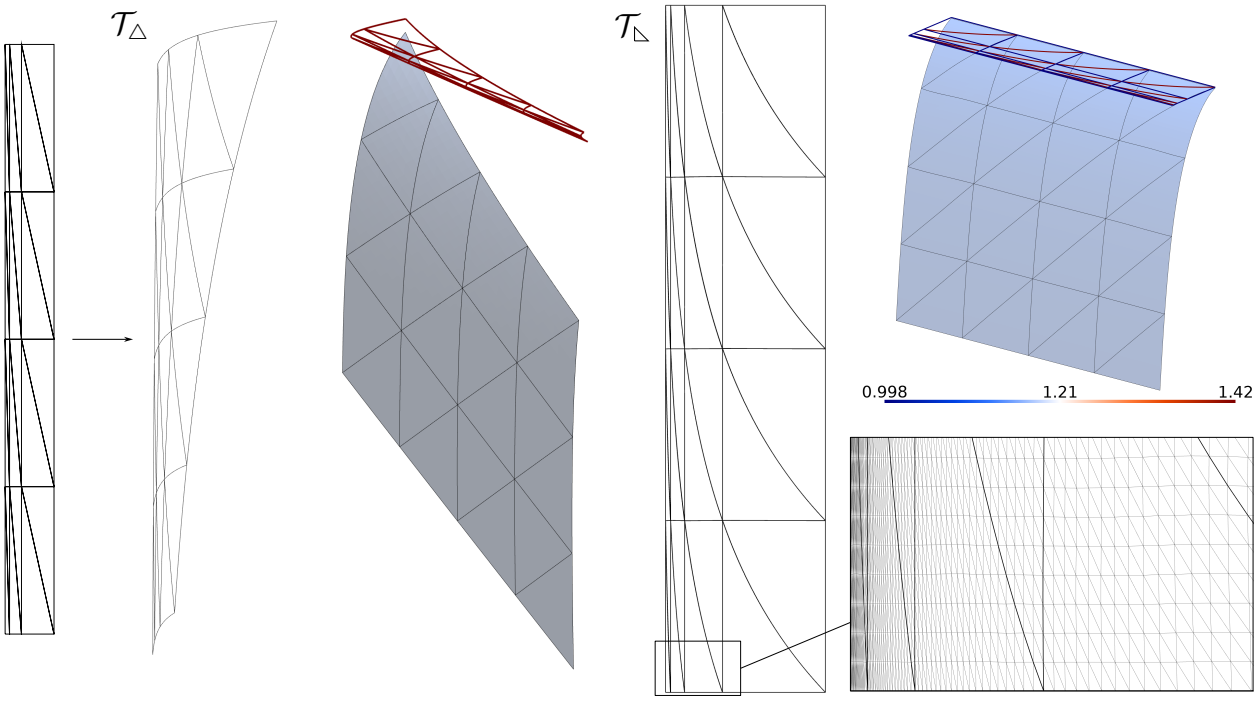


Figure 4.11 Isometric $\mathcal{P}^2$ triangulations with respect to S_4. The equilateral mesh has unit edges everywhere.

Radial function

The surface S_5 is the graph of $f(x, y) = x^2 + y^2 = r^2$. In polar coordinates, the metric writes:

$$[\mathcal{M}] = \begin{pmatrix} 1 + 4r^2 & 0 \\ 0 & 1 \end{pmatrix}, \tag{4.67}$$

hence the length of an arc on the surface with constant radius is equal to its Euclidean length, and the metric only modifies the computation of lengths in the radial direction. We start from a uniform square mesh of $[-4, 4]^2$ and let all vertices move freely during the optimization, yielding the curved meshes on top of Fig. 4.12. The top row is optimized for $\mathcal{T}_\triangle$ and the bottom row for $\mathcal{T}_\triangleleft$. As S_5 is cup-shaped with circular isolines, the boundaries curve to approximate a circle, which may yield an invalid triangulation despite all subtriangles being valid. For this configuration, the mesh for $\mathcal{T}_\triangleleft$ is invalid and the boundary of the mesh for $\mathcal{T}_\triangle$ is also about to become inverted. Since the macrotriangles are free to curve into parameterizations which are not necessarily polynomial, their validity cannot be easily enforced by an additional term in the cost function, as is done in Toulorge et al. [16] for $\mathcal{P}^2$ triangles with a log-barrier involving the Bézier control coefficients of the Jacobian determinant. To obtain a valid mesh, the boundary is fixed instead, as the interior elements cannot degenerate without

inverting any subtriangles. The distortion-based quality and the length of the edges with respect to the metric are also shown in Fig. 4.12. Away from the singularities, the quality of the subtriangles is close to 1 and the edges are close to unit.

Away from the origin, we can expect the optimized mesh to be very close to isometric. In Fig. 4.13, we start from a uniform mesh of $[-1, 5] \times [-3, 3]$, not centered at the origin. The optimized meshes are essentially isometric to $\mathcal{T}_\triangle$ or $\mathcal{T}_\triangleleft$: the curved edges match the geodesics, the distortion-based quality lies in $[0.992, 1]$ for all subtriangles and the edges preserve the length of the reference simplex with respect to the metric. The edges length lie within $[0.94, 1.02]$ for triangles isometric to the unit equilateral triangle and within $[0.96, 1.42]$ for triangles isometric to the right triangle of sides 1 and $\sqrt{2}$. The honeycomb pattern of $\mathcal{T}_\triangle$ and the split squares pattern of $\mathcal{T}_\triangleleft$ only differ by their inner angles with respect to the metric, which are $\pi/3$ for the former and $\pi/2$ and $\pi/4$ for the split squares (the exact angles of the optimized meshes with respect to the metric are not shown). In the Euclidean space, it is thus subtle to distinguish between both patterns, as their inner angles depends on the position, however the edges length plot shows which edges are longer than the others and gives a better idea of the inner angles. The optimized triangulations are folded back onto S_5 in Fig 4.14. This allows to quickly assess the quality of the optimization: geodesic meshes should be straight on the surface despite being curved in the parameter space. This is clearly not the case for the first two meshes. The last one away from zero, on the other hand, is very close to being isometric: its projection is a straight uniform tiling of equilateral triangles on the surface.

To fix the boundary at a better position, we consider a circular domain and a ring domain matching the isolines of f, Fig. 4.15. For both geometries, only the vertices on the outer boundaries of the starting mesh are fixed, so that the added subvertices initially on straight edges can move towards the circular geodesics. For the circle, the radius $r_0 = N_0/2\pi$ is chosen so that the perimeter can be divided into an integer number $N_0 = 30$ of edges. For the ring, the inner radius is set to fit $N_0 = 10$ edges. There is an additional constraint on the outer radius r_1, as it should be chosen so that an integer number of edges fits on the perimeter, as well as in the radial direction of length $r_1 - r_0$. In the metric induced by S_5, the length of a radial geodesic $\gamma(t) = (r_0 + t(r_1 - r_0), 0)$ with $r_1 > r_0$ has a closed-form expression given by:

$$\begin{aligned}\ell_{\mathcal{M}}(\gamma) &= \int_0^1 \sqrt{\gamma'^T[\mathcal{M}_{\gamma(t)}]\gamma'}\, dt = \int_0^1 \sqrt{(r_1 - r_0)^2\left[1 + 4(r_0 + t(r_1 - r_0))^2\right]}\, dt \\ &= -\frac{1}{4}\left(\operatorname{arcsinh}\left(2r_0\sqrt{1 + 4r_1^2} - 2r_1\sqrt{1 + 4r_0^2}\right)\right) + 2r_0\sqrt{1 + 4r_0^2} - 2r_1\sqrt{1 + 4r_1^2}.\end{aligned} \tag{4.68}$$

To get an outer radius fitting an integer number of edges along both directions, we compute

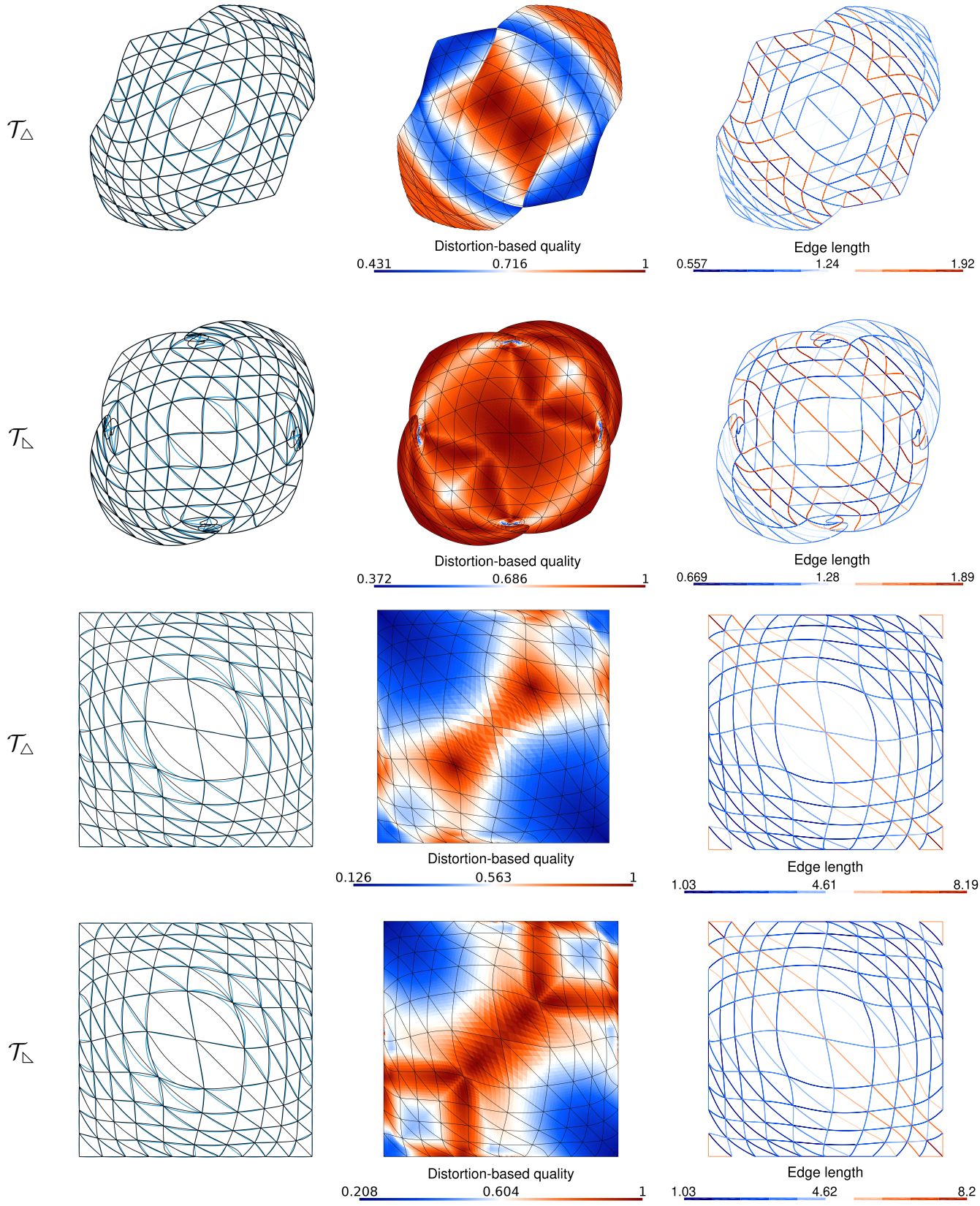


Figure 4.12 Optimized triangulations for S_5 around the origin with free (top) and fixed (bottom) boundary.

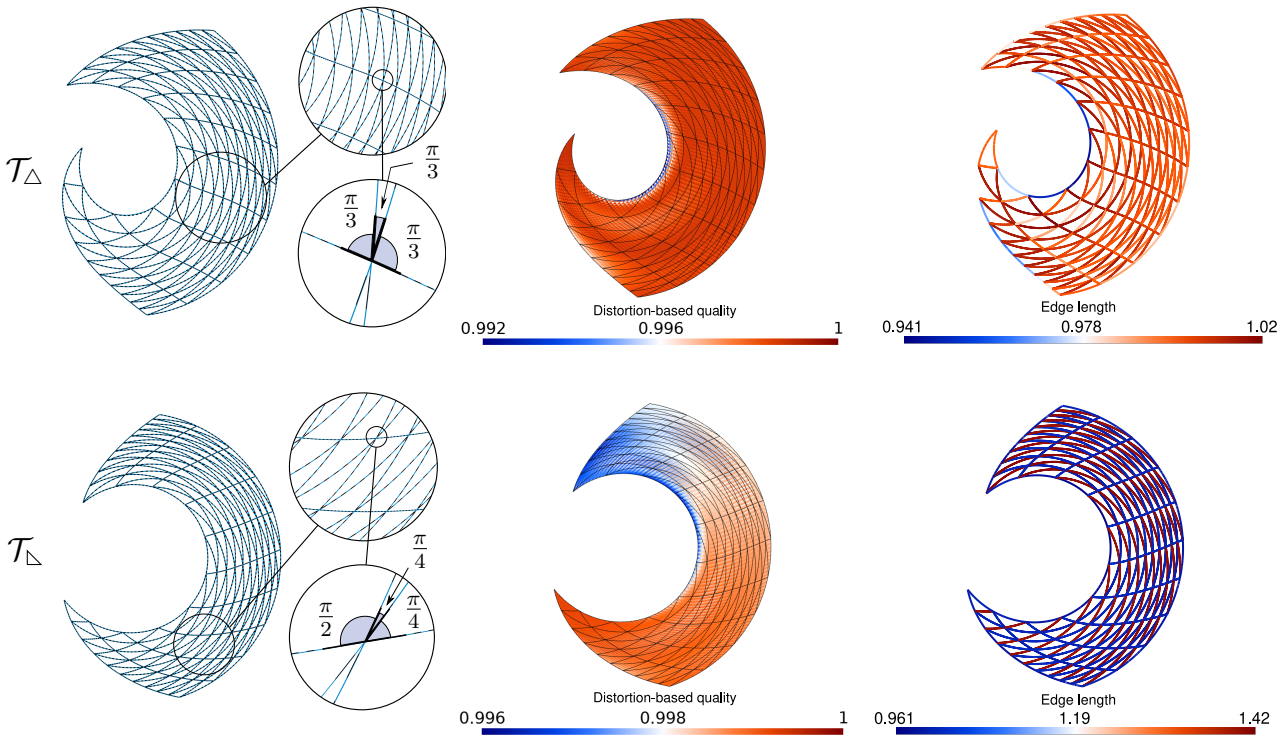


Figure 4.13 Quasi-isometric triangulations for S_5 away from the origin.

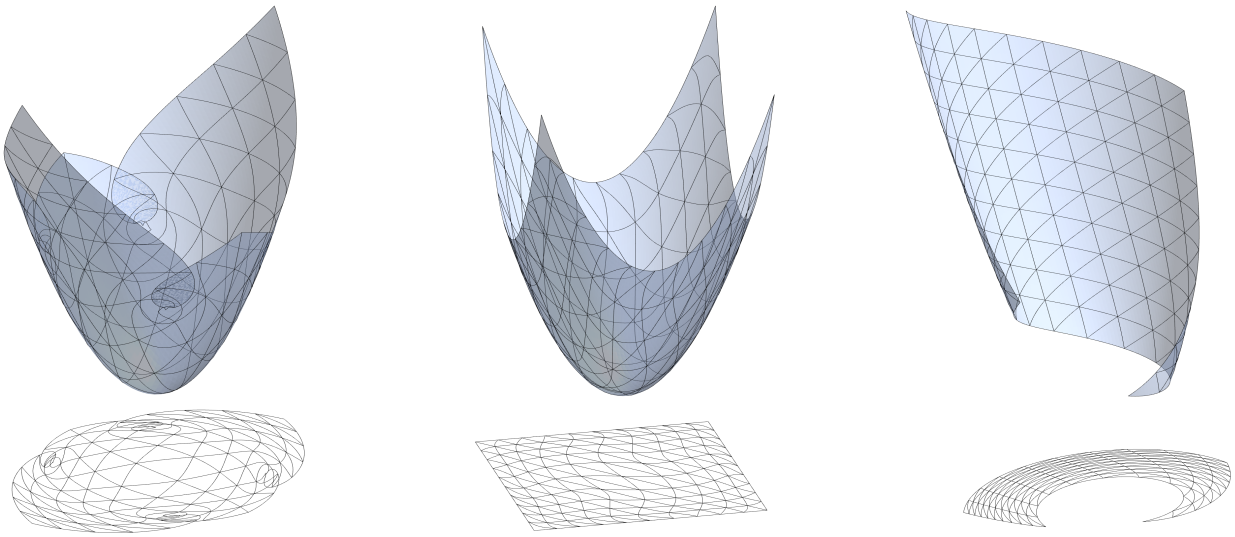


Figure 4.14 Optimized triangulations for $S_5 = (x, y, x^2 + y^2)$ and their projection on S_5.

r_1 for various integer lengths $\ell_{\mathcal{M}}(\gamma)$ using a bisection, then select the r_1 yielding a perimeter the closest from an integer number, while keeping the ring relatively small. For $r_0 = 10/2\pi = 1.592$, we obtain $r_1 = 4.612$, yielding exactly $\ell_{\mathcal{M}}(\gamma) = 19$ unit edges in the radial direction and

$28.98 \simeq 29$ unit edges on the perimeter. Contrary to the other examples, the initial mesh is anisotropic and complies with the metric, Fig. 4.15. It is created with the anisotropic mesh generator BAMG [66], interfaced by Gmsh. As a standard metric-based mesh generator, BAMG aims at generating edges of unit length with respect to $\mathcal{M}$, thus linear unit elements with respect to $K_\triangle$, in the usual sense of Definition 4.1. To keep the optimization consistent with this initial adaptation, the meshes are optimized for $\mathcal{T}_\triangle$ only. Moreover, the mesh is obtained from standard insert and collapse operations, hence the topology is arbitrary and not the optimal one of $\mathcal{T}_\triangle$, that is, inner vertices do not always have 6 neighbouring triangles.

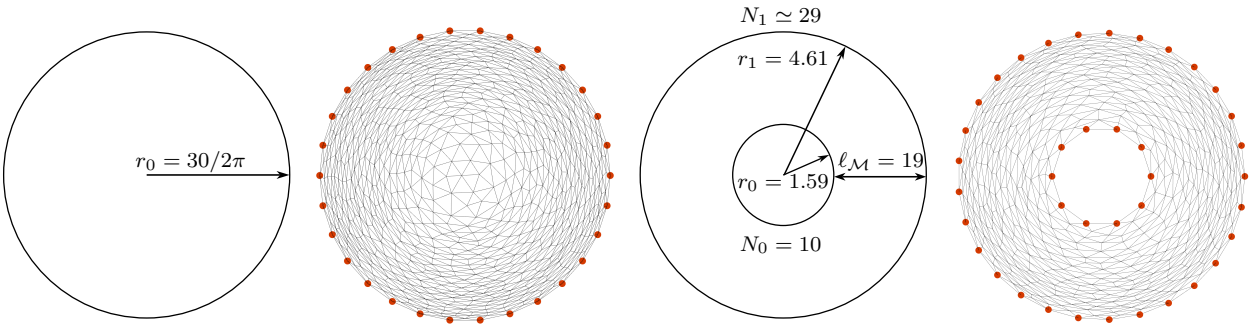


Figure 4.15 Circle and ring geometry and initial anisotropic meshes. The red dots are the fixed vertices.

The optimized triangulations for the circle and ring domains are shown in Fig. 4.16: the edges follow the geodesics almost everywhere, contrary to the examples starting from a square mesh of Fig. 4.12. The quality is high and the edges are close to unit. In particular, except for a few outliers, the edges are quasi-unit in the usual sense of $\ell_{\mathcal{M}} \in [1/\sqrt{2}, \sqrt{2}]$, as shown by the histograms. The elevations on S_5 are shown in Fig. 4.17. They are mostly straight and structured, despite their projections being curved. This confirms that the edges are close to geodesics on S_5 for the Euclidean metric of $\mathbb{R}^3$.

Ondulating boundary layer

The surface S_6 is adapted from the one proposed by Aparicio-Estrems et al. in [26]. It mimics an ondulating boundary layer and its metric is defined by:

$$[\mathcal{M}] = J_\varphi^T \begin{pmatrix} 1 & 0 \\ 0 & h^{-2}(x,y) \end{pmatrix} J_\varphi, \quad \text{with} \quad \varphi(x,y) = \left(x, \frac{10y - \cos(2\pi x)}{\sqrt{100 + 4\pi^2}} \right) \tag{4.69}$$

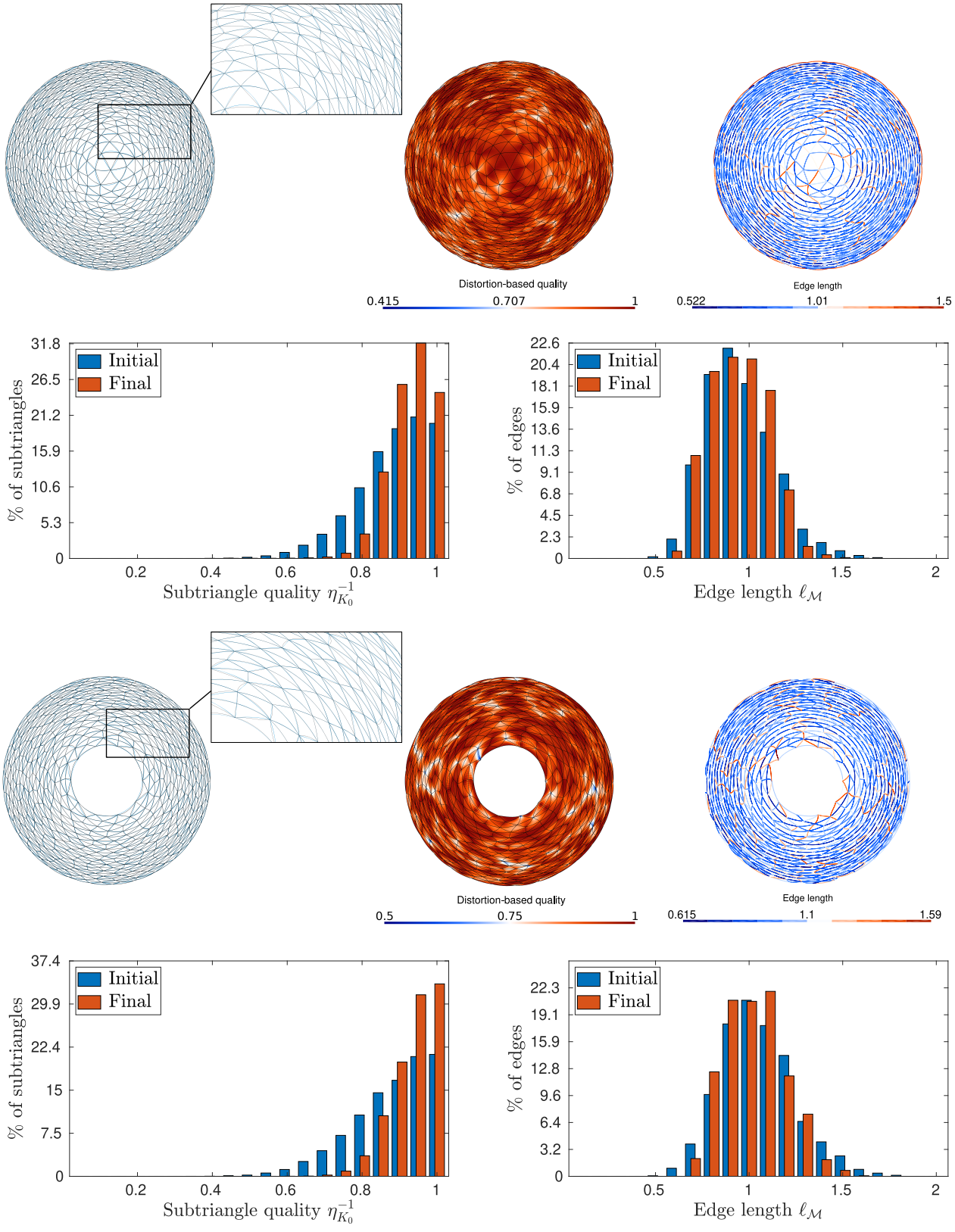


Figure 4.16 Quasi-isometric triangulations for S_5 from an initial anisotropic mesh.

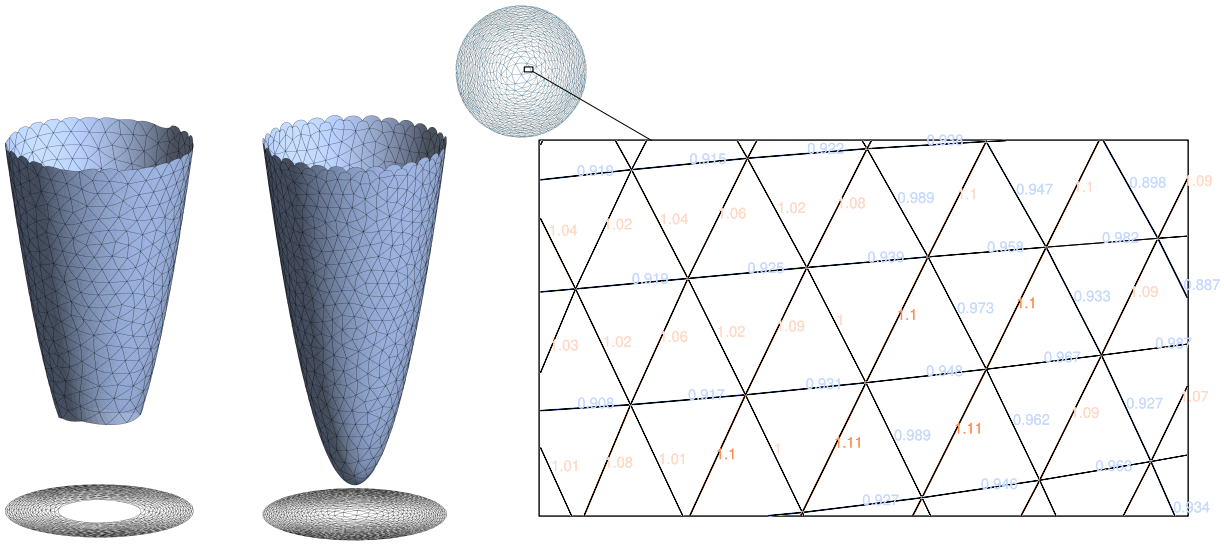


Figure 4.17 Left: Optimized triangulations for S_5 and their projection on the surface. Right: Quasi-unit edge length of subtriangles.

and where the sizing function given by:

$$h(x,y) = 0.1 + b\left|\frac{10y - \cos(2\pi x)}{\sqrt{100 + 4\pi^2}}\right|. \tag{4.70}$$

The mesh size is constant in the x-direction (before applying the map φ) and grows from 0.1 in the y-direction according to $h(x,y)$. The anisotropic quotient of $\mathcal{M}$, defined in two dimensions by [1]:

$$\text{quo} = \max_{i=1,2}\sqrt{\frac{h_i^2}{h_1 h_2}} = \max_{i=1,2}\frac{(\det \mathcal{M})^{1/4}}{\sqrt{\lambda_i}} \tag{4.71}$$

is shown in Fig. 4.18. It is a ratio of principal lengths obtained from the ratio of areas associated to the principal sizes h_1, h_2. The anisotropy is the highest along the curve $10y - \cos(2\pi x) = 0$. The initial triangulation is a uniform mesh of $[-1,1]^2$, Fig. 4.18 and all

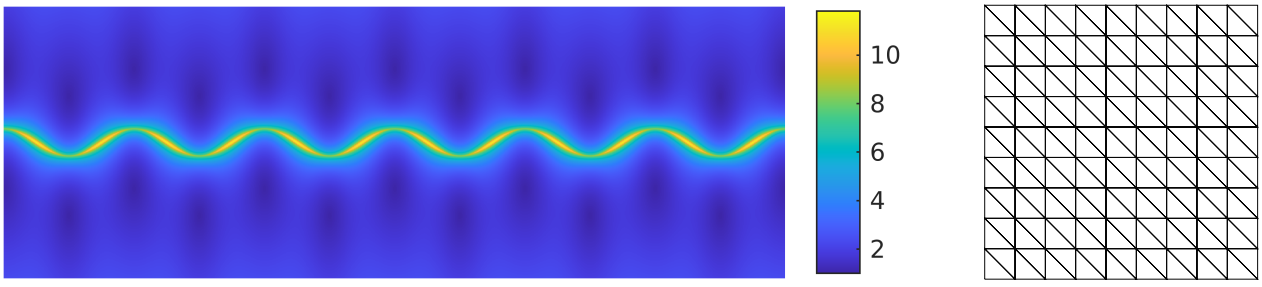


Figure 4.18 Anisotropic quotient of $\mathcal{M}$ in log scale for $[-3,3]\times[-1,1]$ and initial mesh.

vertices are free to move. The optimized meshes with respect to $\mathcal{T}_\triangle$ and $\mathcal{T}_\triangleleft$ are shown in Fig. 4.19. The curved elements follow the waves of the boundary layer as expected. Because the characteristic mesh size is set to 1 along the x-direction and the argument of the cosine is $2\pi x$, a unit edge covers exactly one wavelength. Thus, to accurately represent the curved edges with high-order triangles, polynomials of degree at least 3 would be required. For both optimized meshes, the quality of the subtriangles lies within $[0.9, 1]$ almost everywhere, but drops as low as 0.72 in the boundary layer where the curvature is the highest. This drop depends directly on the number of subtriangles used to represent the region of high curvature and is reduced by further refining the macrotriangles. The edge lengths with respect to the metric agree with the chosen reference triangle, and homologous edges are quasi-isometric. In [26], the meshes are optimized with respect to the distortion $\eta_{K_\triangle}$. As the distortion is length-agnostic and yields conformally flat triangulations, the elements size is thus determined by the initial mesh. In other words, the meshes obtained in [26] after distortion minimization are (quasi-)conformally equivalent to $\mathcal{T}_\triangle$, that is, they preserve the angle of the tiling. They are not (quasi-)isometric to $\mathcal{T}_\triangle$ in general, unless the optimized mesh also happens to preserve the lengths. Here, the length is enforced locally instead and the minimization of the distortion is achieved as a consequence of the macrotriangles being quasi-isometric to either $K_\triangle$ or $K_\triangleleft$, in agreement with Proposition 4.5.

Smooth compactly supported bump

Finally, the surface S_7 is the bump defined for $r_0 = 2$ by:

$$S_7(r) = \begin{cases} \exp\left(1/(r^2 - r_0^2)\right), & \text{if} \quad r < r_0, \\ 0, & \text{if} \quad r \geq r_0. \end{cases} \tag{4.72}$$

This is a typical and simple example of a smooth function with compact support, as it has continuous derivatives of arbitrary order at $r = r_0$ and it vanishes outside of r_0. As a result, the induced metric for $r > r_0$ is exactly the Euclidean metric, and we can set the initial meshes to be the reference tilings $\mathcal{T}_\triangle$ or $\mathcal{T}_\triangleleft$ and fix their boundary, since boundary elements are already of the right length. The optimized meshes are shown in Fig. 4.21 for $\mathcal{T}_\triangle$ and in Fig. 4.22 for $\mathcal{T}_\triangleleft$. The effect of the moving log-barriers on the optimization is also illustrated. In both cases, the initial mesh is unit for the Euclidean metric: because the metric in the bump is the Euclidean metric supplemented by strictly positive terms, the area of the initial subtriangles in the bump is greater than 1, so that the minimum of $|k|_{\mathcal{M}}/|k|_{\text{target}}$ is 1 when starting the optimization, but quickly decreases during the process. Each new barrier pushes the minimum area towards 1 and accelerates the optimization.

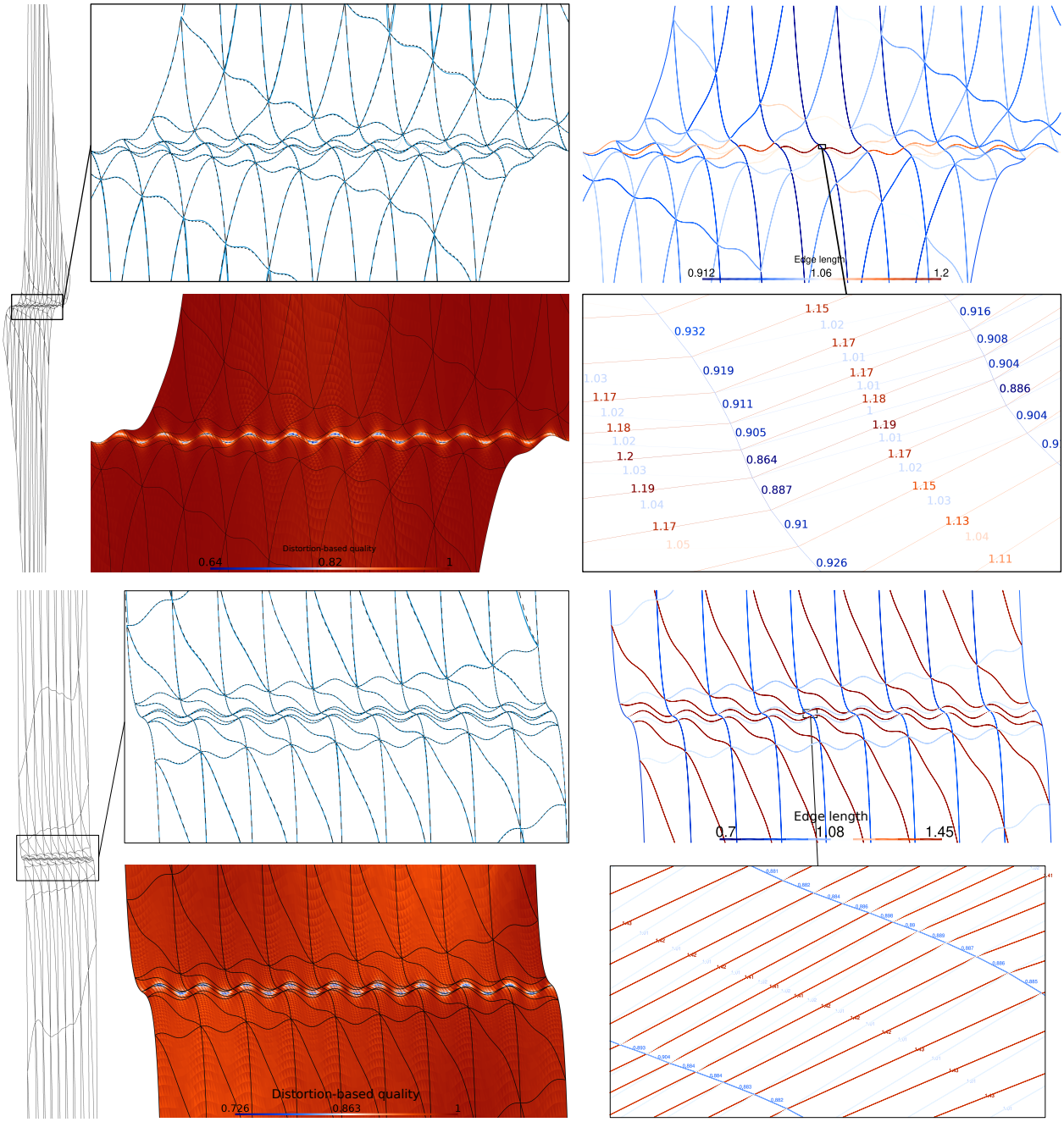


Figure 4.19 Triangulation for S_6 with $N = 12$ (top) and $N = 16$ (bottom) subdivisions and optimized with respect to $\mathcal{T}_{\triangle}$. First and third rows: curvilinear mesh with geodesics and edges length with respect to the metric. Second and fourth rows: distortion-based quality and edge length of some subtriangles.

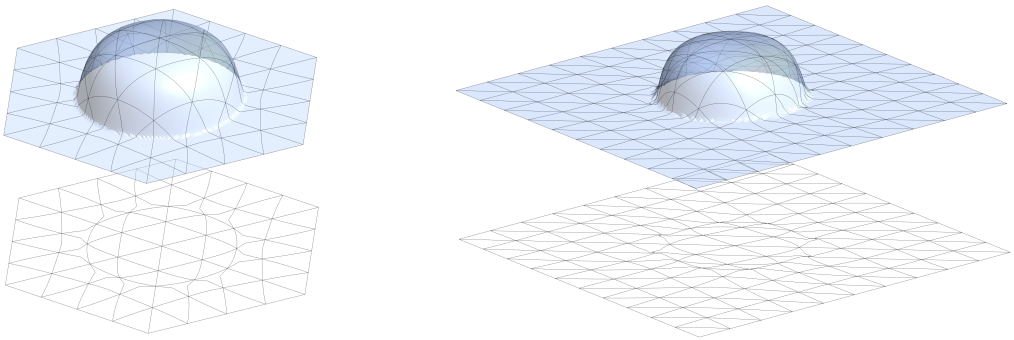


Figure 4.20 Optimized triangulations for S_7 and their projection on the surface.

4.5 Continuous interpolation error estimates on unit elements

We now turn to the estimation of interpolation error on linear and curvilinear unit triangles, with the end goal of writing local upper bounds for the error involving only the metric, called **continuous error estimates**. Following Alauzet and Loseille [55] and Loseille [1], these continuous estimates are integrated over the domain to obtain metric-based error bounds in L^p norm, which are then minimized to yield an optimal Riemannian metric for the adaptation. In the literature, error estimates are available in a discrete setting on arbitrary linear triangles, with bounds involving the vertices or edges of the triangle. These discrete estimates are translated to continuous estimates using the definition of unitness and the associated invariants.

Discrete anisotropic estimates for linear interpolation ($k = 1$) on linear simplices ($q = 1$) are available in the literature for several target integral or pointwise norms, see for instance the work of D'Azevedo and Simpson [71], Berzins [72], Apel [73], Formaggia and Perotto [74], Chen et al. [75, 76] and the references therein. These classical estimates relate the linear interpolation error to the Hessian matrix of the function f of interest, which, in turn, is used to derive a Riemannian metric. An estimate is typically first obtained on a reference simplex, then written for an arbitrary linear simplex by applying the reference-to-physical transformation.

Estimates for quadratic and high-order interpolation ($k > 1, q = 1$) have been investigated by e.g. Cao [77–79], Mirebeau [38, 80], Mbinky [10], Kuate and Hecht [81], Dolejší [82] and Coulaud and Loseille [11,12]. General-purpose estimates extend the linear case by relating the interpolation error to higher-order derivatives. The main obstacle for anisotropic adaptation is that higher-order derivatives do not immediately relate to a Riemannian metric. To mimic the linear case, the high-order derivatives are bounded by a positive-definite quadratic form

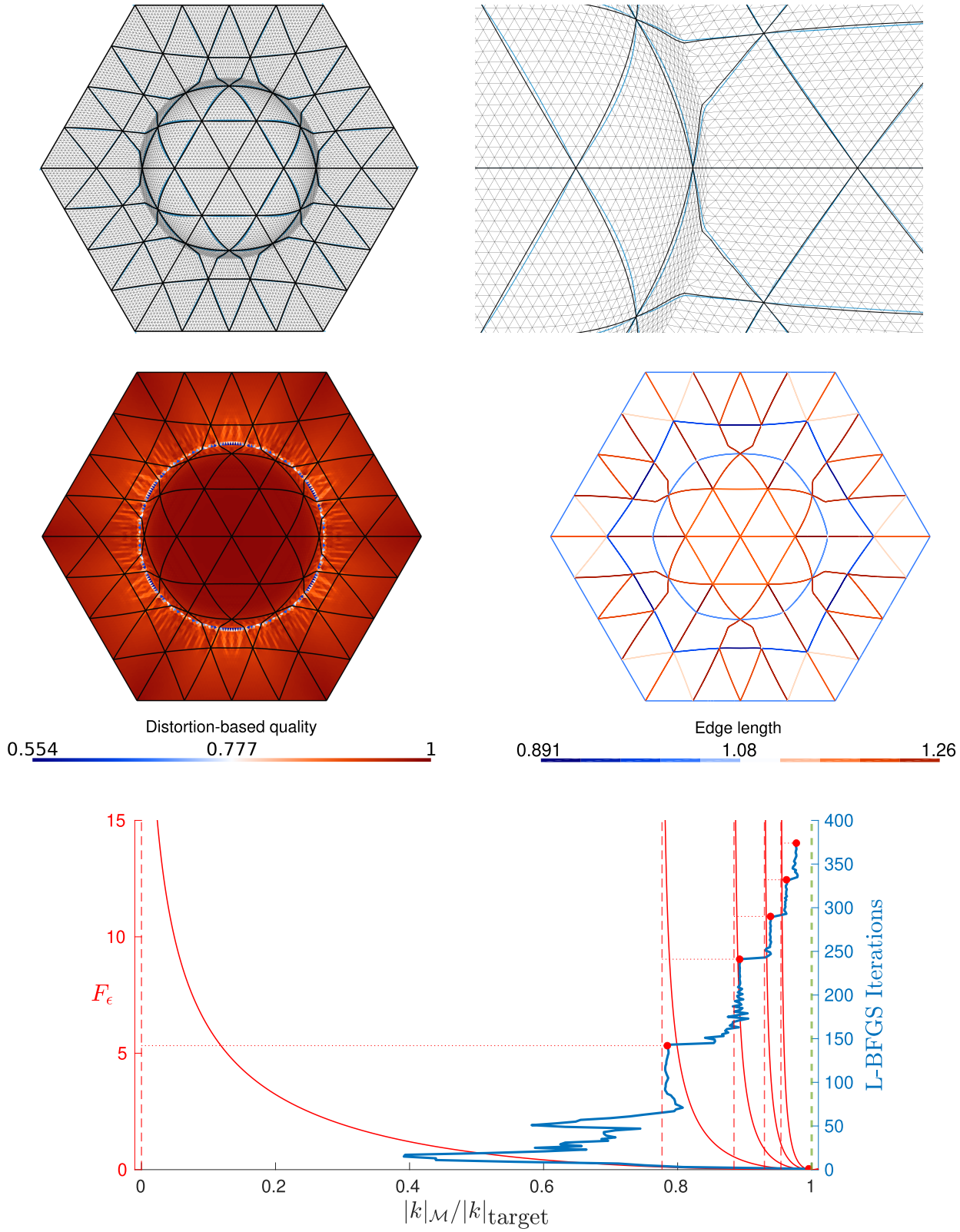


Figure 4.21 Optimization of $\mathcal{T}_\triangle$ for S_7 with $N = 16$. Curved quasi-isometric mesh (top), quality of the subtriangles and edge length (middle) and influence of the log-barriers on the evolution of the minimum metric-weighted area during the optimization (bottom).

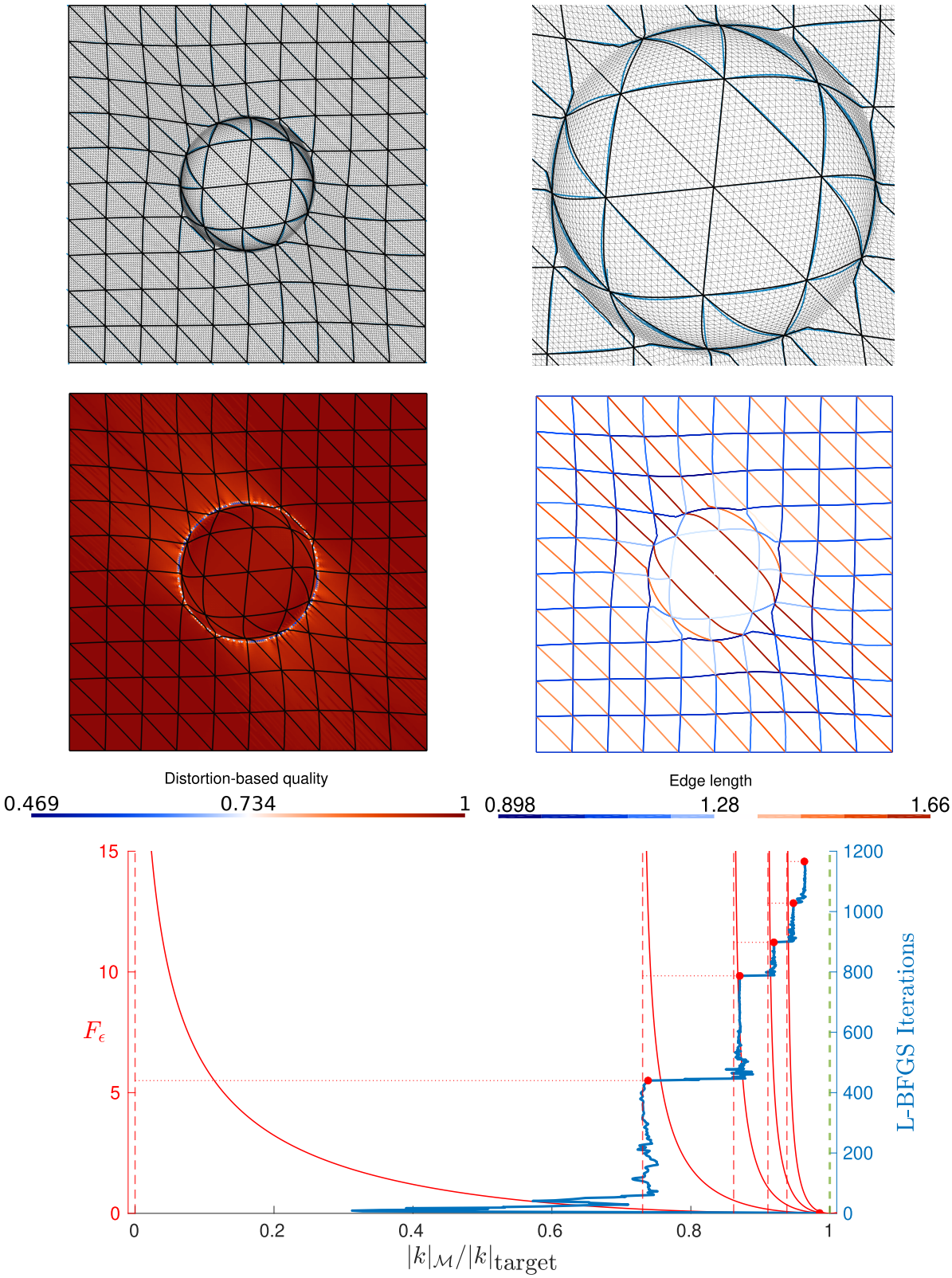


Figure 4.22 Optimization of $\mathcal{T}_{\triangleleft}$ for S_7 with $N = 16$. Curved quasi-isometric mesh (top), quality of the subtriangles and edge length (middle) and influence of the log-barriers on the evolution of the minimum metric-weighted area during the optimization (bottom).

$\mathcal{Q}$, defining a metric as a tight upper bound on the high-order behaviour of f. In Cao, the quadratic form $\mathcal{Q}$ is called the *anisotropic measure* of the high-order derivatives of f.

Fewer results are available for interpolation error estimates on curved elements ($q > 1$). Aside from the early work of Ciarlet and Raviart [52], an isotropic and conservative estimate is given in Moxey et al. [48], involving an upper bound on the derivatives of f. To the best of our knowledge, the only anisotropic estimate is given by Rochery [17, 33] for linear interpolation on $\mathcal{P}^2$ meshes, extending the estimate of Chen [75] to curved triangles.

We briefly present existing discrete interpolation error estimates for both linear and curved triangles, as well as their continuous counterpart on unit triangles, involving only the unknown Riemannian metric and the derivatives of f. Then, we propose a simple extension of high-order estimates to curved unit triangles. This estimate can be used to derive optimal metrics for high-order mesh adaptation. The presented estimates are based on Taylor expansions, and thus assume sufficiently smooth functions, that is, at least C^{k+1} for interpolation of degree k.

4.5.1 Linear elements

In this section, K is a unit linear triangle with respect to the Riemannian metric $\mathcal{M}$, with vertices $(\boldsymbol{x}_i)_{i=1}^3$ and edges $\boldsymbol{e}_i = \boldsymbol{e}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i$. Since K is unit for $\mathcal{M}$, there is an isometry $F \triangleq F_{\triangle \to K}$ between $K_\triangle$ and K with constant Jacobian matrix $J_{\triangle \to K} = \mathcal{M}^{-1/2} R$. We denote by $\Pi_K^k f$ the bivariate Lagrange interpolation polynomial of degree k of f on K, and by $e_K^k f = f - \Pi_K^k f$ the associated interpolation error.

Linear interpolation

A continuous estimate for linear interpolation was originally derived by Alauzet and Loseille [55], by writing a Taylor expansion for the error function then manipulating the diagonalized form of the Hessian matrix of f. Here, we derive the continuous estimate following Rochery [17], where it is obtained directly from the edge-based estimate of Chen [75]. We denote by $H_f \triangleq (\nabla^2 f)_{ij}$ the Hessian matrix of f, and $|H_f|$ its absolute value defined by (3.72). For any $\boldsymbol{p} \in K$ and for some positive constant C, the following interpolation error bound holds [75, 76]:

$$|e_K^1 f(\boldsymbol{p})| = |(f - \Pi_K^1 f)(\boldsymbol{p})| \leq C \sum_{i,j=1}^{3} \boldsymbol{e}_{ij}^T |H_{f,i}(\boldsymbol{p})| \boldsymbol{e}_{ij}, \tag{4.73}$$

where, from Taylor's integral remainder at vertex $\boldsymbol{x}_i$:

$$|H_{f,i}(\boldsymbol{p})| = 2\int_0^1 t\,|H_f(\boldsymbol{x}_i + t\boldsymbol{x}_i\boldsymbol{p})|\,dt = 2\int_0^1 (1-s)\,|H_f(\boldsymbol{p} + s\boldsymbol{p}\boldsymbol{x}_i)|\,ds. \tag{4.74}$$

Taylor's expansion for $H_f(\boldsymbol{p} + s\boldsymbol{p}\boldsymbol{x}_i)$ writes:

$$H_f(\boldsymbol{p} + s\boldsymbol{p}\boldsymbol{x}_i) = H_f(\boldsymbol{p}) + \nabla^3 f(\boldsymbol{p}) \cdot s\boldsymbol{p}\boldsymbol{x}_i + o(\|s\boldsymbol{p}\boldsymbol{x}_i\|^2), \tag{4.75}$$

thus assuming $\boldsymbol{p}\boldsymbol{x}_i$ and the edges $\boldsymbol{e}_{ij}$ are both $\mathcal{O}(h)$, we write:

$$|e_K^1 f(\boldsymbol{p})| \leq C \sum_{i,j=1}^{3} \boldsymbol{e}_{ij}^T |H_f(\boldsymbol{p})| \boldsymbol{e}_{ij} + o(h^3). \tag{4.76}$$

Using the third invariant of Proposition 4.2 and dropping the asymptotic error term, the upper bound evaluates on $K_\triangle$ as:

$$|e_K^1 f(\boldsymbol{p})| \leq C\ \mathrm{tr}\left(\mathcal{M}_{\boldsymbol{p}}^{-1/2} |H_f(\boldsymbol{p})| \mathcal{M}_{\boldsymbol{p}}^{-1/2}\right). \tag{4.77}$$

The explicit dependence in the geometry of K was replaced by the metric tensor, which contains all the information relative to its unit elements. In particular, for $K = K_\triangle$ unit for the Euclidean metric, the continuous error estimate at $\boldsymbol{y} \in K_\triangle$ becomes:

$$|e_{K_\triangle}^1 f(\boldsymbol{y})| \leq C\ \mathrm{tr}\left(|H_f(\boldsymbol{y})|\right). \tag{4.78}$$

With a pointwise error indicator available, the L^p norm of the error on a domain Ω is:

$$\mathcal{E}^p(\mathcal{M}) = \int_\Omega |e_K^1 f(\boldsymbol{x})|^p\,d\boldsymbol{x} \leq C \int_\Omega \mathrm{tr}\left(\mathcal{M}_{\boldsymbol{x}}^{-1/2} |H_f(\boldsymbol{x})| \mathcal{M}_{\boldsymbol{x}}^{-1/2}\right)^p\,d\boldsymbol{x}. \tag{4.79}$$

Quadratic and higher interpolation

For interpolation of order $k \geq 2$ on triangles, there is no straightforward generalization of (4.76) involving only a sum of high-order derivatives evaluations along the edges. These are required to substitute the metric into the error estimate and translate it into a continuous estimate. We start by giving a discrete estimate for interpolation of arbitrary order in L^∞ norm, using Taylor's polynomial. Then, for practical applications, we introduce a majoration of the higher-order derivatives by the anisotropic measure $\mathcal{Q}$, as proposed by e.g. Cao [78]. This majoration provides a closed-form solution for the optimal Riemannian metric, as for the linear case.

Majoration of the higher-order derivatives

Our starting point is the error term for Taylor's polynomial for f around a point $\boldsymbol{p} \in K$, noted $T^k_{\boldsymbol{p}} f$. Using a multi-index α, the error term for $T^k_{\boldsymbol{p}} f$ writes:

$$(f - T^k_{\boldsymbol{p}} f)(\boldsymbol{x}) = \frac{1}{(k+1)!} \sum_{|\alpha|=k+1} D^\alpha f(\boldsymbol{p})(\boldsymbol{x} - \boldsymbol{p})^\alpha + o(\|\boldsymbol{x} - \boldsymbol{p}\|^{k+1}) \tag{4.80}$$

We consider the L^∞ norm of the interpolation error on K, and we derive an upper bound using only Taylor's remainder. Adding and subtracting Taylor's polynomial:

$$\begin{aligned}
\|f - \Pi^k_K f\|_{L^\infty(K)} &= \sup_{\boldsymbol{x} \in K} \left| f(\boldsymbol{x}) - \Pi^k_K f(\boldsymbol{x}) \right| \\
&\leq \sup_{\boldsymbol{p} \in K} \left(\sup_{\boldsymbol{x} \in K} \left| f(\boldsymbol{x}) - T^k_{\boldsymbol{p}} f(\boldsymbol{x}) \right| + \sup_{\boldsymbol{x} \in K} \left| T^k_{\boldsymbol{p}} f(\boldsymbol{x}) - \Pi^k_K f(\boldsymbol{x}) \right| \right) \\
&= \sup_{\boldsymbol{p} \in K} \left(\|f - T^k_{\boldsymbol{p}} f\|_{L^\infty(K)} + \|T^k_{\boldsymbol{p}} f - \Pi^k_K f\|_{L^\infty(K)} \right).
\end{aligned} \tag{4.81}$$

Taylor's polynomial is a degree k polynomial and is its own interpolant, thus $T^k_{\boldsymbol{p}} f = \Pi^k_K T^k_{\boldsymbol{p}} f$ and we write for the second term:

$$\begin{aligned}
\|T^k_{\boldsymbol{p}} f - \Pi^k_K f\|_{L^\infty(K)} &= \left\| \Pi^k_K \left(f - T^k_{\boldsymbol{p}} f \right) \right\|_{L^\infty(K)} \\
&= \left\| \sum_{i=1}^{N(k)} \phi_i(\boldsymbol{x}) \left(f(\boldsymbol{x}_i) - T^k_{\boldsymbol{p}} f(\boldsymbol{x}_i) \right) \right\|_{L^\infty(K)} \\
&\leq \sum_{i=1}^{N(k)} \|\phi_i(\boldsymbol{x})\|_{L^\infty(K)} \times \max_{\boldsymbol{x}_i} \left| f(\boldsymbol{x}_i) - T^k_{\boldsymbol{p}} f(\boldsymbol{x}_i) \right| \\
&\leq \underbrace{\sum_{i=1}^{N(k)} \|\phi_i(\boldsymbol{x})\|_{L^\infty(K)}}_{C_0} \times \sup_{\boldsymbol{x} \in K} \left| f(\boldsymbol{x}) - T^k_{\boldsymbol{p}} f(\boldsymbol{x}) \right|.
\end{aligned} \tag{4.82}$$

Inserting this in the estimate for the interpolation error:

$$\|f - \Pi^k_K f\|_{L^\infty(K)} \leq (1 + C_0) \left(\sup_{\boldsymbol{p} \in K} \sup_{\boldsymbol{x} \in K} \left| f(\boldsymbol{x}) - T^k_{\boldsymbol{p}} f(\boldsymbol{x}) \right| \right) \tag{4.83}$$

$$\leq (1 + C_0) \left(\sup_{\boldsymbol{p} \in K} \sup_{\boldsymbol{x} \in K} \frac{1}{(k+1)!} \left| \sum_{|\alpha|=k+1} D^\alpha f(\boldsymbol{p})(\boldsymbol{x} - \boldsymbol{p})^\alpha \right| \right) + o(\|\boldsymbol{x} - \boldsymbol{p}\|^{k+1})$$

In particular, the displacement $(\boldsymbol{x} - \boldsymbol{p})^\alpha$ with $|\alpha| = k+1$ relates to the terms h^{k+1} or $|K|^{\frac{k+1}{n}}$ typically appearing in estimators in integral norm and ensuring $\mathcal{O}(k+1)$ mesh convergence.

For practical applications, we must assume that the high-order derivatives do not vary too much on K, so that it is sufficient to evaluate these derivatives at a point $\boldsymbol{p}$, that is:

$$\|f - \Pi_K^k f\|_{L^\infty(K)} \leq \frac{(1+C_0)}{(k+1)!} \left(\sup_{\boldsymbol{x}\in K} \left| \sum_{|\alpha|=k+1} D^\alpha f(\boldsymbol{p})(\boldsymbol{x}-\boldsymbol{p})^\alpha \right| \right). \tag{4.84}$$

In practice, $\boldsymbol{p}$ is typically chosen as a point where the recovery of high-order derivatives is convenient, such as mesh vertices $\boldsymbol{x}_i$, where the methods of Zhu and Zienkiewicz [83, 84] or Zhang and Naga [85] can be applied. The leading error term depends on the $k+1$-tensor $D^\alpha f(\boldsymbol{p})$ of high-order derivatives, whose manipulation is impractical. To work in a setting similar to the linear case, the anisotropic measure of $D^\alpha f$ is introduced as a field of symmetric positive-definite matrices $\mathcal{Q}$ which is a tight upper bound on the derivatives of order $k+1$ of f, in the sense that at each $\boldsymbol{p}$, the following majoration holds:

$$\left| \sum_{|\alpha|=k+1} D^\alpha f(\boldsymbol{p})\, \boldsymbol{\xi}^\alpha \right| \leq \left(\boldsymbol{\xi}^T \mathcal{Q}_{\boldsymbol{p}} \boldsymbol{\xi} \right)^{\frac{k+1}{2}}, \qquad \forall\, \boldsymbol{\xi} \in \mathbb{R}^2, \tag{4.85}$$

where $\boldsymbol{\xi}^\alpha = (\xi^1)^{\alpha_1}(\xi^2)^{\alpha_2}$ in two dimensions. As $\mathcal{Q}_{\boldsymbol{p}}$ is SPD, it is the data of a Riemannian metric evaluated at the vertices $\boldsymbol{p}$, and observing that $\left(\boldsymbol{\xi}^T \mathcal{Q}_{\boldsymbol{p}} \boldsymbol{\xi} \right)^{\frac{k+1}{2}} = \left(\ell_{\mathcal{Q}_{\boldsymbol{p}}}(\boldsymbol{\xi}) \right)^{k+1}$, the upper bound above has a straightforward interpretation on a Riemannian manifold M: we seek the metric $\mathcal{Q}$ such that at each $\boldsymbol{p} \in M$ the length with respect to $\mathcal{Q}$ of the tangent vectors $\boldsymbol{\xi} \in T_{\boldsymbol{p}}M$ bounds the directional derivative of f along $\boldsymbol{\xi}$. The metric $\mathcal{Q}$ is presented in more details in Section 5.2, while Sections 5.4.1 and 5.4.2 detail its computation. In particular, a closed-form solution for $\mathcal{Q}_{\boldsymbol{p}}$ is available for quadratic interpolation [38]. Sampling the derivatives at $\boldsymbol{p} = \boldsymbol{x}_i$ and expressing $\boldsymbol{x} - \boldsymbol{x}_i$ in terms of the edges using barycentric coordinates u_j and the identity:

$$\boldsymbol{x} - \boldsymbol{x}_i = \sum_j u_j(\boldsymbol{x})(\boldsymbol{x}_j - \boldsymbol{x}_i) = \sum_j u_j \boldsymbol{e}_{ij}, \tag{4.86}$$

we write the estimate (4.84) in terms of $\mathcal{Q}_{\boldsymbol{p}}$ and the edges of K only:

$$\|f - \Pi_K^k f\|_{L^\infty(K)} \leq C \sum_{i,j=1}^{3} \left(\boldsymbol{e}_{ij}^T \mathcal{Q}(\boldsymbol{p}) \boldsymbol{e}_{ij} \right)^{\frac{k+1}{2}} \tag{4.87}$$

As both $\boldsymbol{e}_{ij}^T \mathcal{Q}_{\boldsymbol{p}} \boldsymbol{e}_{ij}$ and the exponent $(k+1)/2$ are positive, the latter can be applied to the sum, so that, similarly to (4.76), we obtain:

$$|e_K^k f(\boldsymbol{p})| = \|f - \Pi_K^k f\|_{L^\infty(K)} \leq C \left(\sum_{i,j=1}^{3} \boldsymbol{e}_{ij}^T \mathcal{Q}(\boldsymbol{p}) \boldsymbol{e}_{ij} \right)^{\frac{k+1}{2}}. \tag{4.88}$$

This finally yields the continuous error estimate for high-order interpolation [11, 12]:

$$|e_K^k f(\boldsymbol{p})| \leq C \left(\operatorname{tr}\left(\mathcal{M}_{\boldsymbol{p}}^{-1/2} \mathcal{Q}_{\boldsymbol{p}} \mathcal{M}_{\boldsymbol{p}}^{-1/2}\right)\right)^{\frac{k+1}{2}}, \tag{4.89}$$

and the error functional:

$$\mathcal{E}^p(\mathcal{M}) \leq C \int_\Omega \left(\operatorname{tr}\left(\mathcal{M}_{\boldsymbol{x}}^{-1/2} \mathcal{Q}(\boldsymbol{x}) \mathcal{M}_{\boldsymbol{x}}^{-1/2}\right)\right)^{\frac{p(k+1)}{2}} d\boldsymbol{x}. \tag{4.90}$$

4.5.2 Curved elements

We now turn to interpolation error on curved elements. To the best of our knowledge, interpolation error estimates on curved elements are only available in Rochery [17, 33] for linear interpolation and in Moxey et al. [48] for arbitrary orders of both interpolation and geometry. The estimates proposed by Moxey et al. take a form similar to the third step of (4.82) and involve the composition of the polynomial basis with the nonlinear reference-to-physical transformation. Although these estimates are generic for curved triangles, they feature an upper bound on the derivatives of f and are thus quite conservative. As a result, they predict a reduced order of convergence of $\lfloor k/q \rfloor + 1$ for interpolation error of degree k in L^∞ norm on arbitrarily curved meshes with geometry of degree q. This loss of convergence, already predicted as early as in Ciarlet and Raviart [52], then by Botti [47] for L^2-projection on finite element meshes, is observed in practice for meshes with arbitrary constant curvature. As our introductory example indicates, however, interpolation on appropriately curved meshes provides substantial gain in convergence order, even with reference-space interpolation basis. Furthermore, Rochery recently showed that curved unit triangles created with their one-step fixed-point method exhibit edge curvature that decreases as $\mathcal{O}(h^2)$, a necessary condition given in Ciarlet to maintain the optimal $k+1$ convergence rate for interpolation [33].

We derive continuous estimates starting on $K_\triangle$, using the estimate for linear triangles for the composition $f \circ F_{\triangle \to K}$. Notably, derivatives of all orders of the objective function f appear in the derivatives of the composition, and derivatives of order as low as $k+2-q$ remain in the error estimate. For linear interpolation, we present the continuous estimate obtained by Rochery [17]. This is, for now, the only result whose generalization to curved elements is straightforward, as higher-order estimates involve the anisotropic measure $\mathcal{Q}$, which is *not* a derivative operator. The result is the estimate (4.77), supplemented by a term involving the derivatives of the Riemannian metric. For higher-order interpolation, we modify (4.85) so that $\mathcal{Q}$ depends on the transformation $F_{\triangle \to K}$, hence on the metric. This yields a nonlinear error operator with an added implicit dependence on the Riemannian metric.

In this section, K is a curved unit triangle with respect to the Riemannian metric $\mathcal{M}$ and to the reference equilateral triangle $K_\triangle$. As we aim at writing the error in L^p norm on the computational domain, the isometric change of variable formula for $e_K^k f = f - \Pi_K^k f$ comes in handy:

$$\int_K e_K^k f \, dV_{\mathcal{M}} = \int_{K_\triangle} (e_K^k f \circ F_{\triangle \to K}) \, dy. \tag{4.91}$$

In the following, we simply note $F \triangleq F_{\triangle \to K}$ and we set $f_\triangle \triangleq f \circ F = f \circ F_{\triangle \to K}$, following [17,33], so that $f_\triangle$ is a real-valued function defined on $K_\triangle$. The composition $e_K^k f \circ F_{\triangle \to K}$ in (4.91) is an interpolation error on $K_\triangle$, since:

$$\begin{aligned} (e_K^k f) \circ F = (f - \Pi_K^k f) \circ F &= f(F) - \Pi_K^k f(F) \\ &= (f \circ F) - \Pi_{K_\triangle}^k (f \circ F) = e_{K_\triangle}^k (f_\triangle), \end{aligned} \tag{4.92}$$

thus the estimate derived for linear triangles applies to $f_\triangle$. From (4.91), the error on an element K is computed on $K_\triangle$ and depends on $F_{\triangle \to K}$. It is important to note that the function f and the isometry $F_{\triangle \to K}$ are not a priori related, so the error is not expected to be the same on different elements. Only when $F_{\triangle \to K}$ is defined by a metric tailored for the interpolation error can we expect the error to be equidistributed.

Linear interpolation

For $\boldsymbol{p} \in K$ and $\boldsymbol{y} \in K_\triangle$, the continuous error estimate writes, using (4.78):

$$e_K^1 f(\boldsymbol{p}) = e_{K_\triangle}^1 (f \circ F)(\boldsymbol{y}) \leq C \ \mathrm{tr}\left(|H_{f_\triangle}(\boldsymbol{y})| \right) = C \ \mathrm{tr}\left(\left| \frac{\partial^2 (f \circ F)}{\partial y^i \partial y^j}(\boldsymbol{y}) \right| \right). \tag{4.93}$$

Using the chain rule, the Hessian writes:

$$\begin{aligned} \frac{\partial^2 (f \circ F)}{\partial y^i \partial y^j} &= \frac{\partial^2 f}{\partial x^k \partial x^\ell}\bigg|_{F(\boldsymbol{y})} \frac{\partial F^\ell}{\partial y^j} \frac{\partial F^k}{\partial y^i} + \frac{\partial f}{\partial x^k}\bigg|_{F(\boldsymbol{y})} \frac{\partial^2 F^k}{\partial y^i \partial y^j} \\ \downarrow & \\ H_{f_\triangle}(\boldsymbol{y}) &= J_{\triangle \to K}^T H_f(F(\boldsymbol{y})) J_{\triangle \to K} + H_F(\boldsymbol{y}) \cdot \nabla f(F(\boldsymbol{y})), \end{aligned} \tag{4.94}$$

where ∇f here is the gradient of f, and not the total covariant derivative. For quadratic and higher-order triangles, the Hessian of the transformation H_F does not vanish, hence the gradient of f remains, contrary to linear elements where only the $k+1$-th derivatives are involved. For interpolation of degree k on $\mathcal{P}^q$ elements, the chain rule for $f \circ F$ (also known

as Faà di Bruno's formula) shows the derivatives of order as low as $k+1-(q-1)$ remain. As K is unit w.r.t. $K_\triangle$ and $\mathcal{M}$, its Jacobian matrix writes $J_{\triangle\to K} = \mathcal{M}^{-1/2}R$ for some rotation R. Taking the trace of the first term of $H_{f_\triangle}(\boldsymbol{y})$ yields:

$$\begin{aligned}\operatorname{tr}\left(J_{\triangle\to K}^T H_f J_{\triangle\to K}\right) &= \operatorname{tr}\left(R^T\mathcal{M}^{-1/2}H_f\mathcal{M}^{-1/2}R\right)\\ &= \operatorname{tr}\left(\mathcal{M}^{-1/2}H_f\mathcal{M}^{-1/2}\right),\end{aligned} \tag{4.95}$$

since R does not change the trace. For the second term, the Hessian of the transformation writes:

$$\begin{aligned}(H_F(\boldsymbol{y}))_{ij}^k &= \frac{\partial^2 F^k}{\partial y^i \partial y^j} = \frac{\partial}{\partial y^j}(J_{\triangle\to K})_{ki} = \frac{\partial}{\partial y^j}\left[(\mathcal{M}^{-1/2}\circ F)_{kl}R_{li}\right]\\ &= \left.\frac{\partial \mathcal{M}_{kl}^{-1/2}}{\partial x^m}\right|_{F(\boldsymbol{y})} \underbrace{\frac{\partial F^m}{\partial y^j}}_{=(J_{\triangle\to K})_{mj}} \quad R_{li} = \left.\frac{\partial \mathcal{M}_{kl}^{-1/2}}{\partial x^m}\right|_{F(\boldsymbol{y})}\left[\mathcal{M}_{mp}^{-1/2}R_{pj}\right](R_{il})^T,\end{aligned} \tag{4.96}$$

so that its trace is:

$$\operatorname{tr}\left(H_F\cdot\nabla f\right) = \operatorname{tr}(H_{F,ij}^k\partial_k f) = H_{F,ii}^k\partial_k f = \left.\frac{\partial \mathcal{M}_{kl}^{-1/2}}{\partial x^m}\right|_{F(\boldsymbol{y})}\mathcal{M}_{mp}^{-1/2}\underbrace{R_{pi}(R_{il})^T}_{=\delta_{pl}}\frac{\partial f}{\partial x^k} \tag{4.97}$$

$$= \mathcal{M}_{ml}^{-1/2}\left.\frac{\partial \mathcal{M}_{lk}^{-1/2}}{\partial x^m}\right|_{F(\boldsymbol{y})}\frac{\partial f}{\partial x^k}. \tag{4.98}$$

For a more compact notation, we define $(\partial\mathcal{M}^{-1/2})_{ijk} \triangleq \partial\mathcal{M}_{ij}^{-1/2}/\partial x^k$ and the double contraction $A:B \triangleq A_{ijk}B_{jk}$. As $\mathcal{M}^{-1/2}$ is symmetric, the trace writes:

$$\operatorname{tr}\left(H_F\nabla\cdot f\right) = (\partial\mathcal{M}^{-1/2})_{klm}\mathcal{M}_{lm}^{-1/2}\frac{\partial f}{\partial x^k} = (\partial\mathcal{M}^{-1/2} : \mathcal{M}^{-1/2})\cdot\nabla f. \tag{4.99}$$

It depends only on the Riemannian metric, its derivatives and the derivatives of f. The pointwise continuous error operator for linear interpolation on K thus reads:

$$\begin{aligned}|e_K^{(1)}f(\boldsymbol{p})| &\le C\left|\operatorname{tr}\left(\mathcal{M}_{\boldsymbol{p}}^{-1/2}H_f(\boldsymbol{p})\mathcal{M}_{\boldsymbol{p}}^{-1/2}\right) + \operatorname{tr}\left(H_F\cdot\nabla f\right)\right|\\ &= C\left|\operatorname{tr}\left(\mathcal{M}_{\boldsymbol{p}}^{-1/2}H_f(\boldsymbol{p})\mathcal{M}_{\boldsymbol{p}}^{-1/2}\right) + (\partial\mathcal{M}_{\boldsymbol{p}}^{-1/2} : \mathcal{M}_{\boldsymbol{p}}^{-1/2})\cdot\nabla f\right|,\end{aligned} \tag{4.100}$$

yielding the following error functional in L^p norm:

$$\mathcal{E}^p(\mathcal{M}) \le C\int_\Omega \left|\operatorname{tr}\left(\mathcal{M}_{\boldsymbol{x}}^{-1/2}H_f(\boldsymbol{x})\mathcal{M}_{\boldsymbol{x}}^{-1/2}\right) + (\partial\mathcal{M}^{-1/2}(\boldsymbol{x}) : \mathcal{M}_{\boldsymbol{x}}^{-1/2})\cdot\nabla f(\boldsymbol{x})\right|^p dx. \tag{4.101}$$

When the metric is constant, we recover the estimate (4.77) for linear elements. This agrees with the definition of unit elements as isometries, yielding only linear triangles for a constant metric.

Quadratic and higher interpolation

The main obstacle for extending (4.89) to curved elements is that, contrary to the Hessian in (4.77), the matrix $\mathcal{Q}$ is not a derivative operator. For instance, for quadratic interpolation and $\boldsymbol{y} \in K_\triangle$,

$$\mathcal{Q}_{\boldsymbol{y}} \neq \left| \frac{\partial^3 (f \circ F)}{\partial y^i \partial y^j \partial y^k}(\boldsymbol{y}) \right|. \tag{4.102}$$

To circumvent this, we propose to consider in the reference space the anisotropic measure $\mathcal{Q}_\triangle$ of the $k+1$-th derivatives $f \circ F$ defined at $\boldsymbol{y} \in K_\triangle$ by:

$$\left| \sum_{|\alpha|=k+1} D^\alpha (f \circ F(\mathcal{M}))(\boldsymbol{y})\, \boldsymbol{\zeta}^\alpha \right| \leq \left(\boldsymbol{\zeta}^T \mathcal{Q}_\triangle(\boldsymbol{y}) \boldsymbol{\zeta} \right)^{\frac{k+1}{2}}, \qquad \forall\, \boldsymbol{\zeta} \in \mathbb{R}^2, \tag{4.103}$$

where the nonlinear transformation $F(\mathcal{M})$ depends on the metric. In other words, when seeking the metric minimizing the L^p norm of the error on the domain, the anisotropic measures are no longer fixed data of the minimization problem, but must be solved for alongside the rest of the error functional. The local continuous estimate thus reads, similarly to linear elements:

$$|e_K^k f(\boldsymbol{p})| = |e_{K_\triangle}^k (f \circ F)(\boldsymbol{y})| \leq C \, \left(\operatorname{tr} \mathcal{Q}_\triangle(\mathcal{M}) \right)^{\frac{k+1}{2}}, \tag{4.104}$$

Although the problem (4.103) is formulated in the reference space, each term can be evaluated in the physical space after substituting the Jacobian matrix $J_{\triangle \to K} = \mathcal{M}^{-1/2} R$, as we show below, so that we effectively compute a matrix field $\mathcal{Q}_{\boldsymbol{p}}(\mathcal{M})$ in the physical space, whose associated continuous error estimate is:

$$|e_K^k f(\boldsymbol{p})| \leq C \, \left(\operatorname{tr} \left(\mathcal{M}_{\boldsymbol{p}}^{-1/2} \mathcal{Q}_{\boldsymbol{p}}(\mathcal{M}) \mathcal{M}_{\boldsymbol{p}}^{-1/2} \right) \right)^{\frac{k+1}{2}}, \tag{4.105}$$

where $\mathcal{Q}_{\boldsymbol{p}}(\mathcal{M})$ hides the added complexity of the high-order geometry. The associated error functional is:

$$\mathcal{E}^p(\mathcal{M}) \leq C \int_\Omega \left(\operatorname{tr} \left(\mathcal{M}_{\boldsymbol{x}}^{-1/2} \mathcal{Q}_{\boldsymbol{x}}(\mathcal{M}) \mathcal{M}_{\boldsymbol{x}}^{-1/2} \right) \right)^{\frac{p(k+1)}{2}} d\boldsymbol{x}. \tag{4.106}$$

To illustrate, we consider the formulation of the upper bound $\mathcal{Q}$ for quadratic interpolation. Higher-order interpolants are treated similarly, but the number of terms in the chain rule

quickly becomes cumbersome to write in full form. Writing $J = J_{\triangle \to K}$, the chain rule yields:

$$\begin{aligned}\frac{\partial^3 (f \circ F)}{\partial y^i \partial y^j \partial y^k}(\boldsymbol{y}) &= \left.\frac{\partial^3 f}{\partial x^m \partial x^n \partial x^p}\right|_{F(\boldsymbol{y})} \frac{\partial F^m}{\partial y^i}\frac{\partial F^n}{\partial y^j}\frac{\partial F^p}{\partial y^k} \\ &+ \left.\frac{\partial^2 f}{\partial x^m \partial x^n}\right|_{F(\boldsymbol{y})} \left(\frac{\partial F^n}{\partial y^j}\frac{\partial^2 F^m}{\partial y^i \partial y^k} + \frac{\partial F^m}{\partial y^i}\frac{\partial^2 F^n}{\partial y^j \partial y^k} + \frac{\partial F^n}{\partial y^k}\frac{\partial^2 F^m}{\partial y^i \partial y^j}\right) \\ &+ \left.\frac{\partial f}{\partial x^m}\right|_{F(\boldsymbol{y})} \frac{\partial^3 F^m}{\partial y^i \partial y^j \partial y^k} \\ &= C_{f,mnp}(F(\boldsymbol{y}))J_{mi}J_{nj}J_{pk} + \sum_{\substack{\text{cyclic}\\ i,j,k}} H^m_{F,ij} H_{f,mn}(F(\boldsymbol{y}))J_{nk} + \nabla f(F(\boldsymbol{y}))_m C^m_{F,ijk},\end{aligned} \tag{4.107}$$

with $C^l_{F,ijk} \triangleq \partial_{ijk} F^l$ the third derivatives of the transformation F, which vanish for quadratic triangles. The cyclic sum acts on the three permutations ijk, kij and jki. Substituting the metric and its derivatives appearing in the derivatives of F, the contraction with arbitrary vectors $\boldsymbol{\zeta}$ in (4.103) reads:

$$\begin{aligned}\sum_\alpha D^\alpha (f \circ F(\mathcal{M}))\,\boldsymbol{\zeta}^\alpha = \Bigg(& C_{f,mnp}(F(\boldsymbol{y}))\mathcal{M}^{-\frac{1}{2}}_{mr}\mathcal{M}^{-\frac{1}{2}}_{ns}\mathcal{M}^{-\frac{1}{2}}_{pt} R_{ri}R_{sj}R_{tk} \\ & + \sum_{\substack{\text{cyclic}\\ i,j,k}} H_{f,mn}(F(\boldsymbol{y}))\mathcal{M}^{-\frac{1}{2}}_{ns}\mathcal{M}^{-\frac{1}{2}}_{qp}\partial_q \mathcal{M}^{-\frac{1}{2}}_{ml} R_{li}R_{pj}R_{sk} \\ & + \nabla f(F(\boldsymbol{y}))_l \mathcal{M}^{-\frac{1}{2}}_{st}\left(\mathcal{M}^{-\frac{1}{2}}_{mp}\partial_{ms}\mathcal{M}^{-\frac{1}{2}}_{lq} + \partial_m \mathcal{M}^{-\frac{1}{2}}_{lq}\partial_s \mathcal{M}^{-\frac{1}{2}}_{mp}\right) R_{qi}R_{pj}R_{tk}\Bigg)\zeta^i\zeta^j\zeta^k.\end{aligned} \tag{4.108}$$

The transformation $F(\boldsymbol{y})$ appears explicitly only at the evaluation points of the physical space derivatives. It follows that all the quantities in parentheses above can be evaluated in the physical space, so that only the derivatives of the unknown transformation, written in terms of the metric, remain. Renaming the dummy indices, the right-hand side writes:

$$\underbrace{\begin{aligned}\Bigg(& C_{f,mno}(\boldsymbol{p})\mathcal{M}^{-\frac{1}{2}}_{mq}\mathcal{M}^{-\frac{1}{2}}_{np}\mathcal{M}^{-\frac{1}{2}}_{ot} + \sum_{\substack{\text{cyclic}\\ q,p,t}} H_{f,mn}(\boldsymbol{p})\mathcal{M}^{-\frac{1}{2}}_{nt}\mathcal{M}^{-\frac{1}{2}}_{op}\partial_o \mathcal{M}^{-\frac{1}{2}}_{mq} \\ & + \nabla f(\boldsymbol{p})_l \mathcal{M}^{-\frac{1}{2}}_{st}\left(\mathcal{M}^{-\frac{1}{2}}_{mp}\partial_{ms}\mathcal{M}^{-\frac{1}{2}}_{lq} + \partial_m \mathcal{M}^{-\frac{1}{2}}_{lq}\partial_s \mathcal{M}^{-\frac{1}{2}}_{mp}\right)\Bigg)\end{aligned}}_{\triangleq\, T_{qpt}} (R_{qi}\zeta^i)(R_{pj}\zeta^j)(R_{tk}\zeta^k). \tag{4.109}$$

As the vectors $\boldsymbol{\zeta}$ are arbitrary, the $R_{qi}\zeta^i$ are also the components of an arbitrary vector $\widetilde{\boldsymbol{\zeta}}$, and can be renamed $\widetilde{\zeta}^q$. Arbitrary tangent vectors $\boldsymbol{\xi} \in T_{F(\boldsymbol{y})}M$ in the physical space are obtained from the linear transformation $\boldsymbol{\xi} = \mathcal{M}^{-1/2}_{F(\boldsymbol{y})}\boldsymbol{\zeta}$, and conversely $\boldsymbol{\zeta} = \mathcal{M}^{1/2}_{\boldsymbol{p}}\boldsymbol{\xi}$ at $\boldsymbol{p} = F(\boldsymbol{y})$. In the

expression of T_{qpt} above, each term features $\mathcal{M}_{F(\boldsymbol{y})}^{-1/2}$ three times, either explicitly or hidden in each derivative of the metric. Each instance transforms a coordinate ζ^q in the reference space to a coordinate ξ^q in the physical space, so that the problem (4.103) initially formulated in the reference space is also fully defined in the physical space:

$$\begin{aligned} \left|T_{ijk}(F(\boldsymbol{y}))\,\widetilde{\zeta}^i\widetilde{\zeta}^j\widetilde{\zeta}^k\right| &\leq \left(\boldsymbol{\zeta}^T\mathcal{Q}_{\triangle}(\boldsymbol{y})\boldsymbol{\zeta}\right)^{\frac{k+1}{2}}, \qquad \forall\,\boldsymbol{\zeta}\in\mathbb{R}^2. \\ \left|\widetilde{T}_{ijk}(\boldsymbol{p})\,\xi^i\xi^j\xi^k\right| &\leq \left(\boldsymbol{\xi}^T\mathcal{Q}(\boldsymbol{p})\boldsymbol{\xi}\right)^{\frac{k+1}{2}}, \qquad \forall\,\boldsymbol{\xi}\in T_{F(\boldsymbol{y})}M, \end{aligned} \tag{4.110}$$

where $\widetilde{T}$ is the tensor T after the three instances of $\mathcal{M}_{F(\boldsymbol{y})}^{-1/2}$ have been factored. Since $\boldsymbol{\xi} = \mathcal{M}_{F(\boldsymbol{y})}^{-1/2}\boldsymbol{\zeta}$, the solutions to each problem are related by pullback by $\mathcal{Q}_{\triangle}(\boldsymbol{y}) = \mathcal{M}_{F(\boldsymbol{y})}^{-1/2}\mathcal{Q}_{\boldsymbol{p}}\mathcal{M}_{F(\boldsymbol{y})}^{-1/2}$, as expected, or conversely $\mathcal{Q}(\boldsymbol{p}) = \mathcal{M}_{\boldsymbol{p}}^{1/2}\mathcal{Q}_{\triangle}\mathcal{M}_{\boldsymbol{p}}^{1/2}$. In Chapter 5, we describe how to compute $\mathcal{Q}$ based only on the components of the tensor T: we can provide $T_{ijk}(F(\boldsymbol{y}))$ to compute $\mathcal{Q}_{\triangle}$ directly, thus avoiding the computation of the metric derivatives required by $\widetilde{T}_{ijk}(\boldsymbol{p})$. For linear elements, we have trivially:

$$\begin{aligned} \left|\frac{\partial^3(f\circ F)}{\partial y^i\partial y^j\partial y^k}(\boldsymbol{y})\zeta^i\zeta^j\zeta^k\right| &\leq \left(\boldsymbol{\zeta}^T\mathcal{Q}_{\triangle}(\boldsymbol{y})\boldsymbol{\zeta}\right)^{\frac{3}{2}}, \qquad \forall\,\boldsymbol{\zeta}\in\mathbb{R}^2 \\ &\downarrow \\ \left|\frac{\partial^3(f\circ F)}{\partial y^i\partial y^j\partial y^k}(\boldsymbol{y})\zeta^i\zeta^j\zeta^k\right| &= \left|\frac{\partial^3 f}{\partial x^m\partial x^n\partial x^o}(F(\boldsymbol{y}))\mathcal{M}_{mq}^{-\frac{1}{2}}\mathcal{M}_{np}^{-\frac{1}{2}}\mathcal{M}_{ot}^{-\frac{1}{2}}(R_{qi}\zeta^i)(R_{pj}\zeta^j)(R_{tk}\zeta^k)\right| \\ &= \left|\frac{\partial^3 f}{\partial x^m\partial x^n\partial x^o}(\boldsymbol{p})\,\underbrace{\mathcal{M}_{mq}^{-\frac{1}{2}}\widetilde{\zeta}^q}_{\xi^m}\underbrace{\mathcal{M}_{np}^{-\frac{1}{2}}\widetilde{\zeta}^p}_{\xi^n}\underbrace{\mathcal{M}_{ot}^{-\frac{1}{2}}\widetilde{\zeta}^t}_{\xi^o}\right| \\ &\leq \left(\boldsymbol{\xi}^T\mathcal{Q}(\boldsymbol{p})\boldsymbol{\xi}\right)^{\frac{3}{2}}, \qquad \forall\,\boldsymbol{\xi}\in T_{F(\boldsymbol{y})}M=\mathbb{R}^2, \end{aligned} \tag{4.111}$$

so that the components of $\widetilde{T}_{ijk}(\boldsymbol{p})$ are simply $\partial_{ijk}f$. In particular, solving (4.103) yields the same continuous error operator as (4.89), which motivates its use to compute error bounds for interpolation of degree $k\geq 2$ on curved elements.

4.6 Conclusion

Unit elements, the central ingredient of the continuous mesh framework, were discussed in this chapter. We reviewed the current definitions of unit simplices and exhibited their shortcomings, then we introduced a new definition based on Riemannian isometries. This definition naturally englobes the current ones when unit elements are defined as isometries of the regular simplex $K_{\triangle}$. In addition, isometries open the way to other choices of reference

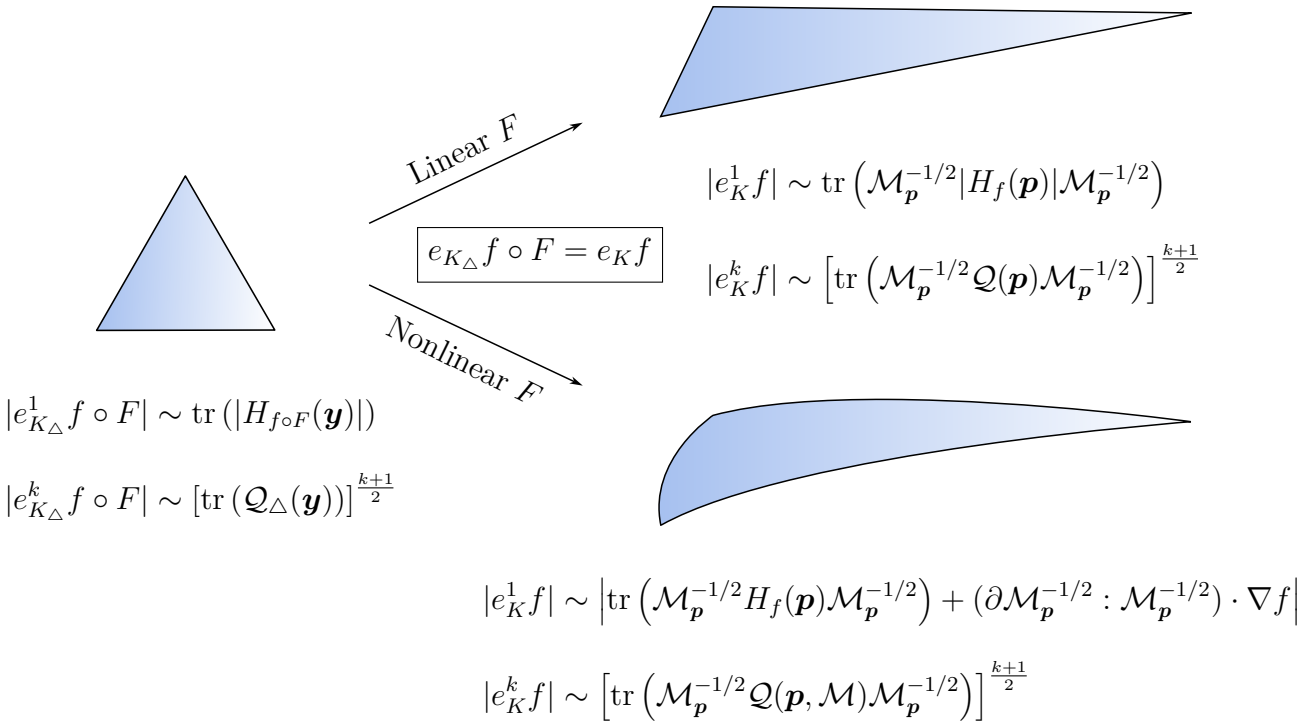


Figure 4.23 Summary of the presented error estimates on linear and curved triangles.

elements, such as the reference right-angled simplex $K_\triangleright$. Proofs of concept of isometric triangulations with respect to non-Euclidean metrics were obtained by optimizing a starting mesh at fixed connectivity, with the goal of creating meshes isometric to uniform tilings of $\mathbb{R}^2$, made of copies of either $K_\triangle$ or $K_\triangleright$. As the reference simplices are defined in the Euclidean space, the sought isometries only exist between $\mathbb{R}^n$ and flat manifolds, whose Gaussian curvature vanishes everywhere. As a result, isometric triangulations only exist in 2D for developable surfaces, a very restrictive class of surfaces, but useful nonetheless for a visual representation of a perfectly unit mesh. The triangulations on these surface are made of geodesic edges and preserve the lengths, angles and areas of their counterpart in $\mathbb{R}^2$.

Continuous operators involving only the metric were derived for the interpolation error on unit elements. Existing estimates were reviewed, and we proposed an extension to curved elements of the anisotropic measure, bounding the high-order derivatives appearing in error estimates for interpolation of order $k \geq 2$, This way, continuous error bounds can be computed for all interpolation degrees and on simplicial geometries of arbitrary order.

In differential geometry, *Killing vector fields* are the infinitesimal generators of (nonreflection) isometries. For instance, the Killing vectors of $\mathbb{R}^2$ are the fields $(1, 0)$, $(0, 1)$ and $(-y, x)$. Their *flow* respectively generates the translations along the x and y-axis, and the rotations around

the origin. As isometric unit elements are isometries of one another, future developments could explore using Killing vectors to transport unit elements isometrically directly in the physical space.

CHAPTER 5 OPTIMAL METRICS FOR ANISOTROPIC ADAPTATION WITH LINEAR TRIANGLES

Metric-based mesh adaptation is a powerful framework, because it allows to work directly with the abstract tool of a Riemannian metric on which continuous tools are well-defined. Differentiation and optimization can be performed with respect to the metric, then a discrete mesh is selected from the equivalence class of unit meshes by a mesh generator. Finding the optimal Riemannian metric for a given adaptation target is thus a key step of metric-based mesh adaptation. Interpolation error, with which we are concerned in this thesis, is a straightforward target for anisotropic mesh adaptation, and has indeed been the topic of extensive research in the past two decades. Limiting the interpolation error offers a global control on the fields of interest and the adapted meshes capture their variations throughout the domain. Optimal metrics minimizing the interpolation error in integral norms (mainly L^p and $W^{1,p}$) are available on linear meshes and are reviewed in this chapter. For industrial applications, the global accuracy of the resolved fields matters less, rather, the adaptation target is a local functional of the simulation variables, also called *output functionals* or *quantities of interest* [1,31,86,87]. Lift and drag are typical examples of quantities of interest for aerodynamics computations. Adaptation driven by a quantity of interest is often called *goal-oriented mesh adaptation*, whereas adaptation driven by the control of interpolation error is sometimes referred to as *feature-based mesh adaptation* [88]. Optimal metrics for goal-oriented mesh adaptation minimize the error over the target functional and require solving an adjoint problem to the PDE of interest. Their expression typically features the optimal metric for interpolation error weighted by the solution of the adjoint problem [1], so that the metrics minimizing the interpolation error discussed in this chapter also find a use for goal-oriented mesh adaptation.

In this chapter, we place these optimal metrics in a broader context and we study optimal linear triangulations with respect to the interpolation error of degree k. They are defined as the triangulations minimizing the L^p norm or the $W^{1,p}$ seminorm of the interpolation error from among all triangulations of a given maximal number N of elements or vertices. Such triangulations can be studied from a discrete point of view, by considering local shape optimization problems for the triangles, or from a continuous point of view, by manipulating the Riemannian metrics for which these triangulations are unit. Discrete error estimates study the interpolation error over arbitrary triangulations. They reveal that the optimal triangulations are those whose triangles are aligned with the directions of anisotropy of the

function f of interest. Discrete estimates yield existence results: they guarantee the existence of optimal triangulations exhibiting e.g. $\mathcal{O}(N^{-(k+1)/n}) \sim \mathcal{O}(h^{k+1})$ convergence for the interpolation error in L^p norm. They do not, in general, provide a practical way of constructing such triangulations, however. On the other hand, continuous estimates involve Riemannian metrics, for which it is nowadays straightforward to generate unit or quasi-unit meshes with any metric-based meshing software, and are thus constructive results. Continuous error estimates relate the error on a unit mesh to its metric: the metric can be arbitrary and needs not be adapted to f. In agreement with the discrete case, the optimal metrics are the ones whose principal directions match the anisotropy of f. The major advantage of the continuous approach is that one can derive the optimal Riemannian metrics by direct minimization of the continuous error estimate. These metrics are of the form:

$$\mathcal{M} = c\,(\det \mathcal{Q})^{\alpha}\mathcal{Q}, \tag{5.1}$$

where $\mathcal{Q}$ is a Riemannian metric that describes the anisotropy of f, α is a real number depending on the norm to minimize, the degree of the interpolation and the space dimension, and $c > 0$ controls the number of elements in the adapted mesh. These are the state-of-the-art Riemannian metrics for interpolation-based and metric-based mesh adaptation on linear triangles. Unit meshes generated from these metrics exhibit the optimal convergence rate predicted by the discrete estimates. In other words, the optimal triangulations described by discrete estimates are precisely unit triangulations with respect to the optimal metrics described by continuous estimates. The discrete estimates presented here are due to Chen et al. [75, 76] and Mirebeau [38, 80], and the continuous estimates are due to Cao [77–79] and Loseille et al. [1, 11, 55]. Their main results are compared in a systematic manner.

This chapter is organized as follows. We first present in Section 5.1 discrete estimates for arbitrary triangulations without considering metrics. They involve the geometry of the triangles and rely on a shape optimization problem. Moreover, they show that optimal convergence (that is, of rate $k+1-s$ in $W^{s,p}$ seminorm) with respect to the number of mesh elements N is only possible if the triangles are shaped according to the higher-order derivatives of f, and are thus anisotropic. Continuous error estimates, presented in Section 5.3, involve a tight upper bound $\mathcal{Q}$ on the high-order derivatives of f, called the anisotropic measure and already introduced in Section 4.5. This upper bound is formally defined and studied in Section 5.2. While $\mathcal{Q}$ is exactly the absolute Hessian for linear interpolation, obtaining an expression for the upper bound is not trivial for interpolants of arbitrary order. In Section 5.4, we detail the analytic solution proposed by Mirebeau for quadratic interpolants. Cubic and higher in-

terpolants must be tackled numerically, and we detail the log-simplex method from Coulaud and Loseille [11, 12]. Numerical examples are finally presented in Section 5.5. We present sequences of anisotropic triangulations adapted from the optimal Riemannian metrics, which exhibit the convergence rate of $\mathcal{O}(N^{-(k+1)/n})$ predicted by both the discrete and continuous estimates. These triangulations are adapted to both analytic fields and solutions of PDEs.

Notations and definitions

In this chapter, $n = 2$ is the space dimension and $\Omega \subset \mathbb{R}^2$ is the bounded physical space. We consider conforming linear triangulations $\mathcal{T}$ of Ω, and denote by $\#(\mathcal{T})$ the number of elements in $\mathcal{T}$. We consider Lagrange interpolation of degree k over each triangle, so that the **Lagrange interpolation operator on $\mathcal{T}$**, simply noted Π^k, is defined for a smooth function $f : \Omega \to \mathbb{R}$ by its restriction $\Pi^k f|_K \triangleq \Pi^k_K f$ on each triangle $K \in \mathcal{T}$. Following Mirebeau, we set $m \triangleq k + 1$ and we define the **space of homogeneous polynomials of degree** m in two variables similarly to Chapter 2 by:

$$\mathbb{H}^m \triangleq \Big\{ \text{ span } x^i y^j \mid x, y \in \mathbb{R},\ i + j = m \Big\}. \tag{5.2}$$

The polynomials $\pi \in \mathbb{H}^m$ satisfy $\pi(\lambda \boldsymbol{z}) = \lambda^m \pi(\boldsymbol{z})$, for any $\boldsymbol{z} \in \mathbb{R}^2$ and $\lambda \in \mathbb{R}$. We slightly abuse the notation and write both $\pi(\boldsymbol{z})$ for $\boldsymbol{z} \in \mathbb{R}^2$ as well as $\pi(x, y)$ for $x, y \in \mathbb{R}$, but it is understood that polynomials in $\mathbb{H}^m$ are always bivariate. For a smooth function f, we denote by $d^{(m)} f_{\boldsymbol{p}}(\boldsymbol{\xi})$ the leading error term in Taylor's expansion of f at $\boldsymbol{p}$ in the direction $\boldsymbol{\xi} = \boldsymbol{x} - \boldsymbol{p}$ without the $1/m!$ term:

$$d^{(m)} f_{\boldsymbol{p}}(\boldsymbol{\xi}) \triangleq \sum_{|\alpha|=m} D^\alpha f(\boldsymbol{p}) \boldsymbol{\xi}^\alpha = \sum_{|\alpha|=m} D^\alpha f(\boldsymbol{p}) (\boldsymbol{x} - \boldsymbol{p})^\alpha. \tag{5.3}$$

Since we work in two dimensions and to avoid confusion between superscripts, the coordinates of a point $\boldsymbol{p}$ are noted x_p and y_p, hence we write $\boldsymbol{x} - \boldsymbol{p} = (x - x_p, y - y_p) \triangleq (x^1 - p^1, x^2 - p^2)$. At each $\boldsymbol{p} \in \Omega$, $d^{(m)} f_{\boldsymbol{p}}$ is a homogeneous polynomial of degree m in $\boldsymbol{\xi}$ and we refer to it as $\pi_{\boldsymbol{p}} \in \mathbb{H}^m$. It can be written in the compact form:

$$\pi_{\boldsymbol{p}}(\boldsymbol{\xi}) \triangleq d^{(m)} f_{\boldsymbol{p}}(\boldsymbol{\xi}) = (\boldsymbol{\xi} \cdot \nabla)^m f(\boldsymbol{p}), \tag{5.4}$$

or explicitly:

$$\pi_{\boldsymbol{p}}(x - x_p, y - y_p) = \sum_{i+j=m} \binom{m}{i} \frac{\partial^m f}{\partial x^i \partial y^j}(\boldsymbol{x}_0)(x - x_p)^i (y - y_p)^j, \quad \text{with } \binom{m}{i} = \frac{m!}{i!\, j!}. \tag{5.5}$$

If $\|\boldsymbol{\xi}\| = 1$, then $\pi_{\boldsymbol{p}}(\boldsymbol{\xi})$ is the directional derivative of order m of f at $\boldsymbol{p}$ along $\boldsymbol{\xi}$.

Interpolation error is measured in L^p norms and $W^{s,p}$ seminorms. These norms are defined for integer $1 \leq p \leq \infty$ and $s \geq 0$ by:

$$\|f\|_{L^p(\Omega)} = \left(\int_\Omega f^p \, dx\right)^{\frac{1}{p}}, \qquad |f|_{W^{s,p}(\Omega)} = \left(\sum_{|\alpha|=s} \|D^\alpha f\|^p_{L^p(\Omega)}\right)^{\frac{1}{p}}, \tag{5.6}$$

with the usual modification when $p = \infty$. In particular, the L^p norm and the $W^{1,p}$ seminorm of the interpolation error $e^k_{\mathcal{T}} f = f - \Pi^k f$ on a triangulation $\mathcal{T}$ write:

$$\|e^k_{\mathcal{T}} f\|_{L^p(\mathcal{T})} = \left(\sum_{K\in\mathcal{T}} \|f - \Pi^k_K f\|^p_{L^p(K)}\right)^{\frac{1}{p}}, \quad |e^k_{\mathcal{T}} f|_{W^{1,p}(\mathcal{T})} = \left(\sum_{K\in\mathcal{T}} \|\nabla(f - \Pi^k_K f)\|^p_{L^p(K)}\right)^{\frac{1}{p}}. \tag{5.7}$$

Several inequalities involve strictly positive constants: we use the same constants c or C, but it is understood that the constants differ for each inequality. The real number τ, originating from Hölder's inequality, is defined by:

$$\frac{1}{\tau} \triangleq \frac{m-s}{n} + \frac{1}{p}, \qquad \text{that is,} \qquad \tau = \frac{pn}{p(m-s)+n}. \tag{5.8}$$

Lastly, we denote by Sym_2 the set of symmetric 2×2 matrices with real coefficients and by $\mathrm{Sym}_2^+ \subset \mathrm{Sym}_2$ the subset of symmetric positive-definite matrices.

5.1 Discrete error estimates in $W^{s,p}$ norm

We first review estimates for the interpolation error based only on the triangulation, i.e., the discrete mesh, and the function to approximate, without relying on a Riemannian metric. As a starting point, we consider the problem of finding an optimal triangulation proposed by Mirebeau [38, 80]. It is formulated as follows: for a given norm X and interpolation degree k, find among all triangulations with a prescribed maximum number of elements N the triangulation that minimizes the norm of the interpolation error $e^k_{\mathcal{T}} f$. Thus, we seek the triangulation solving the minimization problem:

$$\min_{\#(\mathcal{T})\leq N} |f - \Pi^k f|_{X(\mathcal{T})}, \tag{5.9}$$

where X is either the $L^p = W^{0,p}$ norm or the $W^{1,p}$ seminorm. To study (5.9), Mirebeau provided asymptotic estimates for $|f - \Pi^k f|_{X(\mathcal{T})}$ and established the properties of the triangulations on which these estimates are reached. These asymptotic results are discussed in

the next two subsections.

5.1.1 Error bounds for the L^p norm

For the special case of the L^p norm and linear ($k = 1$) interpolation, Chen et al. [75] established the upper bound:

$$\|f - \Pi^1 f\|_{L^p(\mathcal{T})} \leq C N^{-\frac{2}{n}} \left\| |\det H_f|^{\frac{1}{n}} \right\|_{L^\tau(\Omega)}, \tag{5.10}$$

which is the anisotropic analog of classical isotropic estimates of the form ([89], Cor. 1.109):

$$\|f - \Pi^k f\|_{L^p(\mathcal{T})} \leq C h^{k+1} |f|_{W^{k+1,p}(\Omega)}. \tag{5.11}$$

The quantity $N^{-\frac{1}{n}}$ plays a role analog to the maximum mesh size h of isotropic meshes. In particular, (5.10) shows that a sequence of triangulations $\mathcal{T}_N$ exhibits $\mathcal{O}(N^{-1})$ (i.e., second order) convergence if the quantity $\sqrt{|\det H_f|}$, depending nonlinearly on the Hessian of f, remains controlled. This estimate motivated the development of mesh adaptation procedures based on the Hessian. Indeed, the results hereafter show that the optimal triangulations should be constructed so as (i) to equidistribute the L^p interpolation error on the triangles, and (ii) to feature triangles which are anisotropic in the usual Euclidean metric, but isotropic in a metric based on H_f. More precisely, it is shown further that the area of the triangles is obtained from the local value of $\det H_f$, while their aspect ratio and orientation are obtained from the eigenvalues and eigenvectors of H_f, respectively.

The bound (5.10) is limited to linear interpolation and to the L^p norm. In his thesis, Mirebeau extended this result to higher-order interpolation and to the $W^{1,p}$ seminorm, including the $H^1 = W^{1,2}$ seminorm that naturally appears when treating elliptic PDEs. We begin with the L^p norm, and define the **shape function** of a polynomial $\pi \in \mathbb{H}^m$ as the real number $S_{m,p}(\pi)$ that is the solution of the following shape optimization problem:

$$S_{m,p}(\pi) \triangleq \inf_{|K|=1} \|\pi - \Pi^k \pi\|_{L^p(K)}, \tag{5.12}$$

where the minimum is taken among all triangles of unit area. The shape function $S_{m,p}(\pi)$ describes the shape of the triangles that are best adapted to the polynomial π, that is, that minimize the interpolation error with respect to π in L^p norm. Since π is homogeneous, $S_{m,p}(\pi)$ actually also describes the optimal triangles of arbitrary area, by scaling. For a given point $\boldsymbol{p} \in \Omega$, determining the shape function of $\pi_{\boldsymbol{p}}$ is a local approximation of the problem

(5.9). Indeed, using Taylor's expansion $f \simeq P + \pi_{\boldsymbol{p}}$ on a triangle K containing $\boldsymbol{p}$ for a polynomial $P \in \mathbb{P}^k(K)$, the interpolation error for f can be replaced by the interpolation error for $\pi_{\boldsymbol{p}}$:

$$f - \Pi^k f \simeq P + \pi_{\boldsymbol{p}} - \Pi^k(P + \pi_{\boldsymbol{p}}) = \pi_{\boldsymbol{p}} - \Pi^k \pi_{\boldsymbol{p}} \tag{5.13}$$

since $\Pi^k P = P$ as Π^k preserves the polynomials in $\mathbb{P}^k(K)$. Using the shape function, Mirebeau showed that there exists a sequence of triangulations with at most N elements such that the following asymptotic estimate holds ([38], Th. 2.1.1):

$$\limsup_{N\to\infty} \left(N^{\frac{m}{2}} \|f - \Pi^k f\|_{L^p(\mathcal{T})} \right) \leq \left\| S_{m,p} \left(\frac{d^{(m)} f}{m!} \right) \right\|_{L^\tau(\Omega)}, \tag{5.14}$$

where the $\limsup_{N\to\infty}$ operator is defined by $\limsup_{N\to\infty} x_N \triangleq \inf_{N\geq 0} \sup_{m\geq N} x_m$. This lim sup is an important feature of the estimate. It essentially means that the estimate guarantees that the error is controlled by the behaviour of the high-order derivatives of f *past a certain number of elements*, but it does not characterize the error on coarse triangulations. In other words, the triangulations should be fine enough to capture the features of the solution for this estimate to apply. In the asymptotic regime, the error on each optimal mesh of this sequence decreases as $N^{-\frac{m}{2}} = N^{-\frac{k+1}{2}}$, indicating convergence with a rate $k+1$, similarly to isotropic convergence rates. The overall decrease is controlled by the shape function, which depends nonlinearly on the high-order derivatives of f. This results generalizes Chen's estimate (5.10) for interpolation of degree k. Indeed, for linear interpolation, denoting by a, b, c the coefficients of H_f, the shape function is given by:

$$S_{2,p}(ax^2 + 2bxy + cy^2) = C\sqrt{|b^2 - ac|} = C\sqrt{|\det H_f|}, \tag{5.15}$$

for some positive constant $C = C(p)$, thus we recover Chen's estimate for $n = 2$:

$$\limsup_{N\to\infty} \left(N \ \|f - \Pi^1 f\|_{L^p(\mathcal{T})} \right) \leq C \left\| \sqrt{|\det H_f|} \right\|_{L^\tau(\Omega)}. \tag{5.16}$$

The proof of (5.14) reveals that the optimal triangulations equidistribute the interpolation error in L^p norm on the triangles, and feature triangles whose shape is governed by $d^{(m)} f$. The sequence of triangulations appearing in the proof of (5.14) is not constructed from a Riemannian metric, but relies on an *ad hoc* technique that is not tractable for practical applications. However, similarly to Hessian-based adaptation, a Riemannian metric can be derived from $d^{(m)} f$ and metric-based adaptation can be performed to create comparable optimal triangulations satisfying the same error bounds, see Section 5.3.

5.1.2 Error bounds for the $W^{1,p}$ seminorm

For the $W^{1,p}$ seminorm, the counterpart of the shape function is:

$$S'_{m,p}(\pi) \triangleq \inf_{|K|=1} |\pi - \Pi^k \pi|_{W^{1,p}(K)}. \tag{5.17}$$

Similarly to (5.14), Mirebeau showed that there exists a sequence of triangulations with at most N elements such that asymptotically ([38], Th. 3.1.1):

$$\limsup_{N\to\infty} \left(N^{\frac{m-1}{2}} |f - \Pi^k f|_{W^{1,p}(\mathcal{T})} \right) \leq \left\| S'_{m,p} \left(\frac{d^{(m)} f}{m!} \right) \right\|_{L^\tau(\Omega)}. \tag{5.18}$$

The error on optimal triangulations decreases as $N^{-\frac{m-1}{2}} = N^{-\frac{k}{2}}$, indicating convergence with a rate k. As for the L^p case, optimal triangulations should equidistribute the $W^{1,p}$ interpolation error on the triangles, and the shape of those triangles should follow $d^{(m)} f$. In addition, and specifically for $W^{1,p}$ error minimization, the largest angle of the triangles should be bounded away from 180 degrees. This last criterion, limiting the *sliverness* of the elements, is difficult to control with off-the-shelf mesh generators. For instance, it cannot be explicitly controlled with `mmg2d`, the meshing library used in this thesis. This does not seem to alter the convergence rate in the test cases presented here, however, save for linear interpolation.

5.2 Anisotropic measure of the higher-order derivatives

Before giving the main results of metric-based error estimators, we formally define the auxiliary field of SPD matrices $\mathcal{Q} \in \mathrm{Sym}_2^+$, called the *anisotropic measure* of f and already introduced in Section 4.5, and we discuss its properties. This matrix links the higher-order derivatives $d^{(m)} f$, which are m-tensor-valued quantities, to the Riemannian metric $\mathcal{M}$, which is intrinsically a 2-tensor field. At each point $\boldsymbol{p} \in \Omega$, $\mathcal{Q}(\boldsymbol{p})$ is defined as a tight upper bound on either the homogeneous polynomial $\pi_{\boldsymbol{p}}(\xi)$ when minimizing the L^p norm of the interpolation error, or on its derivative $\nabla \pi_{\boldsymbol{p}}$ when minimizing the $W^{1,p}$ seminorm. The matrix $\mathcal{Q}$ thus encodes the principal directions of anisotropy of $d^{(m)} f$. Since it controls the derivatives of f, the anisotropic measure also controls the shape functions $S_{m,p}$ and $S'_{m,p}$.

5.2.1 Upper bound for L^p norm minimization

The upper bound is understood in the sense that at each $\boldsymbol{p} \in \Omega$, there exists $\mathcal{Q} = \mathcal{Q}(\boldsymbol{p}) \in \mathrm{Sym}_2^+$ such that:

$$|(\boldsymbol{\xi} \cdot \nabla)^m f(\boldsymbol{p})| = |\pi_{\boldsymbol{p}}(\boldsymbol{\xi})| \leq \left(\boldsymbol{\xi}^T \mathcal{Q} \boldsymbol{\xi}\right)^{\frac{m}{2}}, \quad \forall \boldsymbol{\xi} \in \mathbb{R}^2. \tag{5.19}$$

Although this upper bound must be verified for any vector, it can be simplified as follows thanks to the homogeneity in $\boldsymbol{\xi}$ of both sides of (5.19). For any $\pi \in \mathbb{H}^m$, we define the **level curve 1 of** $|\pi|$ by the set:

$$\Lambda(\pi) \triangleq \left\{\boldsymbol{z} \in \mathbb{R}^2 \mid |\pi(\boldsymbol{z})| = 1\right\}. \tag{5.20}$$

Then:

Proposition 5.1. *The inequality* (5.19) *is equivalent to:*

$$\boldsymbol{z}^T \mathcal{Q} \boldsymbol{z} \geq 1, \quad \forall \boldsymbol{z} \in \Lambda(\pi_{\boldsymbol{p}}). \tag{5.21}$$

Proof. See Appendix A.4.1.

This equivalent inequality needs only be verified for points in $\Lambda(\pi_{\boldsymbol{p}})$. Geometrically, (5.21) imposes that the ellipse $\boldsymbol{z}^T \mathcal{Q} \boldsymbol{z} = 1$ is contained in $\Lambda(\pi_{\boldsymbol{p}})$. The tightest upper bound is thus associated with the ellipse of largest area and included in the level curve 1 of $|\pi_{\boldsymbol{p}}|$. Let $\mathbb{E}$ be the set of ellipses in $\mathbb{R}^2$ centered at zero, and let $E \in \mathbb{E}$ denote the ellipse associated to a matrix $M \in \mathrm{Sym}_2^+$, given by the set:

$$E = \left\{\boldsymbol{z} \in \mathbb{R}^2 \mid \boldsymbol{z}^T M \boldsymbol{z} = 1\right\}. \tag{5.22}$$

The area of E is $|E| = \boldsymbol{\pi}/\sqrt{\det M}$ with $\boldsymbol{\pi} = 3.14159...$ The largest area, when it exists, is thus solution of:

$$\sup_{E \in \mathbb{E},\ E \subset \Lambda(\pi_{\boldsymbol{p}})} |E|/\boldsymbol{\pi} = \sup_{M \in \mathrm{Sym}_2^+,\ E \subset \Lambda(\pi_{\boldsymbol{p}})} \sqrt{\det M}^{-1}, \tag{5.23}$$

which can be infinite. In the following, we consider the equivalent minimization problem:

$$\mathcal{Q}(\boldsymbol{p}) = \arg\inf \left\{\det M \mid M \in \mathrm{Sym}_2^+, \ \ \boldsymbol{z}^T M \boldsymbol{z} \geq 1, \ \ \forall \boldsymbol{z} \in \Lambda(\pi_{\boldsymbol{p}})\right\} \tag{5.24}$$

which determines the tightest upper bound $\mathcal{Q}$ at each $\boldsymbol{p} \in \Omega$. Before studying the solution $\mathcal{Q}$, let us observe that it is related to the largest ellipse describing the anisotropy of the leading term of the interpolation error of f at $\boldsymbol{p}$. On the other hand, the shape function $S_{m,p}$ defines

the shape of the triangles leading to minimal interpolation error. Mirebeau showed that the solution to (5.24) is equivalent to the shape function $S_{m,p}(\pi_{\boldsymbol{p}})$, in the sense that there exist constants $0 < c \leq C$ such that:

$$c\left(\det \mathcal{Q}(\boldsymbol{p})\right)^{\frac{m}{4}} \leq S_{m,p}(\pi_{\boldsymbol{p}}) = \inf_{|K|=1} \|\pi - \Pi^k \pi\|_{L^p(K)} \leq C\ \left(\det \mathcal{Q}(\boldsymbol{p})\right)^{\frac{m}{4}}. \tag{5.25}$$

In other words, the interpolation error for π is controlled by (a power of) the area of the largest ellipse inscribed in $\Lambda(\pi)$.

As pointed out by Cao [77], some choices of $\mathcal{Q}$ are straightforward. For linear interpolation, since Taylor's remainder is controlled by the Hessian matrix H_f, we can set for $\alpha > 0$:

$$\mathcal{Q} = |H_f| + \alpha I, \tag{5.26}$$

where the absolute value acts on the eigenvalues of H_f. Hence, $\mathcal{Q}$ is simply a majoration of the Hessian matrix with a small positive perturbation to avoid a degenerate matrix when H_f is singular. This expression of $\mathcal{Q}$ does not actually solve problem (5.24), but a constrained version where the maximum diameter of E is set to 2α, see Section 5.4.1. If one is only interested in isotropic refinement, it is sufficient to consider the largest circle D_π contained in $\Lambda(\pi)$, that is:

$$\mathcal{Q} = \max_{\|\boldsymbol{\xi}\|=1} |\pi(\boldsymbol{\xi})|\, I = \max_{\|\boldsymbol{\xi}\|=1} |(\boldsymbol{\xi} \cdot \nabla)^m f|\, I, \tag{5.27}$$

which amounts to considering only the largest directional derivative of f. Obtaining an upper bound for the general case is not trivial. In the paper where it was introduced [77], Cao presented the computation of an approximate upper bound $\mathcal{Q}$ in arbitrary dimension and for arbitrary interpolation degree based on numerical optimization. Later, Mirebeau proposed an analytical solution for the linear ($k = 1$) and quadratic ($k = 2$) case in two dimensions. As metrics are to be computed multiple times and at each mesh vertex in the mesh adaptation loop, an analytic solution is particularly desirable. While the linear case of (5.26) is easily handled, the quadratic case deserves more attention and is detailed in Section 5.4.1. For $k > 2$, in two and three dimensions, there is no analytical solution available for $\mathcal{Q}$, thus another approach must be considered. In this work, we implemented the iterative method of Coulaud and Loseille [11, 12] to find an approximate upper bound. It is described in Section 5.4.2.

5.2.2 Upper bound for $W^{1,p}$ norm minimization

Instead of the minimization problem (5.24), Mirebeau introduced the problem:

$$\mathcal{Q}'(\boldsymbol{p}) = \arg\inf \left\{ \det M \mid M \in \mathrm{Sym}_2^+, \quad (\boldsymbol{z}^T M \boldsymbol{z})^k \geq \|\nabla \pi_{\boldsymbol{p}}\|^2, \quad \forall \boldsymbol{z} \in \mathbb{R}^2 \right\}, \tag{5.28}$$

equivalent by homogeneity to:

$$\mathcal{Q}'(\boldsymbol{p}) = \arg\inf \left\{ \det M \mid M \in \mathrm{Sym}_2^+, \quad \boldsymbol{z}^T M \boldsymbol{z} \geq 1, \quad \forall \boldsymbol{z} \in \Lambda(\nabla \pi_{\boldsymbol{p}}) \right\}, \tag{5.29}$$

that is, seeking the largest ellipse included in the level set 1 of $\|\nabla \pi_{\boldsymbol{p}}\| = \sqrt{(\partial_x \pi_{\boldsymbol{p}})^2 + (\partial_y \pi_{\boldsymbol{p}})^2}$. Similarly to the case of the L^p norm, one shows that the solution of (5.29) is equivalent to the shape function $S'_{m,p}$. Analytic minimizers of (5.29) were derived for linear and quadratic elements. For linear elements, let $\pi = ax^2 + 2bxy + cy^2 \in \mathbb{H}^2$ be the error polynomial, with $ac - b^2 \neq 0$. Then the matrix:

$$\mathcal{Q}' = 4 \begin{pmatrix} a^2 + b^2 & ab + bc \\ ab + bc & b^2 + c^2 \end{pmatrix} \tag{5.30}$$

satisfies the equality $\|\nabla \pi\|^2 = \boldsymbol{z}^T \mathcal{Q} \boldsymbol{z}$ for all $\boldsymbol{z} \in \mathbb{R}^2$ and the associated ellipse is the largest included in $\Lambda(\nabla \pi)$. It is the unique minimizer of (5.29). For quadratic elements, let $\pi = ax^3 + 3bx^2y + 3cxy^2 + dy^3 \in \mathbb{H}^3$. Then the matrix:

$$\mathcal{Q}' = 3 \sqrt{\begin{pmatrix} a^2 + 2b^2 + c^2 & ab + 2bc + cd \\ ab + 2bc + cd & b^2 + 2c^2 + d^2 \end{pmatrix}} \tag{5.31}$$

is a *near*-minimizer of (5.28) in the sense that it satisfies $\|\nabla \pi\|^2 \leq (\boldsymbol{z}^T \sqrt{2} \mathcal{Q}' \boldsymbol{z})^2$, that is, it satisfies the constraint up to the constant $\sqrt{2}$, and its determinant is equivalent to the shape function $S'_{3,p}(\pi)$, in the sense that there exist positive constants c, C such that:

$$\frac{c}{C} \sqrt{\det \mathcal{Q}'} \leq S'_{3,p}(\pi) \leq C \sqrt{\det \mathcal{Q}'}. \tag{5.32}$$

That is, the interpolation error for $\nabla \pi$ is controlled by the area of the largest ellipse inscribed in $\Lambda(\nabla \pi)$, similarly to the case of the L^p norm. For the general $W^{s,p}$ seminorm, Cao introduced instead the problem [77]:

$$\widetilde{\mathcal{Q}}(\boldsymbol{p}) = \arg\min \left\{ \lambda_{\max}(M)^{\frac{s}{2}} (\det M)^{\frac{m-s}{4}} \mid M \in \mathrm{Sym}_2^+, \quad \boldsymbol{z}^T M \boldsymbol{z}, \quad \forall \boldsymbol{z} \in \Lambda(\pi_{\boldsymbol{p}}) \right\}, \tag{5.33}$$

involving the largest eigenvalue and where the constraints are written for the level set 1 of $\pi_{\boldsymbol{p}}$ instead of $\nabla\pi_{\boldsymbol{p}}$. It coincides with (5.24) for the L^p norm, but differs from (5.29) for the $W^{1,p}$ seminorm, hence so do their solutions. In the following, the optimal Riemannian metric for the $W^{1,p}$ seminorm is derived from Mirebeau's solution to problem (5.29) instead of (5.33).

5.3 Continuous error estimates in $W^{s,p}$ norm and optimal metrics

We now consider error estimates written in terms of a Riemannian metric. Just as discrete estimates describe the error for arbitrary triangulations, the estimates hereafter apply to arbitrary metrics, which are not necessarily adapted to the target function f. The optimal metrics, however, follow the anisotropy of f.

A very general estimate, valid for all $W^{s,p}$ seminorms, is due to Cao [78]. To write their estimate, we define the nonsymmetric matrix $\mathcal{N} \triangleq P\Lambda^{-\frac{1}{2}}$ from the diagonalized form $[\mathcal{M}] = P\Lambda P^T$, so that $[\mathcal{M}]^{-1} = \mathcal{N}\mathcal{N}^T$, but $\mathcal{N} \neq \mathcal{M}^{-\frac{1}{2}} = P\Lambda^{-\frac{1}{2}}P^T$. Then, for interpolation of degree k, Cao showed that for any metric $\mathcal{M}$ and for a triangulation with N elements that is quasi-unit for $\mathcal{M}$, the following estimate holds ([77], Th. 2.1):

$$|f - \Pi^k f|_{W^{s,p}(\mathcal{T})} \leq CN^{-\frac{m-s}{n}} \left(\int_\Omega \sqrt{|\det \mathcal{M}|}\, d\boldsymbol{x}\right)^{\frac{m-s}{n}} \left(\int_\Omega \|\mathcal{N}^{-1}\|^{sp} \|\mathcal{N}^T \widetilde{\mathcal{Q}} \mathcal{N}\|^{\frac{mp}{2}}\, d\boldsymbol{x}\right)^{\frac{1}{p}}, \quad (5.34)$$

where $\widetilde{\mathcal{Q}} \in \mathrm{Sym}_2^+$ is solution of (5.33) and controls the overall decrease of the interpolation error. From this general estimate, they showed that from among all Riemannian metrics, the minimizing metric field is the scaled anisotropic measure defined by:

$$\mathcal{M}^{\mathrm{opt}}_{m,s,p} = c_{\mathcal{M}}\, \lambda_{\max}(\widetilde{\mathcal{Q}})^{\frac{sp}{(m-s)p+n}} (\det \widetilde{\mathcal{Q}})^{\frac{-1}{(m-s)p+n}}\, \widetilde{\mathcal{Q}}, \quad (5.35)$$

with $c_{\mathcal{M}}$ given hereafter. For this optimal metric, the error estimate above evaluates to:

$$|f - \Pi^k f|_{W^{s,p}(\mathcal{T})} \leq CN^{-\frac{m-s}{n}} \left\| \lambda_{\max}(\widetilde{\mathcal{Q}})^{\frac{m}{2}} (\det \widetilde{\mathcal{Q}})^{\frac{m-s}{2n}} \right\|_{L^\tau(\Omega)}. \quad (5.36)$$

In particular for the L^p norm ($s = 0$), we have $\widetilde{\mathcal{Q}} = \mathcal{Q}$, yielding the following optimal Riemannian metric:

$$\mathcal{M}^{\mathrm{opt}}_{m,0,p} = c_{\mathcal{M}}\, (\det \mathcal{Q})^{\frac{-1}{mp+n}}\, \mathcal{Q}, \quad (5.37)$$

associated to the optimal error bound:

$$\|f - \Pi^k f\|_{L^p(\mathcal{T})} \leq CN^{-\frac{m}{n}} \left\| (\det \mathcal{Q})^{\frac{m}{2n}} \right\|_{L^\tau(\Omega)} \quad (5.38)$$

Since $(\det \mathcal{Q})^{m/2n} = (\det \mathcal{Q})^{m/4}$ is equivalent to the shape function $S_{m,p}$ in 2 dimensions, this optimal estimate is equivalent to Mirebeau's estimate for optimal triangulations (5.14). This is the intersection between the discrete and continuous approaches: indeed, (5.14) describes the error on optimal triangulations not necessarily obtained from a Riemannian metric, whereas (5.34) describes the error for Riemannian metrics, not necessarily optimal. When the metric is in fact optimal, then the optimal metric-based error estimate (5.38) matches the optimal mesh-based estimate (5.14). For the $W^{1,p}$ seminorm, we can derive from Mirebeau ([38], Th. 3.2.6) a similar estimate for sufficiently refined triangulations whose average *sliverness* remains controlled:

$$|f - \Pi^k f|_{W^{1,p}(\mathcal{T})} \leq C N^{-\frac{m-1}{n}} \left\| (\det \mathcal{Q}')^{\frac{m-1}{2n}} \right\|_{L^\tau(\Omega)}. \tag{5.39}$$

The optimal discrete and continuous estimates are summarized for the L^p norm and the $W^{1,p}$ seminorm in Table 5.1, where $\mathcal{Q}$ stands for the matrix associated to the largest ellipse inscribed in $\Lambda(\pi)$, and $\mathcal{Q}'$ is associated to $\Lambda(\nabla\pi)$.

In both expressions of $\mathcal{M}^{\text{opt}}$, the constant $c_{\mathcal{M}}$ controls the mesh density, hence the number of vertices or elements in the final mesh. It can be set by imposing either the target error in $W^{m,p}$ norm on each triangle [1, 38], or, alternatively, setting the metric complexity $\mathcal{C}$ to a target number of vertices N_v. For the latter and the L^p norm, we have:

$$\mathcal{C} = \int_\Omega \sqrt{\det \mathcal{M}^{\text{opt}}_{m,0,p}(x)}\, d\boldsymbol{x} = \int_\Omega c_{\mathcal{M}}^{\frac{n}{2}} (\det \mathcal{Q})^{\frac{-n}{2(mp+n)}} \sqrt{\det \mathcal{Q}}\, d\boldsymbol{x} = N_v, \tag{5.40}$$

hence:

$$c_{\mathcal{M}} = N_v^{\frac{2}{n}} \left(\int_\Omega (\det \mathcal{Q})^{\frac{mp}{2(mp+n)}}\, d\boldsymbol{x} \right)^{-\frac{2}{n}}. \tag{5.41}$$

Alauzet and Loseille, on the other hand, considered the direct minimization over all Riemannian metrics of error functionals $\mathcal{E}(\mathcal{M})$ in L^p norm, introduced in Chapter 4, under an equality constraint on the number of vertices [1, 8, 55]:

$$\min_{\mathcal{C}(\mathcal{M}) \leq N_v} \mathcal{E}(\mathcal{M}). \tag{5.42}$$

This is a continuous counterpart of the problem (5.9) over all triangulations with a prescribed number of elements. For instance, for linear interpolation, the optimal metric satisfies:

$$\mathcal{M}^{\text{opt}}_{2,0,p} = \arg \min_{\mathcal{C}(\mathcal{M}) \leq N_v} \left(\int_\Omega \operatorname{tr} \left(\mathcal{M}_{\boldsymbol{x}}^{-1/2} |H_f(\boldsymbol{x})| \mathcal{M}_{\boldsymbol{x}}^{-1/2} \right)^p d\boldsymbol{x} \right)^{\frac{1}{p}}, \tag{5.43}$$

Discrete	There exists a sequence of optimal triangulations such that:	
	L^p	$\limsup_{N\to\infty}\left(N^{\frac{m}{n}}\|f-\Pi^k f\|_{L^p(\mathcal{T})}\right) \le \left\|S_{m,p}\left(\frac{d^{(m)}f}{m!}\right)\right\|_{L^\tau(\Omega)}$ (5.14)
	$W^{1,p}$	$\limsup_{N\to\infty}\left(N^{\frac{m-1}{n}}\|f-\Pi^k f\|_{W^{1,p}(\mathcal{T})}\right) \le \left\|S'_{m,p}\left(\frac{d^{(m)}f}{m!}\right)\right\|_{L^\tau(\Omega)}$ (5.18)
Continuous	Unit triangulations with respect to $\mathcal{M}^{\text{opt}}(\mathcal{Q})$ satisfy:	
	L^p	$\|f-\Pi^k f\|_{L^p(\mathcal{T})} \le CN^{-\frac{m}{n}}\left\|(\det\mathcal{Q})^{\frac{m}{2n}}\right\|_{L^\tau(\Omega)}$ (5.38)
	$W^{1,p}$	$\|f-\Pi^k f\|_{W^{1,p}(\mathcal{T})} \le CN^{-\frac{m-1}{n}}\left\|(\det\mathcal{Q}')^{\frac{m-1}{2n}}\right\|_{L^\tau(\Omega)}$ (5.39)

Table 5.1 Optimal discrete and continuous error estimates.

and for higher-order interpolation:

$$\mathcal{M}^{\text{opt}}_{m,0,p} = \arg\min_{\mathcal{C}(\mathcal{M})\le N_v}\left(\int_\Omega \operatorname{tr}\left(\mathcal{M}_{\boldsymbol{x}}^{-1/2}\mathcal{Q}(\boldsymbol{x})\mathcal{M}_{\boldsymbol{x}}^{-1/2}\right)^{\frac{pm}{2}}\,d\boldsymbol{x}\right)^{\frac{1}{p}}, \tag{5.44}$$

Both problems (5.43) and (5.44) are tackled by solving the Euler-Lagrange equations associated to $\mathcal{E}(\mathcal{M})$, and the minimizer for the L^p norm writes:

$$\mathcal{M}^{\text{opt}}_{m,0,p} = N_v^{\frac{2}{n}}\left(\int_\Omega (\det\mathcal{Q})^{\frac{mp}{2(mp+n)}}\,d\boldsymbol{x}\right)^{-\frac{2}{n}}(\det\mathcal{Q})^{\frac{-1}{mp+n}}\,\mathcal{Q}, \tag{5.45}$$

which is exactly the optimal metric (5.37) of Cao. Since the error model is known, substituting the optimal metric in $\mathcal{E}(\mathcal{M})$ yields the following exact error estimate:

$$\begin{aligned}\mathcal{E}(\mathcal{M}^{\text{opt}}_{m,0,p}) = \|f-\pi^k_{\mathcal{M}}f\|_{L^p(\Omega)} &= n^{\frac{m}{mp+2n}}N_v^{-\frac{m}{n}}\left(\int_\Omega(\det\mathcal{Q})^{\frac{mp}{2(mp+n)}}\,d\boldsymbol{x}\right)^{\frac{mp+n}{pn}}\\ &= n^{\frac{m}{mp+2n}}N_v^{-\frac{m}{n}}\left\|(\det\mathcal{Q})^{\frac{m}{2n}}\right\|_{L^\tau(\Omega)}.\end{aligned} \tag{5.46}$$

This is Cao's estimate, (5.38), with an equality and a known constant C. The operator $\pi^k_{\mathcal{M}}f$ is a continuous counterpart of the interpolation operator $\Pi^k f$ and is used extensively in the

continuous mesh framework, see [8] for more details.

Choice of optimal metrics

Although Cao's optimal metric (5.35) is valid for any seminorm and any degree of polynomial interpolation, it requires solving (5.33). Since analytic and numerical solutions exist to compute $\mathcal{Q}$ by solving (5.24) or (5.29), we consider those minimization problems instead. An optimal metric of the same form as (5.37) was proposed by Mirebeau [38] for the $L^p = W^{0,p}$ norm and $W^{1,p}$ seminorm:

$$\mathcal{M}^{\text{opt}}_{m,s,p} = c_{\mathcal{M}} \left(\det \mathcal{Q}\right)^{-\frac{\tau}{2p}} \mathcal{Q}, \tag{5.47}$$

with τ given by (5.8) and $\mathcal{Q}$ solution of (5.24) and (5.29). explicitly, these **optimal metrics** for a target complexity $\mathcal{C} = N_v$ in dimension n write for $s = 0, 1$ and arbitrary p and $m = k + 1$:

$$\mathcal{M}^{\text{opt}}_{m,s,p} = N_v^{\frac{2}{n}} \left(\int_\Omega (\det \mathcal{Q})^{\frac{p(m-s)}{2(p(m-s)+n)}} \, d\boldsymbol{x} \right)^{-\frac{2}{n}} (\det \mathcal{Q})^{\frac{-1}{(m-s)p+n}} \, \mathcal{Q}. \tag{5.48}$$

These are the Riemannian metrics used in this thesis to perform mesh adaptation with linear triangles driven by interpolation error. In the next section, we detail the computation of $\mathcal{Q}$, then we illustrate on some examples.

5.4 Computation of the anisotropic measure

5.4.1 Analytic solution for linear and quadratic interpolation

We detail Mirebeau's analytic solution to determine the anisotropic measure $\mathcal{Q}$ when minimizing the L^p norm of the interpolation error, the case of the $W^{1,p}$ seminorm being already covered in Section 5.2.2. The solution can be found in Mirebeau's thesis [38] for the most part, save for the explicit computation of some parameters, which are detailed here in appendix.

We consider the minimization problem (5.24), which writes:

$$\mathcal{Q}(\boldsymbol{p}) = \arg\inf \left\{ \det M \mid M \in \text{Sym}_2^+, \quad \boldsymbol{z}^T M \boldsymbol{z} \geq 1, \quad \forall z \in \Lambda(\pi_{\boldsymbol{p}}) \right\}$$

for linear and quadratic interpolation, that is, $m = 2, 3$. The linear case can be treated straightaway: if H_f with coefficients a, b, c and $ac - b^2 \neq 0$ is the Hessian of the target function f, then $\pi(x, y) = ax^2 + 2bxy + cy^2 \in \mathbb{H}^2$ represents the local error and $\mathcal{Q}(\boldsymbol{p})$ is identically $|H_f(\boldsymbol{p})|$. If the Hessian is singular, then $\mathcal{Q}$ is given by (5.64) and is the solution of a well-posed constrained problem defined hereafter. Thus, we focus in the following on quadratic interpolation which is less straightforward. For quadratic interpolation, the homogeneous

error polynomial writes $\pi(x,y) = ax^3 + bx^2y + cxy^2 + dy^3 \in \mathbb{H}^3$. The coefficients are identified to the derivatives of f at $\boldsymbol{p}$:

$$\pi_{\boldsymbol{p}}(x,y) = \underbrace{\partial_{xxx} f(\boldsymbol{p})}_{a} x^3 + \underbrace{3\,\partial_{xxy} f(\boldsymbol{p})}_{b} x^2y + \underbrace{3\,\partial_{xyy} f(\boldsymbol{p})}_{c} xy^2 + \underbrace{\partial_{yyy} f(\boldsymbol{p})}_{d} y^3. \tag{5.49}$$

Geometrically, we seek the ellipse $E_\pi \in \mathbb{E}$ centered at zero and of maximal area contained in $\Lambda(\pi)$, the level curve 1 of $|\pi|$. We say that E_π is the ellipse associated to $\mathcal{Q}_\pi$, and conversely $\mathcal{Q}_\pi$ is the matrix of E_π. Such an ellipse has an area close to the supremum (5.23). In the following, we refer to (5.24) recalled above as the *unconstrained problem* and we denote by $\mathcal{Q}_\pi$ its solution. A problem with constrained diameter for E_π is introduced further, hence this distinction. To describe E_π, the main idea is to express any $\pi \in \mathbb{H}^3$ as one of four reference cubic homogeneous polynomials, noted $\tilde{\pi}_i$, through a linear mapping ϕ such that $\pi \circ \phi = \tilde{\pi}_i$. The families of ellipses tangent to each set $\Lambda(\tilde{\pi}_i)$ can be found analytically, then mapped by ϕ to obtain the ellipses tangent to $\Lambda(\pi)$, among which we seek the ellipse of largest area. The characterization of π thus boils down to identifying the linear map ϕ. The four reference polynomials are the set:

$$\{\ \tilde{\pi}_1 = x^3,\quad \tilde{\pi}_2 = x^2y,\quad \tilde{\pi}_3 = x(x^2+3y^2),\quad \tilde{\pi}_4 = x(x^2-3y^2)\ \}. \tag{5.50}$$

Notably, y^3 is not a reference polynomial, because when $a = 0$, π is divisible by y and there exists a rotation U such that $\pi \circ U$ is not divisible by y, so that $\pi \circ U$ can be mapped to one of the four $\tilde{\pi}_i$.

To classify the homogeneous polynomials in $\mathbb{H}^3$ and describe the map ϕ, we define the following two useful quantities. The **Hessian** of a homogeneous polynomial $\pi \in \mathbb{H}^3$ is the homogeneous polynomial hess $\pi \in \mathbb{H}^2$ defined by:

$$\begin{aligned} \text{hess } \pi(x,y) &\triangleq -\frac{1}{4}\left(\partial_{xx}\pi\,\partial_{yy}\pi - (\partial_{xy}\pi)^2\right) \\ &= (b^2-3ac)x^2 + (bc-9ad)xy + (c^2-3bd)y^2. \end{aligned} \tag{5.51}$$

It is, up to the constant $-1/4$, the determinant of the Hessian matrix of $\pi(x,y)$. The **discriminant** of $\pi \in \mathbb{H}^3$ is the real number disc π defined by:

$$\begin{aligned} \text{disc } \pi &\triangleq -\frac{1}{3}\left[(bc-9ad)^2 - 4(b^2-3ac)(c^2-3bd)\right] \\ &= b^2c^2 - 4ac^3 - 4b^3d + 18abcd - 27a^2d^2. \end{aligned} \tag{5.52}$$

The discriminant can also be written from the factored form $\pi = \lambda(x-r_1y)(x-r_2y)(x-r_3y)$:

$$\text{disc } \pi = \lambda^4(r_1 - r_2)^2(r_2 - r_3)^2(r_3 - r_1)^2. \tag{5.53}$$

Clearly, $\lambda = a$ and the discriminant is zero when there are multiple roots in the factored form of π. The following proposition describes this factorization and gives the linear transformation ϕ explicitly, either directly or through its inverse. Note that although the map ϕ is linear, it is not a rotation in general.

Proposition 5.2. *Let $\pi \in \mathbb{H}^3$.*

1. *If hess $\pi = 0$, then disc $\pi = 0$ and π can be factored as $\pi = \lambda(x - ry)^3$ with r a triple real root. The inverse mapping:*

$$\phi_\pi^{-1} \triangleq \begin{pmatrix} \sqrt[3]{\lambda} & -\sqrt[3]{\lambda}r \\ 0 & 1 \end{pmatrix} \tag{5.54}$$

   is such that $\pi \circ \phi_\pi = \tilde{\pi}_1 = x^3$.

2. *If hess $\pi \neq 0$ and disc $\pi = 0$, then π can be factored as $\pi = \lambda(x - r_1y)^2(x - r_2y)$ with a double real root r_1 and a single real root r_2. The inverse mapping:*

$$\phi_\pi^{-1} \triangleq \begin{pmatrix} 1 & -r_1 \\ \sqrt[3]{\lambda} & -\sqrt[3]{\lambda}r_2 \end{pmatrix} \tag{5.55}$$

   is such that $\pi \circ \phi_\pi = \tilde{\pi}_2 = x^2y$.

3. *If hess $\pi \neq 0$ and disc $\pi < 0$, then π can be factored as $\pi = \lambda(x - r_1y)(x - r_2y)(x - r_3y)$ with a real root r_1 and two complex conjugate roots r_2, r_3 with $\text{Im}(r_2) > 0$. The mapping:*

$$\phi_\pi \triangleq \frac{\lambda}{\sqrt[3]{2disc\ \pi}} \begin{pmatrix} r_1(r_2 + r_3) - 2r_2r_3 & i(r_2 - r_3)r_2\sqrt{3} \\ 2r_1 - (r_2 + r_3) & i(r_2 - r_3)\sqrt{3} \end{pmatrix} \tag{5.56}$$

   is such that $\pi \circ \phi_\pi = \tilde{\pi}_3 = x(x^2+3y^2)$. All entries of ϕ_π are real since $r_2 - r_3 = 2i\,\text{Im}(r_2)$.

4. *If hess $\pi \neq 0$ and disc $\pi > 0$, then π can be factored as $\pi = \lambda(x-r_1y)(x-r_2y)(x-r_3y)$ with three real roots $r_1 < r_2 < r_3$. The mapping:*

$$\phi_\pi \triangleq \frac{\lambda}{\sqrt[3]{2disc\ \pi}} \begin{pmatrix} r_1(r_2 + r_3) - 2r_2r_3 & (r_2 - r_3)r_2\sqrt{3} \\ 2r_1 - (r_2 + r_3) & (r_2 - r_3)\sqrt{3} \end{pmatrix} \tag{5.57}$$

   is such that $\pi \circ \phi_\pi = \tilde{\pi}_4 = x(x^2 - 3y^2)$.

5. *If $a = 0$, then π is divisible by y, but there exists a rotation U such that $\pi \circ U$ is not divisible by y. Since $\det U = 1$ and the discriminant satisfies the invariance property disc $(\pi \circ \phi) = (\det \phi)^6$ disc π, we have disc $(\pi \circ U) =$ disc π and the mapping is $\phi_\pi = U \circ \phi_{\pi \circ U}$, with $\phi_{\pi \circ U}$ obtained from one of the mappings above.*

Proof. The cases 3, 4 and 5 are described by Lemma 2.4.2 in Mirebeau's thesis [38]. The case 1 and 2 are the cases where disc $\pi = 0$, and can be verified by direct computation. □

With ϕ known, we now describe the ellipses tangent to the level curve of each of the four reference polynomials. The level curve 1 of $|\pi|$ has a branch for each real root in its factored form, see Fig. 5.1. Following Mirebeau's terminology, an ellipse E is said to be **quadri-tangent** to a level curve $\Lambda(\pi)$ if it is tangent to $\Lambda(\pi)$ in at least four points. This property is invariant under a linear transformation: if E is quadri-tangent to $\Lambda(\pi \circ \phi)$, then $\phi^{-1}(E)$ is quadri-tangent to $\Lambda(\pi)$. Similarly, if H_λ is the matrix associated to E, then $\phi^{-T} H_\lambda \phi^{-1}$ is the matrix associated to $\phi^{-1}(E)$. As any $\pi \in \mathbb{H}^3$ can be mapped to one of the four reference polynomials, it is then sufficient to characterize the ellipses quadri-tangent to $\Lambda(\tilde{\pi}_i)$. As we discuss next and judging from the branches of the reference polynomials in Fig. 5.1, these quadri-tangent ellipses are well-defined only for $\tilde{\pi}_3$ and $\tilde{\pi}_4$, that is, for polynomial with nonzero discriminant. Their solution is summarized in Proposition 5.3. For polynomials with zero discriminant, we consider ellipses of constrained diameter, and the solution is summarized in the more general Proposition 5.4.

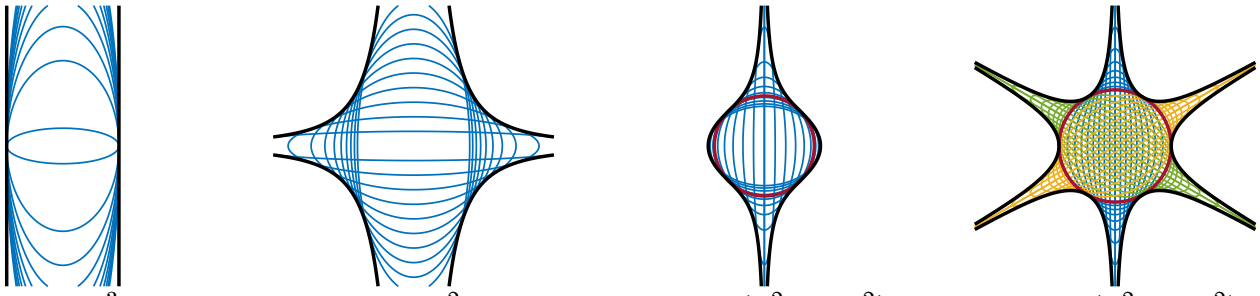


Figure 5.1 Level curve 1 of the reference polynomials $\tilde{\pi}_i$ (in black) and examples of bi- (for $\tilde{\pi}_1$ only) or quadri-tangent ellipses (in blue). $\tilde{\pi}_4$ has three families of quadri-tangent ellipses, which are rotation of 0, 60 or 120 degrees of one another. For $\tilde{\pi}_3$ and $\tilde{\pi}_4$, the ellipse of maximal area is the circle of radius $\sqrt[3]{2}$ and 1 respectively (in red).

There are no quadri-tangent ellipse for the polynomial $\tilde{\pi}_1 = x^3$, as the level curve consists of two parallel lines, Fig. 5.1. The ellipses included in $\Lambda(\tilde{\pi}_1)$ are bi-tangent to the level curve and there is no ellipse of maximal area. A solution for $\mathcal{Q}$ is obtained hereafter by introducing a

constraint on the diameter of E. In the three other cases, we consider the system of equations obtained by imposing simultaneously that the points of tangency belong to E and $\Lambda(\tilde{\pi}_i)$. The polynomials $\tilde{\pi}_2 = x^2y$ and $\tilde{\pi}_3 = x(x^2 + 3y^2)$ have their principal directions aligned with the x and y axes, see Fig. 5.1, and so does one of the branches of $\tilde{\pi}_4 = x(x^2 - 3y^2)$. Based on this observation, we can restrict the search to the ellipses aligned with the axes, of the form $\lambda_1 x^2 + \lambda_2 y^2 = 1$ with $\lambda_1, \lambda_2 > 0$. This completely describes the ellipses quadri-tangent to $\tilde{\pi}_2$ and $\tilde{\pi}_3$; the ellipses tangent to the other two branches of $\tilde{\pi}_4$ are obtained by a rotation of 60 and 120 degrees. We solve the following system for λ_1 and λ_2 by eliminating the unknown tangency points (x, y):

$$\begin{cases} |\tilde{\pi}_i(x,y)| & = 1, \\ \lambda_1 x^2 + \lambda_2 y^2 & = 1, \\ \lambda_2 y \partial_x \tilde{\pi}_i(x,y) - \lambda_1 x \partial_y \tilde{\pi}_i(x,y) & = 0. \end{cases} \tag{5.58}$$

The first and second equations impose that the tangency points belong to the level curve of $\tilde{\pi}_i$ and of E, respectively, whereas the third equation is the alignment of the tangent vectors to both curves at the tangency points. For $\tilde{\pi}_2 = x^2y$, the solution is $\lambda_2 = 4/27\lambda_1^2$, thus the ellipses quadri-tangent to x^2y are defined for $\lambda > 0$ by the matrices $H_\lambda = \mathrm{diag}(\lambda, 4/27\lambda^2)$. The determinant goes to zero as $\lambda \to \infty$, so there is no ellipse of maximal area. For $\tilde{\pi}_3 = x(x^2 + 3y^2)$, we have $\lambda_1 = (4 + \lambda_2^3)/3\lambda_2^2$ and the ellipses are described by $H_\lambda = \mathrm{diag}((4 + \lambda_2^3)/3\lambda_2^2, \lambda)$ for $0 < \lambda \leq 2$. For $\lambda > 2$, the ellipses are not tangent to $\Lambda(\tilde{\pi}_3)$ anymore. The determinant reaches its minimum for $\lambda_1 = \lambda_2 = \sqrt[3]{2}$, hence the largest ellipse is the circle of radius $\sqrt[3]{2}$. Finally, for $\tilde{\pi}_4 = x(x^2 - 3y^2)$, we find $\lambda_1 = (4 - \lambda_2^3)/(3\lambda_2^2)$ for $0 < \lambda \leq 1$. The ellipses quadri-tangent to $\Lambda(\tilde{\pi}_4)$ are described by $H_\lambda = V^T \mathrm{diag}((4 - \lambda_2^3)/3\lambda_2^2, \lambda)V$, where V is a rotation of 0, 60 or 120 degrees, see Fig. 5.1. The determinant is minimal for $\lambda_1 = \lambda_2 = 1$, thus the largest ellipse is the unit circle.

From there, one recovers the largest ellipse for an arbitrary π of nonzero discriminant:

Proposition 5.3. *(Solution of the unconstrained problem, Proposition 2.4.4 in [38].)*

- *Let $\pi \in \mathbb{H}^3$ with disc $\pi < 0$. The ellipse of maximal area inscribed in $\Lambda(\pi)$ is the ellipse associated to the matrix:*

$$\mathcal{Q}_\pi = \sqrt[3]{2}\, \phi_\pi^{-T} \phi_\pi^{-1}, \tag{5.59}$$

  with ϕ_π defined by (5.56)*, and* $\det \mathcal{Q}_\pi = \frac{1}{3}\sqrt[3]{|\operatorname{disc} \pi|}$.

- *Let $\pi \in \mathbb{H}^3$ with disc $\pi > 0$. The ellipse of maximal area inscribed in $\Lambda(\pi)$ is the ellipse associated to the matrix:*

$$\mathcal{Q}_\pi = \phi_\pi^{-T} \phi_\pi^{-1}, \tag{5.60}$$

  with ϕ_π defined by (5.57)*, and* $\det \mathcal{Q}_\pi = \frac{2^{-2/3}}{3}\sqrt[3]{\operatorname{disc} \pi}$.

Proof. We apply the mapping $\phi_\pi^{-T} H_\lambda \phi_\pi^{-1}$ to the diagonal matrices associated to the optimal ellipse for $\tilde{\pi}_3$ and $\tilde{\pi}_4$. □

In the case disc $\pi > 0$, simplifications occur and the optimal metric $\mathcal{Q}_\pi$ can be written directly from the coefficients of π, with $B = b/3$ and $C = c/3$:

$$\mathcal{Q}_\pi = \frac{3}{\sqrt[3]{2\text{disc } \pi}} \begin{pmatrix} 2(B^2 - aC) & BC - ad \\ BC - ad & 2(C^2 - Bd) \end{pmatrix}. \tag{5.61}$$

The cases where disc $\pi = 0$ require further attention. For $\tilde{\pi}_1 = x^3$, maximizing the ellipse area yields the degenerate matrix $H_\lambda = \text{diag}(1, \varepsilon)$ with $\varepsilon \to 0$, hence the vertical axis is unbounded. This also occurs when adapting with linear elements when the quadratic polynomial takes the form $\pi = x^2$. In practice, this is not a problem since one can consider a small parameter $\alpha > 0$ and give the mesh generator the modified matrix $\mathcal{Q}_{\pi,\alpha} = \phi_\pi^{-T} H_\lambda \phi_\pi^{-1} + \alpha I$. The modified matrix is bounded and does not degenerate. Here, α is a small real number and not a multi-index. For $\tilde{\pi}_2 = x^2 y$ however, the minimizers are of the form $H_\lambda = \text{diag}(\varepsilon^{-1}, \varepsilon^2)$, with $\det H_\lambda = \varepsilon$ and $\varepsilon \to 0$. The associated ellipses have unbounded vertical axis and their horizontal axis shrinks to zero. Note that the ellipses with unbounded horizontal axis are not minimizers, as $\det H_\lambda = \varepsilon^{-1} \to \infty$ in that case since the other axis shrinks as ε^{-2}. For $\tilde{\pi}_2$, the matrix $\mathcal{Q}_\pi$ itself is thus unbounded and so is $\mathcal{Q}_{\pi,\alpha}$. This naturally leads to consider the optimization problem for ellipses of constrained diameter:

$$\sup_{E \in \mathbb{E},\ E \subset \Lambda(\pi),\ \text{diam}(E) \leq 2\alpha^{-1/2}} |E|, \tag{5.62}$$

where the **diameter** of E is twice its largest semi-axis. This is equivalent to the constrained minimization problem for $\alpha > 0$:

$$\mathcal{Q}_{\pi,\alpha} = \arg\inf \left\{ \det M \mid M \in \text{Sym}_2^+ \text{ such that } \boldsymbol{z}^T M \boldsymbol{z} \geq 1,\ \forall z \in \mathbb{R}^2 \text{ and } M \geq \alpha I \right\}. \tag{5.63}$$

Let $E_{\pi,\alpha}$ be the ellipse associated to $\mathcal{Q}_{\pi,\alpha}$ and solution of (5.62). For linear interpolation, $\mathcal{Q}_{\pi,\alpha}$ is simply given by:

$$\mathcal{Q}_{\pi,\alpha} = U^T \begin{pmatrix} \max(|\lambda_1|, \alpha) & 0 \\ 0 & \max(|\lambda_2|, \alpha) \end{pmatrix} U, \tag{5.64}$$

with λ_i the eigenvalues of the Hessian matrix H_f at $\boldsymbol{p}$ and U the matrix whose columns are the eigenvectors of H_f, similarly to (5.26). For quadratic elements, the largest ellipse

transitions (i) from the largest disc D_π included in $\Lambda(\pi)$ to (ii) the largest ellipse included in $\Lambda(\pi)$ and containing D_π, then to (iii) the unconstrained ellipse E_π, when these ellipses exist. To differentiate between these regimes, Mirebeau defines three numbers $0 \leq \beta_\pi \leq \alpha_\pi \leq \mu_\pi$ and a rotation matrix U_π defined as follows:

- μ_π is the eigenvalue associated to the disc of largest radius inscribed in $\Lambda(\pi)$, that is:

$$2\mu_\pi^{-1/2} \triangleq \max\left\{\mathrm{diam}(D) \mid D \text{ is a disc, } D_\pi \subset \Lambda(\pi)\right\}, \tag{5.65}$$

  or equivalently:

$$\mu_\pi^{-1/2} \triangleq \min\{\|\boldsymbol{z}\| \mid |\pi(\boldsymbol{z})| = 1\}. \tag{5.66}$$

  If $\boldsymbol{z}_\pi$ is such that $|\pi(\boldsymbol{z}_\pi)| = 1$ and $\|\boldsymbol{z}_\pi\| = \mu_\pi^{-1/2}$, then U_π is defined as the rotation which maps $\boldsymbol{z}_\pi$ to $(\|\boldsymbol{z}_\pi\|, 0)$, that is:

$$U_\pi \triangleq \begin{bmatrix} — & \hat{\boldsymbol{z}}_\pi & — \\ — & \hat{\boldsymbol{z}}_\pi^\perp & — \end{bmatrix}, \tag{5.67}$$

  where $\hat{\boldsymbol{z}}_\pi$ is the unit vector aligned with $\boldsymbol{z}_\pi$ and $\hat{\boldsymbol{z}}_\pi^\perp$ is its orthogonal.

- α_π is defined from the diameter of the largest ellipse included in $\Lambda(\pi)$ and containing D_π:

$$2\alpha_\pi^{-1/2} \triangleq \max\left\{\mathrm{diam}(E) \mid E \in \mathbb{E},\ D_\pi \subset E \subset \Lambda(\pi)\right\}. \tag{5.68}$$

  For $\tilde{\pi}_1$, this ellipse is unbounded and we set $\alpha_\pi = 0$.

- β_π is defined through the diameter of the unconstrained optimal ellipse:

$$2\beta_\pi^{-1/2} \triangleq \mathrm{diam}(E(\mathcal{Q}_\pi)) \qquad \text{hence} \qquad \beta_\pi = \lambda_{\min}(\mathcal{Q}_\pi). \tag{5.69}$$

  For $\tilde{\pi}_1$ and $\tilde{\pi}_2$, $\mathcal{Q}_\pi$ does not exist and we set $\beta_\pi = 0$.

The explicit computation of these coefficients, as well as the real number λ_α below, is given in Appendix A.4.2. The following proposition describes $\mathcal{Q}_{\pi,\alpha}$ for any polynomial $\pi \in \mathbb{H}^3$:

Proposition 5.4. *(Solution of the constrained problem, Proposition 2.4.6 in [38].)*

For $\pi \in \mathbb{H}^3$ and $\alpha > 0$, the largest ellipse $E_{\pi,\alpha}$ with diam$(E) \leq 2\alpha^{-1/2}$ inscribed in the level curve 1 of $|\pi|$ and its associated matrix $\mathcal{Q}_{\pi,\alpha} \in Sym_2^+$ are described as follows:

1. *If $\alpha \geq \mu_\pi$, then $\mathcal{Q}_{\pi,\alpha} = \alpha I$ and $E_{\pi,\alpha}$ is the disc of diameter $2\alpha^{-1/2}$.*

2. *If $\alpha \in [\alpha_\pi, \mu_\pi]$, then $E_{\pi,\alpha}$ is the ellipse of diameter $2\alpha^{-1/2}$ inscribed in $\Lambda(\pi)$ and containing D_π. $E_{\pi,\alpha}$ is tangent to the level curve $\Lambda(\pi)$ at the two points z_π and $-z_\pi$, and its matrix is:*

$$\mathcal{Q}_{\pi,\alpha} = U_\pi^T \begin{pmatrix} \mu_\pi & 0 \\ 0 & \alpha \end{pmatrix} U_\pi. \tag{5.70}$$

3. *If $\alpha \in [\beta_\pi, \alpha_\pi]$, then $E_{\pi,\alpha}$ is quadri-tangent to the level curve $\Lambda(\pi)$ and has diameter $2\alpha^{-1/2}$. There are at most three such ellipses, and $E_{\pi,\alpha}$ is the one of largest area. The form of $\mathcal{Q}_{\pi,\alpha}$ depends on the discriminant of π:*

   - *If disc $\pi = 0$ and $\alpha_\pi > 0$, we have:*

$$\mathcal{Q}_{\pi,\alpha} = \phi_\pi^{-T} \begin{pmatrix} \lambda_\alpha & 0 \\ 0 & \frac{4}{27\lambda_\alpha^2} \end{pmatrix} \phi_\pi^{-1}, \tag{5.71}$$

     where ϕ_π^{-1} is given by (5.55) *and λ_α is obtained by imposing that α is an eigenvalue of $\mathcal{Q}_{\pi,\alpha}$, i.e., by solving* $\det(\mathcal{Q}_{\pi,\alpha} - \alpha I) = 0$.

   - *If disc < 0, we have:*

$$\mathcal{Q}_{\pi,\alpha} = \phi_\pi^{-T} \begin{pmatrix} \frac{4+\lambda_\alpha^3}{3\lambda_\alpha^2} & 0 \\ 0 & \lambda_\alpha \end{pmatrix} \phi_\pi^{-1}, \tag{5.72}$$

     where ϕ_π^{-1} is given by (5.55) *and λ_α is obtained by solving* $\det(\mathcal{Q}_{\pi,\alpha} - \alpha I) = 0$.

   - *If disc > 0, we have:*

$$\mathcal{Q}_{\pi,\alpha} = \phi_\pi^{-T} V^T \begin{pmatrix} \frac{4-\lambda_\alpha^3}{3\lambda_\alpha^2} & 0 \\ 0 & \lambda_\alpha \end{pmatrix} V \phi_\pi^{-1}, \tag{5.73}$$

     where ϕ_π^{-1} is given by (5.55)*, V is the rotation of either 0, 60 or 120 degrees that minimizes* $\det \mathcal{Q}_{\pi,\alpha}$ *and λ_α is obtained by solving* $\det(\mathcal{Q}_{\pi,\alpha} - \alpha I) = 0$.

4. *If $\alpha \leq \beta_\pi$, then $E_{\pi,\alpha}$ and $\mathcal{Q}_{\pi,\alpha}$ are the solutions of the unconstrained problem given by Proposition 5.3.*

Proof. See [38].

Illustrative examples of each regime are given in Fig. 5.2, each corresponding to a transformation of a reference polynomial. A `C++` implementation of this proposition was written for this thesis, based in part from Mirebeau's code. The first column is the level curve 1 of $\tilde{\pi}_1 = x^3$ and of a polynomial π such that $\pi \circ \phi = \tilde{\pi}_1$. The ellipses are only bi-tangent, since the level

curve consists of two parallel lines. For each polynomial, the red dot is $\boldsymbol{z}_\pi$, the point on $\Lambda(\pi)$ the closest to the origin. The largest circle D_π of radius $\|\boldsymbol{z}_\pi\| = \mu_\pi^{-1/2}$ inscribed in $\Lambda(\pi)$ is drawn in red. For $\alpha \geq \mu_\pi$, we have $\operatorname{diam}(E) \leq 2\mu_\pi^{-1/2}$ and the optimal ellipses $E_{\pi,\alpha}$ are the circle strictly contained within $\Lambda(\pi)$ and drawn in orange. The only other regime is when $\mu_\pi > \alpha > 0$, that is, when the diameter of $E_{\pi,\alpha}$ increases to infinity. The optimal ellipses are the ones of prescribed diameter contained between the parallel lines and containing D_π. The second column is $\tilde{\pi}_2 = x^2y$ and some π with $\pi \circ \phi = \tilde{\pi}_2$. Here, there exists ellipses quadri-tangent to $\Lambda(\pi)$, drawn in light green, but no such ellipse of maximum area as they can grow arbitrarily large inside the widest branch of $\Lambda(\pi)$. The optimal ellipses for a given α transition from the small circles to D_π, then to the growing ellipses containing D_π, in blue, until they become quadri-tangent. For precisely $\tilde{\pi}_2$, the circle D_π is already quadri-tangent, thus all the blue ellipses associated to the regime for $\alpha \in [\alpha_\pi, \mu_\pi]$ reduce to D_π. In the last two columns, associated to $\tilde{\pi}_3$ and $\tilde{\pi}_4$ all four regimes are visible on the graph of $\Lambda(\pi \circ \phi)$, and the optimal ellipse unconstrained ellipse E_π is drawn in thick green.

The overall computation of the anisotropic measure $\mathcal{Q} = \mathcal{Q}_{\pi,\alpha}$ for quadratic interpolation is summarized in Algorithm 2. Additional steps are included so that the method is numerically robust. First, if all error coefficients are under a small threshold $\epsilon_{\text{isotropic}}$, then the solution is considered to be perfectly captured by the current mesh. The region is isotropic and we set $\mathcal{Q} = h_{\max}^{-2} I$. It will be smoothed during the gradation step. Then, we check if the error polynomial is divisible by y, thus if we are in the case 5 of Proposition 5.2 up to a threshold $\epsilon_{\text{rotation}}$. If it is the case, a rotation R has to be applied to the arguments (x, y) so that $\pi(Rx, Ry)$ is no longer divisible by y. The rotation is actually applied to the coefficients of the polynomial, so that the new coefficients satisfy $\pi'(x, y) = \pi(Rx, Ry)$. If a rotation was applied, the matrix $\mathcal{Q}$ is rotated back by $R^{-1} = R^T$ before returning. Lastly, to avoid computing the discriminant of very large or very small coefficients, all coefficients are scaled so that the largest coefficient is $\mathcal{O}(1)$. The matrix $\mathcal{Q}$ is scaled back with a power $2/m = 2/3$ before returning.

5.4.2 Log-simplex method for higher-order interpolation

For interpolation of degree $k \geq 3$, there is no explicit form for the upper bound $\mathcal{Q}$. Several algorithms have been proposed [10, 11, 77, 90] and we implemented the *log-simplex* method from Coulaud and Loseille. The original method addresses the problem of finding $\mathcal{Q}$ for the L^p norm. It is essentially a linearization of problem (5.24), solved iteratively until the linearized constraints match the original constraints. We briefly detail the method in two

Algorithm 2 Analytic anisotropic measure $\mathcal{Q}_{\pi,\alpha}$ for quadratic interpolation.

Input: Coefficients a, b, c, d of the cubic error polynomial, maximum diameter $\alpha > 0$.
Output: Anisotropic measure $\mathcal{Q}(\boldsymbol{p}) = \mathcal{Q}_{\pi,\alpha}$.

function AnisotropicMeasureQuadratic(a, b, c, d, α)
  if All coefficients are smaller than $\epsilon_{\text{isotropic}}$ **then**
    return I ▷ Identity matrix, isotropic region
  end if
  `wasRotated` = false
  if $|a| < \epsilon_{\text{rotation}}$ **then** ▷ Check if π is divisible by y
    `wasRotated` = true
    $[a, b, c, d, R]$ = rotation(a, b, c, d)
  end if

  `scaling` = max($|a|, |b|, |c|, |d|$)
  $[a, b, c, d] = [a, b, c, d]$/`scaling` ▷ Scale the coefficients

  [hess π, disc π] = getHessianAndDiscriminant(a, b, c, d)
  Set `case` to 1, 2, 3 or 4 according to Proposition 5.2
  Factorize π into $\lambda(x - r_1 y)(x - r_2 y)(x - r_3 y)$
  Compute ϕ_π with Proposition 5.2 depending on `case`
  Compute μ_π with either (5.66), (40) or (41)
  Compute U_π with (5.67)
  Compute α_π with (46) or (47) depending on `case`
  Set $\beta_\pi = 0$

  if `case` = 3 or 4 **then**
    Compute unconstrained solution $\mathcal{Q}_\pi$ with Proposition 5.3
    $\beta_\pi = \lambda_{\min}(\mathcal{Q}_\pi)$
  end if

  if $\alpha \geq \mu_\pi$ **then**
    $\mathcal{Q}_{\pi,\alpha} = \alpha I$
  else if $\alpha \geq \alpha_\pi$ **then**
    Compute $\mathcal{Q}_{\pi,\alpha}$ with (5.70)
  else if $\alpha \geq \beta_\pi$ **then**
    Compute λ_α by solving (51) depending on `case`
    Compute $\mathcal{Q}_{\pi,\alpha}$ with (5.71), (5.72) or (5.73) depending on `case`
  else
    $\mathcal{Q}_{\pi,\alpha} = \mathcal{Q}_\pi$ ▷ Unconstrained solution
  end if

  $\mathcal{Q}_{\pi,\alpha} =$ `scaling`$^{2/3}\mathcal{Q}_{\pi,\alpha}$ ▷ Scale back with power $2/m = 2/3$
  if `wasRotated` **then**
    $\mathcal{Q}_{\pi,\alpha} = R^T \mathcal{Q}_{\pi,\alpha} R$ ▷ Rotate back
  end if
  return $\mathcal{Q}_{\pi,\alpha}$
end function

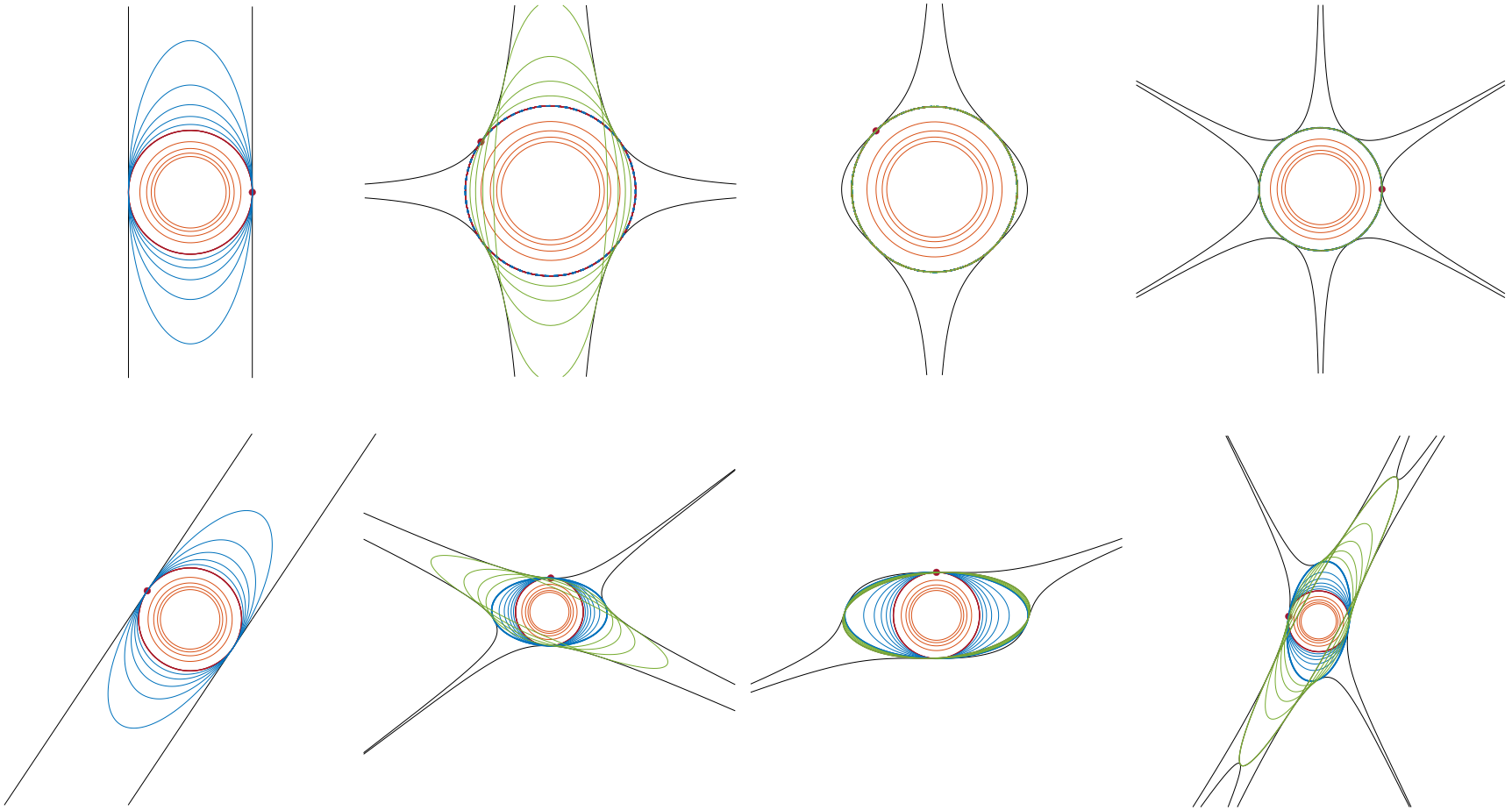


Figure 5.2 Largest ellipses included in $\Lambda(\pi)$ for each of the four reference polynomials and four polynomials mapped from them through a linear transformation. In orange: the optimal ellipses for $\alpha > \mu_\pi$ are circles strictly included in $\Lambda(\pi)$. In thick red: D_π, the largest disc inscribed in $\Lambda(\pi)$ and the contact point z_π (red dot). In blue: the ellipses inscribed in $\Lambda(\pi)$ and containing D_π. E_{π,α_π}, the largest of them, is in thick blue. In green: the transition ellipses between E_{π,α_π} and the unconstrained optimum E_π (in thick green).

dimensions, and we refer to [11, 12] for the treatment of the problem in three dimensions, as well as for more details and examples. The problem definition is identical to the quadratic case and we consider the minimization problem (5.24) for $\pi \in \mathbb{H}^m$ with arbitrary m:

$$\mathcal{Q} = \arg\inf \left\{ \det M \mid M \in \mathrm{Sym}_2^+, \quad \boldsymbol{z}^T M \boldsymbol{z} \geq 1, \quad \forall \boldsymbol{z} \in \Lambda(\pi) \right\}$$

This problem is nonlinear and nonconvex, because the determinant is nonconvex over matrices in Sym_2^+, and is subject to infinitely many linear constraints. A direct discrete counterpart is obtained by discretizing the set $\Lambda(\pi)$ into N_Λ points:

$$\mathcal{Q} = \arg\min \left\{ \det M \mid M \in \mathrm{Sym}_2^+, \quad \boldsymbol{z}_i^T M \boldsymbol{z}_i \geq 1, \quad \boldsymbol{z}_i \in \Lambda(\pi), \quad i = 1, ..., N_\Lambda \right\}. \tag{5.74}$$

Since π is homogeneous, the points $\boldsymbol{z}_i$ on the level set 1 are easily obtained by mapping points on the unit circle. The discrete problem is ill-posed, however, because it is always possible to find an arbitrary long ellipse, whose determinant goes to zero, passing through four points of $\Lambda(\pi)$ and satisfying the linear constraints. To overcome this difficulty, Coulaud and Loseille proposed to solve a linearized discrete problem instead, written for the matrix logarithm $\mathcal{L} = \ln M$ and called the *log-problem*. This is motivated by the fact that $\det \mathcal{Q} = \exp(\mathrm{tr}\, \mathcal{L})$, hence minimizing the linear function $\mathrm{tr}\, \mathcal{L}$ amounts to minimize $\det \mathcal{Q}$ as the (scalar) exponential is increasing. Moreover, while the initial problem allows $\det \mathcal{Q} = 0$, the trace of the log-matrix cannot go to $-\infty$, so that $\det \mathcal{Q} > 0$. The matrix $\mathcal{L}$ is symmetric, but no longer positive-definite in general, and the log-problem writes:

$$\min \left\{ \mathrm{tr}\, \mathcal{L} \mid \mathcal{L} \in \mathrm{Sym}_2, \quad \boldsymbol{z}_i^T (\exp \mathcal{L}) \boldsymbol{z}_i \geq 1, \quad \boldsymbol{z}_i \in \Lambda(\pi), \quad i = 1, ..., N_\Lambda \right\}. \tag{5.75}$$

Here, both the logarithm and the exponential are the matrix operators defined in Section 3.2.4. This problem is linear in the coefficients a, b, c of $\mathcal{L}$, but is now subject to nonlinear constraints. The first idea of the log-simplex method is to use the convexity of the exponential to write (5.75) with approximate linear constraints:

$$\min \left\{ \mathrm{tr}\, \mathcal{L} \mid \mathcal{L} \in \mathrm{Sym}_2, \quad \boldsymbol{z}_i^T \mathcal{L} \boldsymbol{z}_i \geq -\|\boldsymbol{z}_i\|^2 \ln(\|\boldsymbol{z}_i\|^2), \quad \boldsymbol{z}_i \in \Lambda(\pi), \quad i = 1, ..., N_\Lambda \right\}. \tag{5.76}$$

The problem (5.76) is now linear with linear constraints, and can be solved with e.g. the simplex algorithm, which seeks an optimum in $\mathbb{R}^{\dim(\mathcal{L})}$ ($\mathbb{R}^3$ in two dimensions) on the polytope formed by the N_Λ hyperplanes representing the linear constraints. It is also well-posed, in the sense that the trace of $\mathcal{L}$ is bounded from below, hence the volume of the optimal ellipse is

bounded from above. The *general form* (in the sense of linear optimization) of (5.76) writes:

$$\begin{aligned} &\text{Minimize } \boldsymbol{g}^T\boldsymbol{x} = a + c \\ &\text{subject to } A\boldsymbol{x} \geq \boldsymbol{b}, \end{aligned} \tag{5.77}$$

with the coefficients $\boldsymbol{g}^T = (1, 0, 1)$, the vector of unknown $\boldsymbol{x}^T = (a, b, c)$ and the constraints:

$$A = \begin{pmatrix} x_1^2 & 2x_1y_1 & y_1^2 \\ \vdots & \vdots & \vdots \\ x_{N_\Lambda}^2 & 2x_{N_\Lambda}y_{N_\Lambda} & y_{N_\Lambda}^2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} -\|\boldsymbol{z}_1\|^2 \ln(\|\boldsymbol{z}_1\|^2) \\ \vdots \\ -\|\boldsymbol{z}_{N_\Lambda}\|^2 \ln(\|\boldsymbol{z}_{N_\Lambda}\|^2) \end{pmatrix}. \tag{5.78}$$

In our implementation, we use the open-source library SoPlex, part of the SCIP Optimization Suite [91], to solve the dual problem to (5.76), then recover the solution $\mathcal{L}$ of the primal problem. Then, the matrix $\mathcal{Q}$ is recovered with $\mathcal{Q} = \exp \mathcal{L}$.

Iterative algorithm

Although the optimization problem (5.76) is linear and easily solved, the linearized constraints can be far from the original set $\Lambda(\pi)$, yielding an inaccurate upper bound $\mathcal{Q}$ on the higher order derivatives. The second idea of the log-simplex method is to construct a sequence of metrics $\mathcal{Q}_j$ that converges to the solution $\mathcal{Q}$ of the original problem. This is motivated by the following observation. From (4.110), the anisotropic measure can be computed either in the reference space or the physical space and satisfies respectively for linear triangles:

$$\begin{aligned} \left| \sum_{|\alpha|=m} D^\alpha (f \circ F)(\boldsymbol{y})\, \boldsymbol{\zeta}^\alpha \right| &= |\pi_{\boldsymbol{y}}(\boldsymbol{\zeta})| \leq \left(\boldsymbol{\zeta}^T \mathcal{Q}_\triangle(\boldsymbol{y}) \boldsymbol{\zeta} \right)^{\frac{m}{2}}, \qquad \forall\, \boldsymbol{\zeta} \in \mathbb{R}^2, \\ \left| \sum_{|\alpha|=m} D^\alpha f(\boldsymbol{p})\, \boldsymbol{\xi}^\alpha \right| &= |\pi_{\boldsymbol{p}}(\boldsymbol{\xi})| \leq \left(\boldsymbol{\xi}^T \mathcal{Q}(\boldsymbol{p}) \boldsymbol{\xi} \right)^{\frac{m}{2}}, \qquad \forall\, \boldsymbol{\xi} \in \mathbb{R}^2, \end{aligned} \tag{5.79}$$

for two points $\boldsymbol{y}$ and $\boldsymbol{p} = F(\boldsymbol{y})$ in the reference and physical space respectively. For linear triangles, the linear transformation is $F(\boldsymbol{y}) = \mathcal{M}^{-1/2}\boldsymbol{y}$, so that $\pi_{\boldsymbol{y}} = (\pi_{\boldsymbol{p}} \circ \mathcal{M}^{-1/2})(\boldsymbol{\zeta})$ and both anisotropic measures are related by $\mathcal{Q}(\boldsymbol{p}) = \mathcal{Q}(\mathcal{Q}_\triangle, \mathcal{M}) = \mathcal{M}^{1/2}\mathcal{Q}_\triangle(\boldsymbol{y})\mathcal{M}^{1/2}$. Now, the key hypothesis is to assume that the sought anisotropic measure $\mathcal{Q}(\boldsymbol{p})$ is also the target metric $\mathcal{M}$, so that it satisfies the fixed-point relation $\mathcal{Q}(\mathcal{Q}_\triangle, \mathcal{M}) = \mathcal{M}$. In other words, we seek the tangent space on which the metric is precisely $\mathcal{Q}$. From Proposition 5.1, the problem

above in the reference space is equivalent by homogeneity to:

$$\tilde{\boldsymbol{z}}^T \mathcal{Q}_\triangle \tilde{\boldsymbol{z}} \geq 1, \quad \forall \tilde{\boldsymbol{z}} \in \Lambda(\pi_{\boldsymbol{p}} \circ \mathcal{M}^{-1/2}). \tag{5.80}$$

The associated log-problem can be solved to determine $\mathcal{L}_\triangle = \ln \mathcal{Q}_\triangle$, considering the points $\tilde{\boldsymbol{z}}_i$ on the discretized level curve 1 of $\pi_{\boldsymbol{p}} \circ \mathcal{M}^{-1/2}$ instead of π. Then, $\mathcal{Q}(\boldsymbol{p})$ is recovered by $\mathcal{Q}(\boldsymbol{p}) = \mathcal{M}^{1/2}\mathcal{Q}_\triangle\mathcal{M}^{1/2} = \mathcal{M}^{1/2}(\exp \mathcal{L}_\triangle)\mathcal{M}^{1/2}$, and we set $\mathcal{M} = \mathcal{Q}$ for the next fixed-point iteration. Starting from $\mathcal{M}_0 = I$, the fixed-point method is summarized as follows:

$$\begin{aligned} \mathcal{Q}_{\triangle,i} &= \exp(\mathcal{L}_\triangle(\mathcal{M}_i^{-1/2})), \\ \mathcal{Q}_i &= \mathcal{M}_i^{1/2}\mathcal{Q}_{\triangle,i}\mathcal{M}_i^{1/2}, \\ \mathcal{M}_{i+1} &= \mathcal{Q}_i. \end{aligned} \tag{5.81}$$

Note that we actually solve the log-problem in the reference space only. At each iteration, the constraints of the log-problem are described by the points $\tilde{\boldsymbol{z}}_i \in \Lambda(\pi_{\boldsymbol{p}} \circ \mathcal{M}^{-1/2}) = \Lambda(\pi_{\boldsymbol{p}} \circ \mathcal{Q}^{-1/2})$. In the original paper, the $\tilde{\boldsymbol{z}}_i$ are computed from points $\boldsymbol{u}_i$ on the unit circle. Because π is homogeneous, they are immediately given by:

$$\tilde{\boldsymbol{z}}_i = \frac{\boldsymbol{u}_i}{\left|\pi_{\boldsymbol{p}}(\mathcal{Q}_j^{-\frac{1}{2}}\boldsymbol{u}_i)\right|^{\frac{1}{m}}} \quad \text{for} \;\; i = 1 \to N_\Lambda. \tag{5.82}$$

However, since $|\pi(\boldsymbol{z}_i)| = |\pi(\mathcal{Q}_j^{-\frac{1}{2}}\mathcal{Q}_j^{\frac{1}{2}}\boldsymbol{z}_i)| = 1$ for each $\boldsymbol{z}_i \in \Lambda(\pi)$, we clearly have $|\pi(\mathcal{Q}_j^{-\frac{1}{2}}\tilde{\boldsymbol{z}}_i)| = 1$ for $\tilde{\boldsymbol{z}}_i = \mathcal{Q}_j^{\frac{1}{2}}\boldsymbol{z}_i$, thus the transformed points $\mathcal{Q}_j^{\frac{1}{2}}\boldsymbol{z}_i$ are in the level curve 1 of $|\pi_{\boldsymbol{p}} \circ \mathcal{Q}^{-1/2}|$. Hence, the $\boldsymbol{z}_i$ on the level curve of $|\pi|$ can be computed only once, and the modified constraints at each iteration are obtained by simply applying the linear transformation $\tilde{\boldsymbol{z}}_i = \mathcal{Q}_j^{\frac{1}{2}}\boldsymbol{z}_i$. The points $\boldsymbol{z}_i \in \Lambda(\pi)$ are obtained by:

$$\boldsymbol{z}_i = \frac{\boldsymbol{u}_i}{|\pi_{\boldsymbol{p}}(\boldsymbol{u}_i)|^{\frac{1}{m}}} \quad \text{for} \;\; i = 1 \to N_\Lambda. \tag{5.83}$$

One readily checks that:

$$|\pi_{\boldsymbol{p}}(\boldsymbol{z}_i)| = \left|\pi_{\boldsymbol{p}}\left(\frac{\boldsymbol{u}_i}{|\pi_{\boldsymbol{p}}(\boldsymbol{u}_i)|^{\frac{1}{m}}}\right)\right| = \left(\frac{1}{|\pi_{\boldsymbol{p}}(\boldsymbol{u}_i)|^{\frac{1}{m}}}\right)^m |\pi_{\boldsymbol{p}}(\boldsymbol{u}_i)| = 1. \tag{5.84}$$

Using the transformation $\tilde{\boldsymbol{z}}_i = \mathcal{Q}_j^{\frac{1}{2}}\boldsymbol{z}_i$, it is then straightforward to see that if the sequence $\mathcal{Q}_j$ converges to a matrix $\mathcal{Q} = \mathcal{M}$, then (i) the sequence of log-matrices $\mathcal{L}_{\triangle,j}$ converges to

$\mathcal{L}_\triangle = \ln \mathcal{Q}_\triangle = \ln(\mathcal{Q}^{-\frac{1}{2}}\mathcal{Q}\mathcal{Q}^{-\frac{1}{2}}) = \ln I = 0$, and (ii) the sequence of log-constraints $\tilde{\boldsymbol{z}}_i^T \mathcal{L}_{\triangle,j} \tilde{\boldsymbol{z}}_i \geq -\|\tilde{\boldsymbol{z}}_i\|^2 \ln(\|\tilde{\boldsymbol{z}}_i\|^2)$ from (5.76) converge to the initial constraints of (5.74):

$$\begin{aligned} \tilde{\boldsymbol{z}}_i^T \mathcal{L}_\triangle \tilde{\boldsymbol{z}}_i = 0 \geq -\|\tilde{\boldsymbol{z}}_i\|^2 \ln(\|\tilde{\boldsymbol{z}}_i\|^2) \\ \Leftrightarrow 0 \leq \ln(\|\tilde{\boldsymbol{z}}_i\|^2) \\ \Leftrightarrow 1 \leq \|\tilde{\boldsymbol{z}}_i\|^2 \\ \Leftrightarrow 1 \leq \|\mathcal{Q}^{\frac{1}{2}} \boldsymbol{z}_i\|^2 = \boldsymbol{z}_i^T \mathcal{Q}^{\frac{1}{2}} \mathcal{Q}^{\frac{1}{2}} \boldsymbol{z}_i = \boldsymbol{z}_i^T \mathcal{Q} \boldsymbol{z}_i, \end{aligned} \tag{5.85}$$

where all the norms are with respect to the Euclidean metric. This fixed-point method is summarized in Algorithm 3, where $\|.\|_F$ is the Frobenius norm.

Logsimplex method for the $W^{1,p}$ seminorm

For the $W^{1,p}$ seminorm, we solve problem (5.29) instead:

$$\mathcal{Q}' = \arg\inf \left\{ \det M \mid M \in \mathrm{Sym}_2^+, \quad \boldsymbol{z}^T M \boldsymbol{z} \geq 1, \quad \forall \boldsymbol{z} \in \Lambda(\nabla\pi) \right\}.$$

The method is the same as for the L^p case, but we consider the error curve of $\|\nabla\pi\|^2 = (\partial_x\pi)^2 + (\partial_y\pi)^2$, which is a homogeneous polynomial of degree nk. For instance, for cubic interpolation, let $a = \partial_{xxxx} f, b = 4\partial_{xxxy} f, c = 6\partial_{xxyy} f, d = 4\partial_{xyyy} f, e = \partial_{yyyy} f$. Then $\|\nabla\pi\|^2 \in \mathbb{H}^6$ writes:

$$\begin{aligned} \|\nabla\pi\|^2 = (16a^2 + b^2)x^6 + (24ab + 4bc)x^5 y + (9b^2 + 6bd + 4c^2 + 16ac)x^4 y^2 + \\ (16e^2 + d^2)y^6 + (24de + 4cd)xy^5 + (9d^2 + 6bd + 4c^2 + 16ec)x^2 y^4 \\ + (8ad + 8be + 12bc + 12cd)x^3 y^3. \end{aligned} \tag{5.86}$$

5.5 Illustrative examples

To illustrate the properties of the optimal metrics, we examine the convergence of the interpolation error on optimal linear meshes of Lagrange finite elements Π^k of degree $k = 1, 2, 3, 4$. The problems of interest are the interpolation of analytical fields and the steady incompressible flow past a NACA0012 airfoil. The Riemannian metrics are selected to minimize the interpolation error $f - \Pi^k f$ measured in $L^p = W^{0,p}$ norm with $p = 1, 2, \infty$ or in $H^1 = W^{1,2}$ seminorm. In practice, the L^∞ norm is approximated by setting $p = 100$. From (5.48), the

Algorithm 3 Log-simplex method for $\mathcal{Q}$ for interpolation of arbitrary order.

Input: Error polynomial $\pi = \pi(\boldsymbol{p}) \in \mathbb{H}^m$, number of points N_Λ on the discretized level curve $\Lambda(\pi)$, stopping tolerance ϵ, maximum number of iterations `maxIter`.

Output: Anisotropic measure $\mathcal{Q}(\boldsymbol{p})$.

function LogSimplex($\pi, N_\Lambda, \epsilon,$ `maxIter`)
  Set $j = 0$ and $\mathcal{Q}_0 = I$
  Discretize the level curve $\Lambda(\pi)$ into N_Λ points $\boldsymbol{z}_i$ with (5.83)
  `converged` = `false`
  while `not converged` **do**
    Compute the modified constraint points $(\tilde{\boldsymbol{z}}_i)_j = \mathcal{Q}_j^{\frac{1}{2}} \boldsymbol{z}_i$ for $i = 1 \to N_\Lambda$
    Solve (5.76) for $\mathcal{L}_{\triangle,j}$ with constraint points $(\tilde{\boldsymbol{z}}_i)_j$
    Recover the physical-space solution $\mathcal{Q}_{j+1} = \mathcal{Q}_j^{\frac{1}{2}} \exp \mathcal{L}_{\triangle,j} \mathcal{Q}_j^{\frac{1}{2}}$
    Compute stopping criterion `diff` $= \|\mathcal{Q}_{j+1} - \mathcal{Q}_j\|_F / \|\mathcal{Q}_j\|_F$
    Set $\mathcal{Q}_j = \mathcal{Q}_{j+1}$ and increment j
    if `diff` $< \epsilon$ or $j >$ `maxIter` **then**
      `converged` = `true`
    end if
  end while
  return $\mathcal{Q}_j$
end function

optimal Riemannian metric yielding an adapted mesh with roughly N_v vertices has the form:

$$\mathcal{M} = N_v^{\frac{2}{n}} \left(\int_\Omega (\det \mathcal{Q})^{\frac{p(k+1-s)}{2(p(k+1-s)+n)}} \, d\boldsymbol{x} \right)^{-\frac{2}{n}} (\det \mathcal{Q})^{\frac{-1}{(k+1-s)p+n}} \, \mathcal{Q}, \tag{5.87}$$

where the SPD matrix $\mathcal{Q}$ is an anisotropic measure and a tight upper bound of the derivatives of order $k+1$ of f. Its expression depends on the degree of the finite element approximation, and on the target norm or seminorm. For linear and quadratic elements and the L^p norms, we use the analytical solution for $\mathcal{Q}$ provided by Mirebeau in (5.64) and Proposition 5.4 respectively. The maximum diameter of the ellipse associated to $\mathcal{Q}$ is set to $\alpha = 100$. For cubic elements, we compute an approximation of $\mathcal{Q}$ with the log-simplex method of Coulaud and Loseille and solve iteratively the minimization problem (5.24). For the H^1 seminorm, we use the analytical expressions provided by Mirebeau in (5.30) and (5.31) for linear and quadratic elements, and solve the minimization problem (5.29) with the log-simplex method for cubic and quartic elements. Regardless of the target norm and degree of the interpolants, the high-order derivatives of the field of interest are needed to compute $\mathcal{Q}$. These derivatives are only available for analytic fields. For general applications, they are reconstructed at the mesh vertices by applying the method of Zhang and Naga [85], described in Appendix B.1, which consists in fitting in a least-squares sense a polynomial of degree $k+1$ to the nodal

values of the target field. The reconstructed gradient is the gradient of this polynomial, and derivatives of order $k+1$ are obtained by repeating this process $k+1$ times.

For both applications hereafter, the methodology is as follows. We start from a coarse triangulation and generate sequences of meshes $\mathcal{T}_{N_v}, \mathcal{T}_{2N_v}, \mathcal{T}_{4N_v}, \ldots$ adapted to metric fields $\mathcal{M}_{N_v}, \mathcal{M}_{2N_v}, \mathcal{M}_{4N_v}, \ldots$ of increasing complexity ($\sim$ number of vertices), starting with a target of N_v mesh vertices. For each target complexity, a quasi-unit mesh is generated by the open-source meshing library `mmg2d`, part of the `mmg` suite [67]. Since the mesh format used by our in-house finite element code is Gmsh 's native `.msh` format, the metric field is simply appended at the end of the input mesh file as a tensor-valued `NodeData`. The minimum and maximum authorized mesh sizes are set to 10^{-10} and 10^3 respectively. These sizes are enforced by limiting the eigenvalues of the metric field before passing it to `mmg2d`, which is then called without specifying a minimum or maximum size. Gradation is applied to the metric in the exp-metric space (see Section 3.6.3), with the parameter β depending on the application, Since the gradation is already enforced in the metric given to `mmg2d`, its own gradation is disabled by setting `-hgrad -1`. The complete call to `mmg2d` is thus:

```
mmg2d_O3 inputMMG.msh -hgrad -1 -o outputMMG.msh
```

To obtain a converged adapted mesh, this process is performed iteratively until little variation is observed in both the adapted mesh and the measured errors. In practice, the final mesh is the one obtained after 10 iterations. At each iteration, the optimal metric field is computed on the previous adapted mesh for better accuracy.

5.5.1 Analytical fields

We consider the following two fields proposed by Mirebeau [38] and Cao [77] respectively:

1. $f_1(x,y) = x^3 + x^2 y + \tanh\left(\dfrac{2x - \sin(5y)}{\delta}\right)$ on $\Omega = [-2,2] \times [-1,1]$, with $\delta = 0.1$,
2. $f_2(x,y) = x^2 + y^2 + x^3 + y^3 + \exp\left(-K(y - d_1(x))^2\right) + \exp\left(-K(y - d_2(x))^2\right)$ on $\Omega = [0,1]^2$, with $K = 10^4$ and:

$$d_1(x) = 1 + 2x(x - 1.25), \qquad d_2(x) = -(x - 0.2)(x - 1.25)/0.4.$$

These fields are depicted in Fig. 5.3 and are meant to mimic physical solutions of PDE exhibiting strong variations or shocks. The function f_1 features a smoothed sine-shaped

step, while f_2 displays abrupt variations along two parabolas. Each of these is combined with polynomial terms of order up to three, to illustrate how the different $W^{m,p}$ norms capture the small features of the solution. To evaluate the convergence rates for the interpolation error in various norms, we generate meshes for a target number of vertices of ranging from $N_v = 100$ to $51,200$. To show the impact of the norms on the adapted meshes, a very large gradation ($\beta = 5$) is enforced, only to prevent `mmg2d` to yield aberrant meshes when metric variations are large.

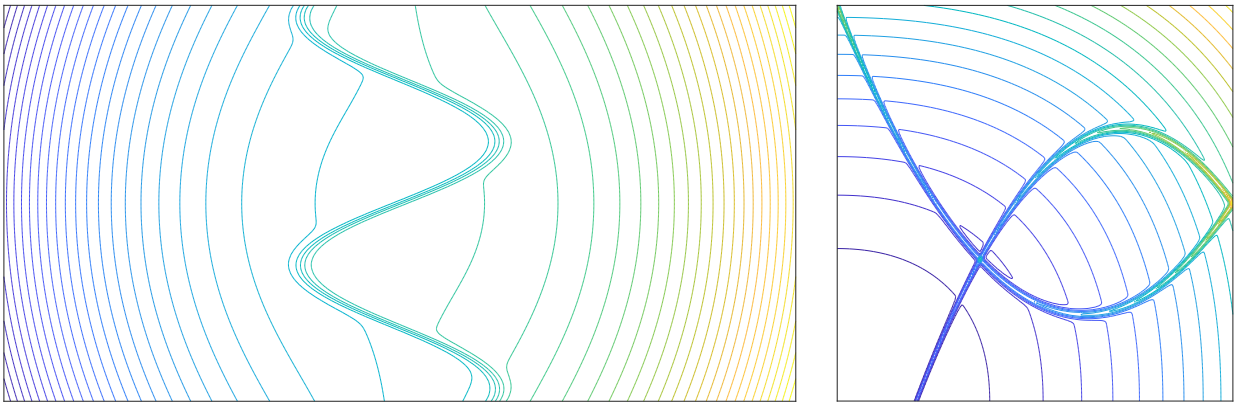


Figure 5.3 The fields f_1 and f_2.

Results

The impact of the different L^p norms is shown in Figs. 5.4 to 5.7 for f_1 and in Figs. 5.10 and 5.11 for f_2. For both fields, linear and quadratic elements do not capture the polynomial parts exactly. As a result, optimal metrics for the L^p norms with p small tend to distribute the mesh vertices more uniformly in the domain and not only around the discontinuities, to provide a better global control on the error. Metrics for the L^∞ norm and the H^1 seminorm, on the other hand, concentrate most of the resources along the regions of high gradients. For the two chosen fields, cubic and quartic elements capture exactly the polynomial terms, hence there is little difference between the L^p norms far from the discontinuities. To magnify this phenomenon, we also consider in Fig. 5.5 the field:

$$\widetilde{f}_1(x,y) = a(x^3 + x^2 y) + \tanh\left(\frac{2x - \sin(5y)}{\delta}\right),$$

where the polynomial terms are tuned by the parameter $a = 0.01$, so that $a(x^3 + x^2 y)$ can be seen as a low-amplitude perturbation to the main discontinuity. Here, the L^∞ norm and the H^1 seminorm now clearly allocate the vertices almost exclusively near the sine. Meshes

for f_1 with interpolation of degree 1 to 4 and for the L^2 norm and the H^1 seminorm are shown in Figs. 5.6 and 5.7 respectively. Cubic and quartic elements capture $x^3 + x^2 y$ exactly and the mesh size is governed by the maximum allowed size and the gradation only. The meshes adapted to f_2 exhibit the same behaviour. The differences between each interpolant are clearly visible on the close-ups of Fig. 5.11, as the mesh follows the derivatives of order $k+1$ of the solution.

Adaptation using optimal metrics for degree k interpolation on straight elements predicts a convergence rate of $k+1$ for the L^p norms and a rate of k for the H^1 seminorm. Those optimal rates are observed in Fig. 5.8 and 5.9 for all norms and all interpolants, with the exception of the H^1 seminorm for $k = 1$, which only converges at the rate of 0.5. We do not currently have an explanation for this issue. More importantly, these graphs show that the error in a given norm is minimal on the meshes adapted for this specific norm. For instance, the L^1 norm is minimal on the meshes adapted with $\mathcal{M}^{\text{opt}}_{m,s,p} = \mathcal{M}^{\text{opt}}_{2,0,1}$ in (5.48), see the close-ups in Fig. 5.8. The same conclusions are drawn for f_2 in Fig. 5.12.

5.5.2 Steady incompressible flow past a NACA0012 airfoil

We consider the steady and incompressible flow past a NACA0012 airfoil for an angle of attack between 0° and 15° and at Reynolds number $Re = 1000$, based on the chord c:

$$Re = \frac{c u_\infty}{\nu}, \tag{5.88}$$

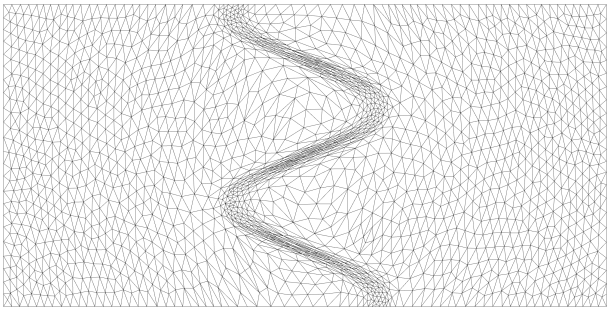
with u_∞ the far-field characteristic velocity and ν the kinematic viscosity of the fluid. We examine the prediction of lift and drag coefficients C_L and C_D, defined by:

$$C_L \triangleq \frac{F_y}{\frac{1}{2}\rho u_\infty^2 c^2}, \qquad C_D \triangleq \frac{F_x}{\frac{1}{2}\rho u_\infty^2 c^2}, \tag{5.89}$$

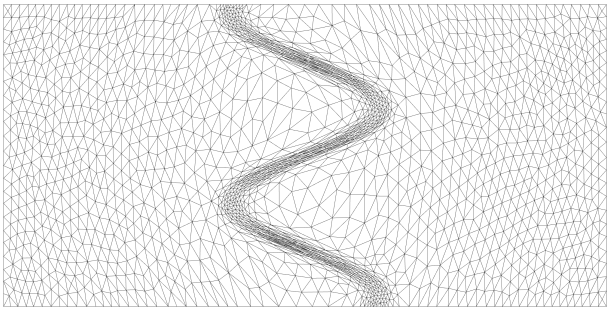
with F_x and F_y the components of the total aerodynamic forces on the profile and ρ is the fluid density. This test case is studied in the documentation of the open-source CFD-DEM solver LETHE[1], among others, and simulation data are available in the literature for comparison, see e.g. Kurtulus [92] and Kouser et al. [93]. The geometry, shown in Fig. 5.13, is a C-type domain obtained from the LETHE repository[2]. The steady incompressible Navier-Stokes equations are solved for the velocity and pressure $(\boldsymbol{u}, p)$, with our in-house solver. We use a

[1] https://lethe-cfd.github.io/lethe/documentation/examples/incompressible-flow/2d-naca0012-low-reynolds/2d-naca0012-low-reynolds.html

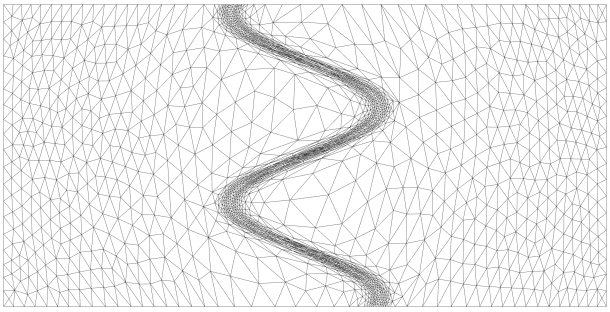
[2] https://github.com/lethe-cfd/lethe/tree/master/examples/incompressible-flow/2d-naca0012-low-reynolds



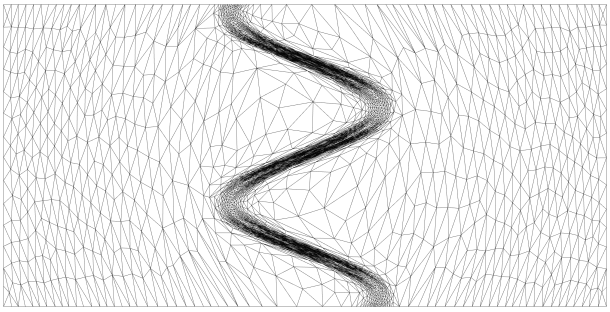
L^1 – 1,903 vertices – 3,614 triangles



L^2 – 1,908 vertices – 3,627 triangles

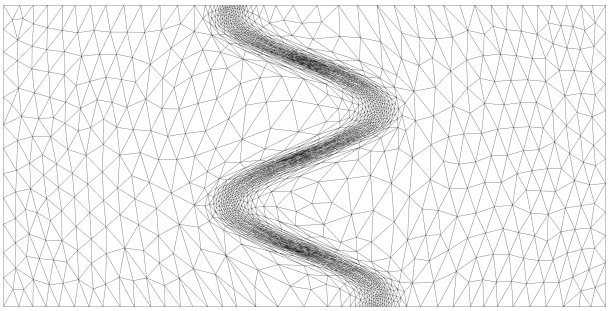


L^∞ – 1,858 vertices – 3,561 triangles

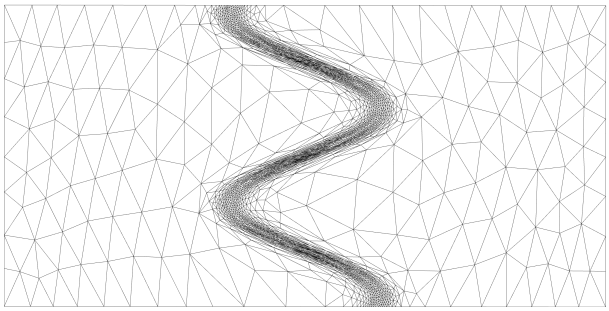


H^1 – 2,445 vertices – 4,701 triangles

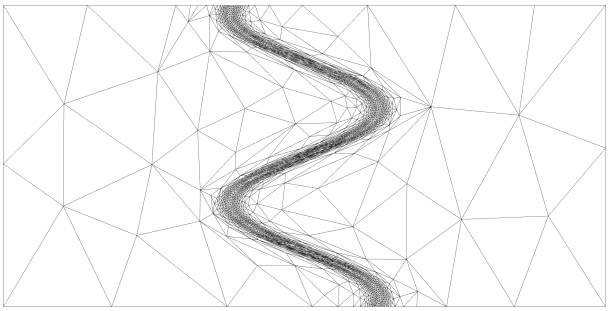
Figure 5.4 Linear interpolation for f_1. Target is 1,600 vertices.



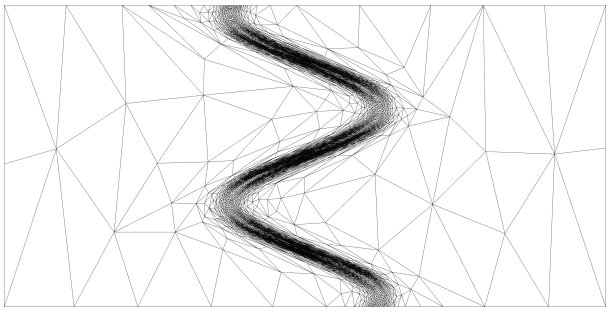
L^1 – 1,851 vertices – 3,570 triangles



L^2 – 1,824 vertices – 3,552 triangles

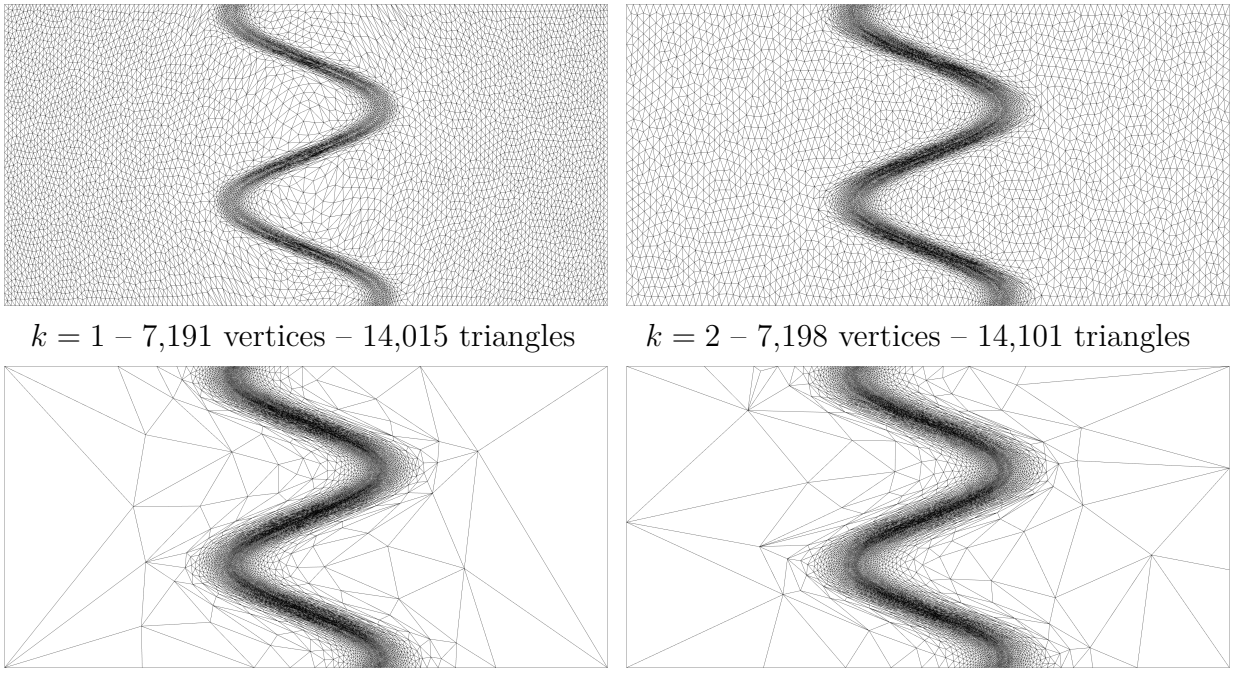


L^∞ – 1,799 vertices – 3,537 triangles



H^1 – 2,767 vertices – 5,455 triangles

Figure 5.5 Linear interpolation for $\widetilde{f}_1$. Target is 1,600 vertices.



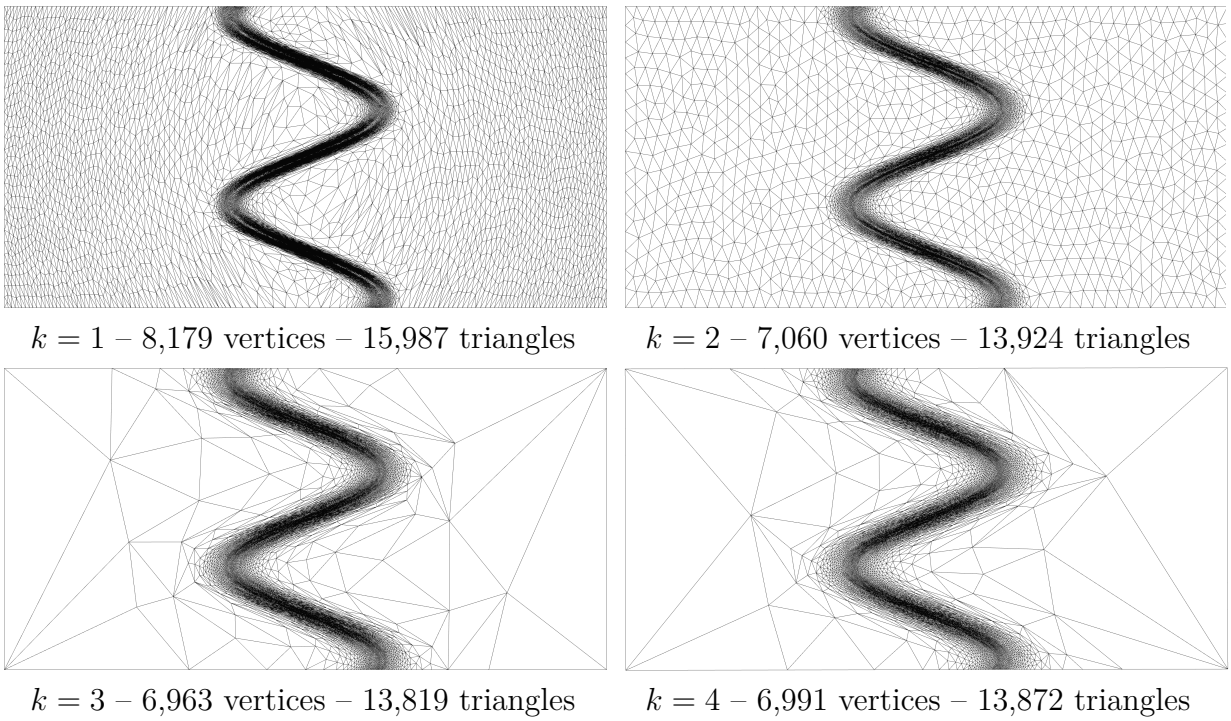
$k = 1$ – 7,191 vertices – 14,015 triangles

$k = 2$ – 7,198 vertices – 14,101 triangles

$k = 3$ – 7,007 vertices – 13,898 triangles

$k = 4$ – 7,007 vertices – 13,884 triangles

Figure 5.6 Interpolation of degree 1 to 4 for f_1 and the L^2 norm. Target is 6,400 vertices.



$k = 1$ – 8,179 vertices – 15,987 triangles

$k = 2$ – 7,060 vertices – 13,924 triangles

$k = 3$ – 6,963 vertices – 13,819 triangles

$k = 4$ – 6,991 vertices – 13,872 triangles

Figure 5.7 Interpolation of degree 1 to 4 for f_1 and the H^1 seminorm. Target is 6,400 vertices.

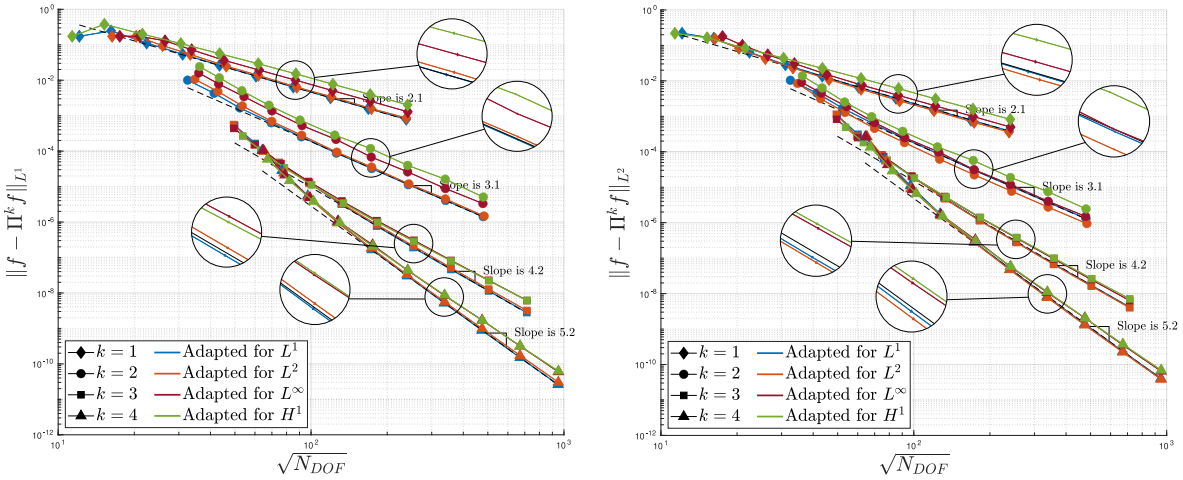


Figure 5.8 L^1 and L^2 norms of the interpolation error for f_1.

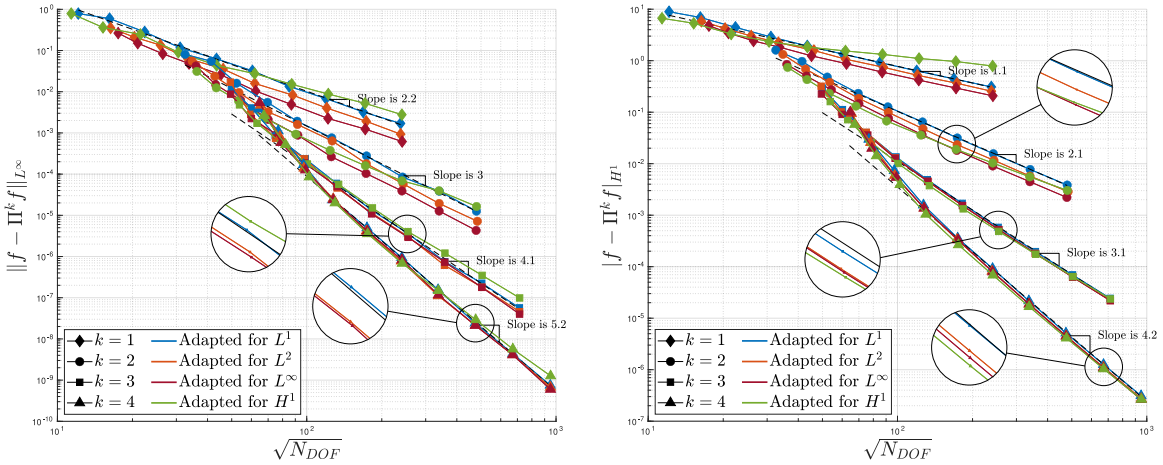
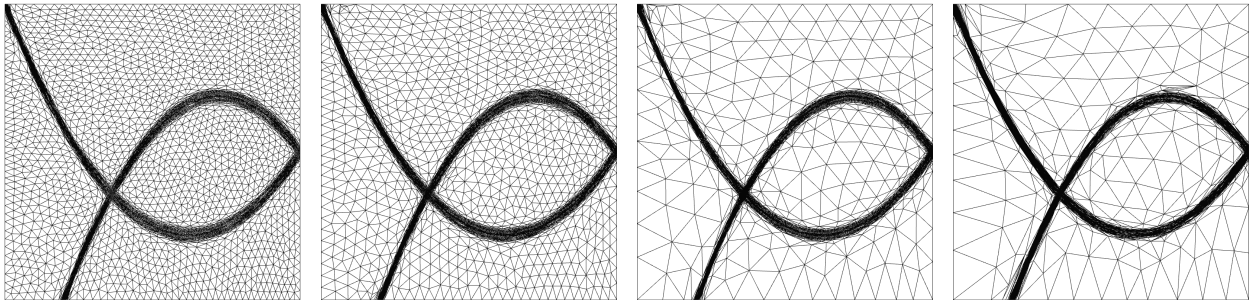


Figure 5.9 L^∞ norm and H^1 seminorm of the interpolation error for f_1.



L^1 – 3,734 vertices, 7,256 triangles
L^2 – 3,607 vertices, 7,038 triangles
L^∞ – 3,615 vertices, 7,102 triangles
H^1 – 3,782 vertices, 7,394 triangles

Figure 5.10 Linear interpolation for f_2. Target is 3,200 vertices.

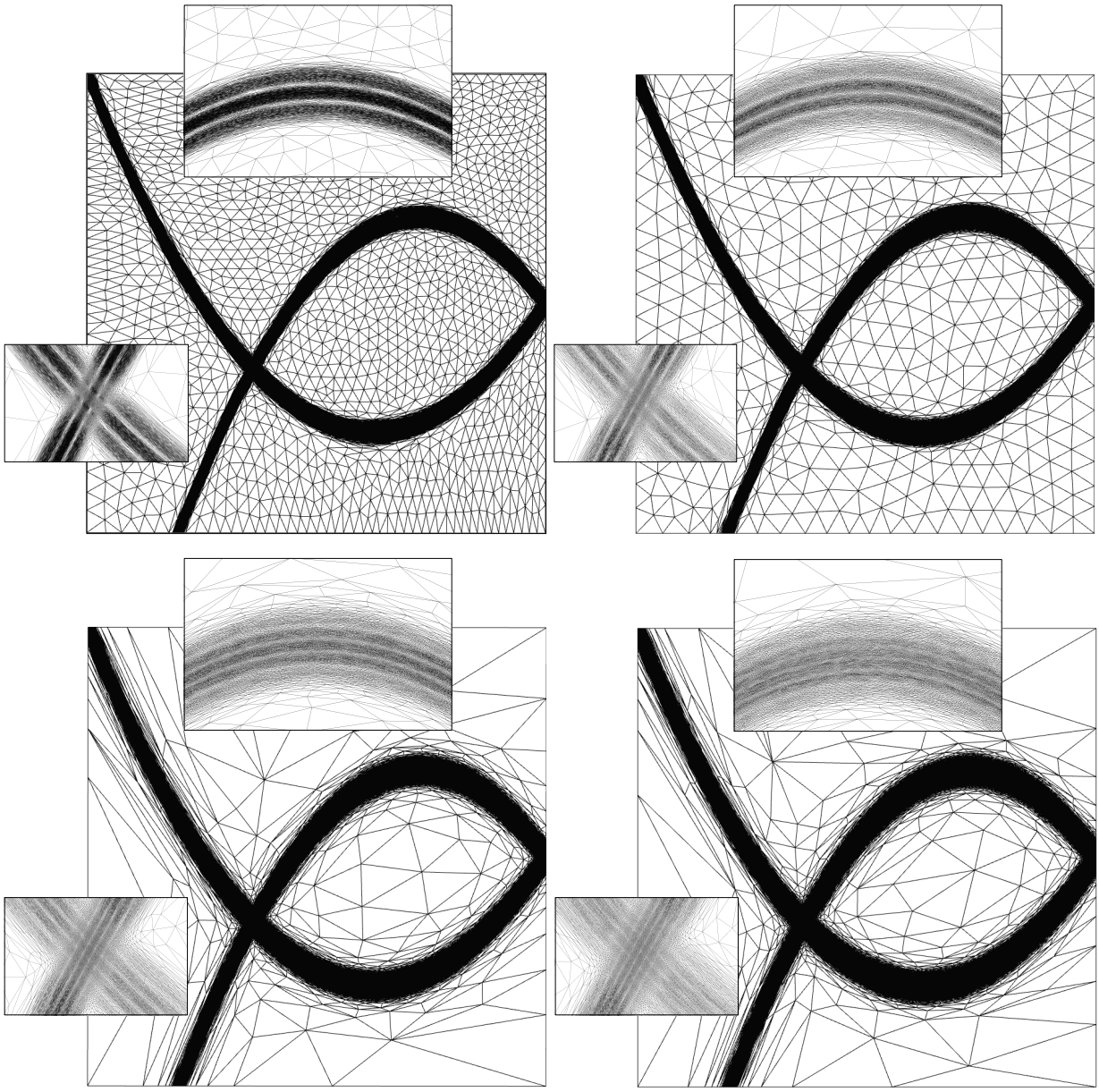


Figure 5.11 Interpolation of degree 1 to 4 for f_2 and the H^1 seminorm for 51,200 vertices.

mixed finite elements formulation to discretize the velocity and pressure spaces, using both Ladyzhenskaya–Babuška–Brezzi (LBB) stable Taylor-Hood elements $\Pi^k\Pi^{k-1}$ with $k = 2, 3$, as well as equal-order $\Pi^1\Pi^1$ elements with SUPG (streamline upwind Petrov-Galerkin) and PSPG (pressure-stabilizing Petrov-Galerkin) stabilizations. Since the number of degrees of freedom (DOF) remains modest throughout the tests (≤ 2 million DOF), we can afford to use a multithreaded direct linear solver, such as Intel MKL's implementation of PARDISO. The same boundary conditions as in the test case of Lethe are applied. On the inlet half-circle,

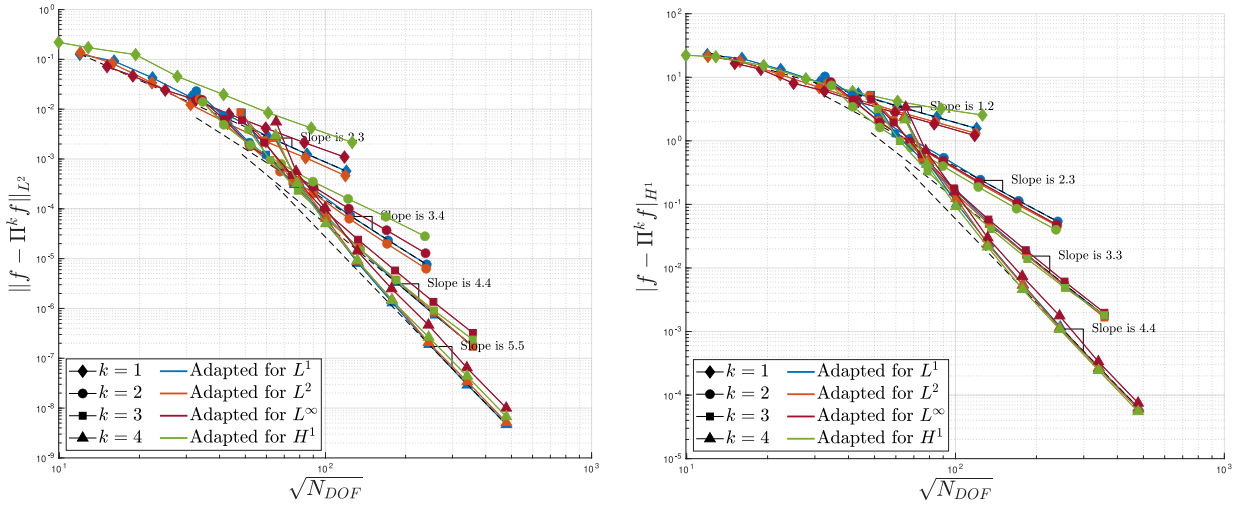


Figure 5.12 L^2 norm and H^1 seminorm of the interpolation error for f_2.

the velocity is set to $\boldsymbol{u}_{\text{inlet}} = (1, 0)$. On the top and bottom of the box, a slip condition $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ is applied, and the outlet is left free, so that a zero traction outflow condition is effectively applied. On the profile, a no-slip condition $\boldsymbol{u} = 0$ is applied and weakly enforced with a vector-valued Lagrange multiplier $\boldsymbol{\lambda}$. This way, the aerodynamic forces on the profile are, up to sign, the scalar components of $\boldsymbol{\lambda}$ and are straightforward to recover:

$$F_x = -\oint_{\text{airfoil}} \lambda_x \, d\ell, \qquad F_y = -\oint_{\text{airfoil}} \lambda_y \, d\ell. \tag{5.90}$$

The Lagrange multiplier $\boldsymbol{\lambda}$ is defined on the boundary of the airfoil and is discretized with line elements of the same degree as the velocity.

Following a similar example in Coulaud and Loseille [12], the main indicator for mesh adaptation is the interpolation error for the total pressure $p_0 = p + \frac{1}{2}\rho\|\boldsymbol{u}\|^2$. This way, both computed fields affecting the force coefficients are encoded in a single scalar indicator. The total pressure is discretized by polynomials of the same degree as the velocity, as it allows to recover p_0 at the DOFs where the solution $(\boldsymbol{u}, p)$ was computed by interpolating the pressure only. As for the analytic fields, we compare the impact of the choice of the Sobolev norm to minimize on the adapted meshes. For feature-based mesh adaptation on CFD applications, which is the case here, Alauzet et al. [] suggest minimizing the L^4 norm of the interpolation error, as it is a good compromise between accuracy in the far-field and in the regions of high gradients. The following results are thus computed to minimize either the L^2, L^4 or L^∞ norm of the interpolation error with respect to the total pressure, for a target complexity

ranging from 2,000 to 64,000. Additionally, the meshes of Fig. 5.17 are adapted for the norm of the velocity vector $\|\boldsymbol{u}\|$ and for the $H^1 = W^{1,2}$ seminorm. As there is no exact solution for this flow, the higher-order derivatives of either p_0 or $\|\boldsymbol{u}\|$, used to define the metric at each mesh vertex, must be estimated numerically with the method of Zhang and Naga. Recovery methods are known to perform poorly close to the boundaries, however, producing noise that is further amplified when high-order derivatives are to be recovered. For higher-order interpolants, this may thus lead to nonphysical refinement close to the boundaries in the final mesh.

For each target complexity, we start from an isotropic coarse mesh (with e.g. 8,937 vertices and 17,550 triangles for an angle of attack of 5°) and adapt the mesh 10 times. To avoid any projection error, the solution on each adapted mesh is recomputed from zero and not interpolated from the previous mesh. We compare our results for the force coefficients C_L and C_D on the adapted meshes to those available in the literature. To quantify the accuracy of the mesh adaptation, we examine the convergence (i) of the total pressure p_0, driving the adaptation process, and (ii) of the lift and drag coefficients. To estimate the error on the total pressure, we rely on the error estimate $e_{p_0} \simeq \widetilde{p}_0 - p_{0,h}$, where $\widetilde{p}_0$ is the local reconstruction of p_0 obtained from Zhang and Naga's method. To assess the convergence of the global force coefficients, we consider the error $|C_F - C_{F,\text{ref}}|$ with respect to the coefficients computed on the finest mesh (i.e., for a target of 64,000 vertices), where C_F stands for either C_L or C_D.

Results

The total pressure and adapted meshes for an angle of attack of 0° and 7° are shown in Fig. 5.13. The computations were performed with Taylor-Hood $\Pi^2\Pi^1$ elements and the adapted mesh minimize the L^2 norm of the interpolation error on p_0. The adapted meshes are refined around the airfoil and in the wake as expected, where the derivatives of p_0 are high. In particular, the triangles are very coarse in the far-field, but highly anisotropic in the wake, where the total pressure varies abruptly in the direction transverse to the flow.

Adapted meshes for the whole domain and for $\Pi^1\Pi^1$ and $\Pi^3\Pi^2$ elements are shown in Fig. 5.14. All meshes on this figure target a complexity of 8,000 vertices and effectively contain between around 9,000 and 9,500 vertices. For each discretization, the L^p norm of the interpolation error with respect to p_0 is minimized, with $p = 2, 4, \infty$. The impact of the choice of p in the minimized norm can clearly be seen for linear elements, with the L^2 and L^4 norms allocating considerably more vertices in the far-field than the L^∞ norm. For the latter, a

minimum number of vertices on the inlet half-circle is enforced to the mesh generator, to keep an accurate representation of the geometry, yielding the branch-like patterns on the inlet boundary. The effect of the choice of p in the L^p is also visible for $\Pi^3\Pi^2$ elements, as the L^∞ norm still yields a coarser mesh in the far-field, but the difference between the L^2 and the L^4 norm is more subtle, since some vertices are gathered near the boundaries, where the quality of the derivatives deteriorates.

The lift and drag predictions are compared with reference simulation results on the left of Fig. 5.15. The reference results from Kurtulus and Kouser et al. are time-averaged force coefficients originating from unsteady simulations, whereas we simulate the steady case. For an angle of attack of up to $7 - 8°$, no vortex shedding is observed in the wake of the airfoil in both reference simulations, so that the flow can be considered steady. Indeed, we observe good agreement between our force coefficients and those of the literature up to an angle of attack of 8°, after which the results of our steady simulations become nonphysical. Each point of data in Fig. 5.15 is the result of the simulation of the steady flow with Taylor-Hood $\Pi^3\Pi^2$ finite elements, on the last mesh obtained by adapting for a target of 8,000 vertices and minimizing the L^2 norm of the error on the total pressure. The effective number of vertices in the adapted meshes is typically around 9,400, yielding around 18,400 triangles. In comparison, the 2D simulations of Kurtulus were performed on uniform grids ranging from 94,000 to 250,000 triangles, and the 3D simulations of Kouser et al. were performed on a mesh obtained by extruding a base 2D mesh of about 200,000 quadrangles. Here, comparable results are obtained using a fifth to a tenth of the number of elements.

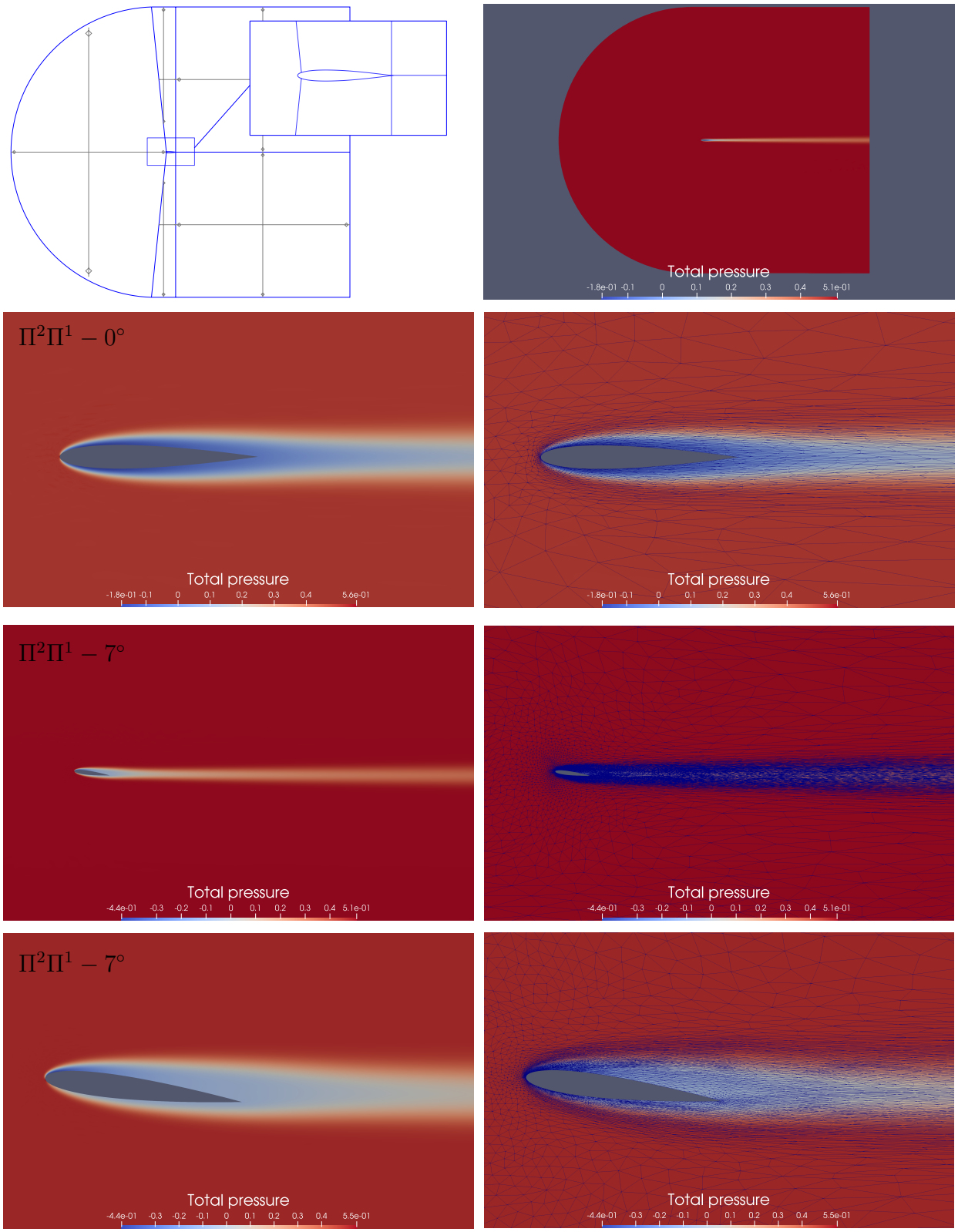


Figure 5.13 Top two rows: Geometry and steady incompressible flow past a NACA0012 airfoil at 0° inclination. Total pressure field computed with $\Pi^2\Pi^1$ finite elements and superimposed adapted mesh. Bottom two rows: Total pressure field for 7° inclination computed with $\Pi^2\Pi^1$ finite elements and superimposed adapted mesh. For both inclinations, the adaptation targets the L^2 norm.

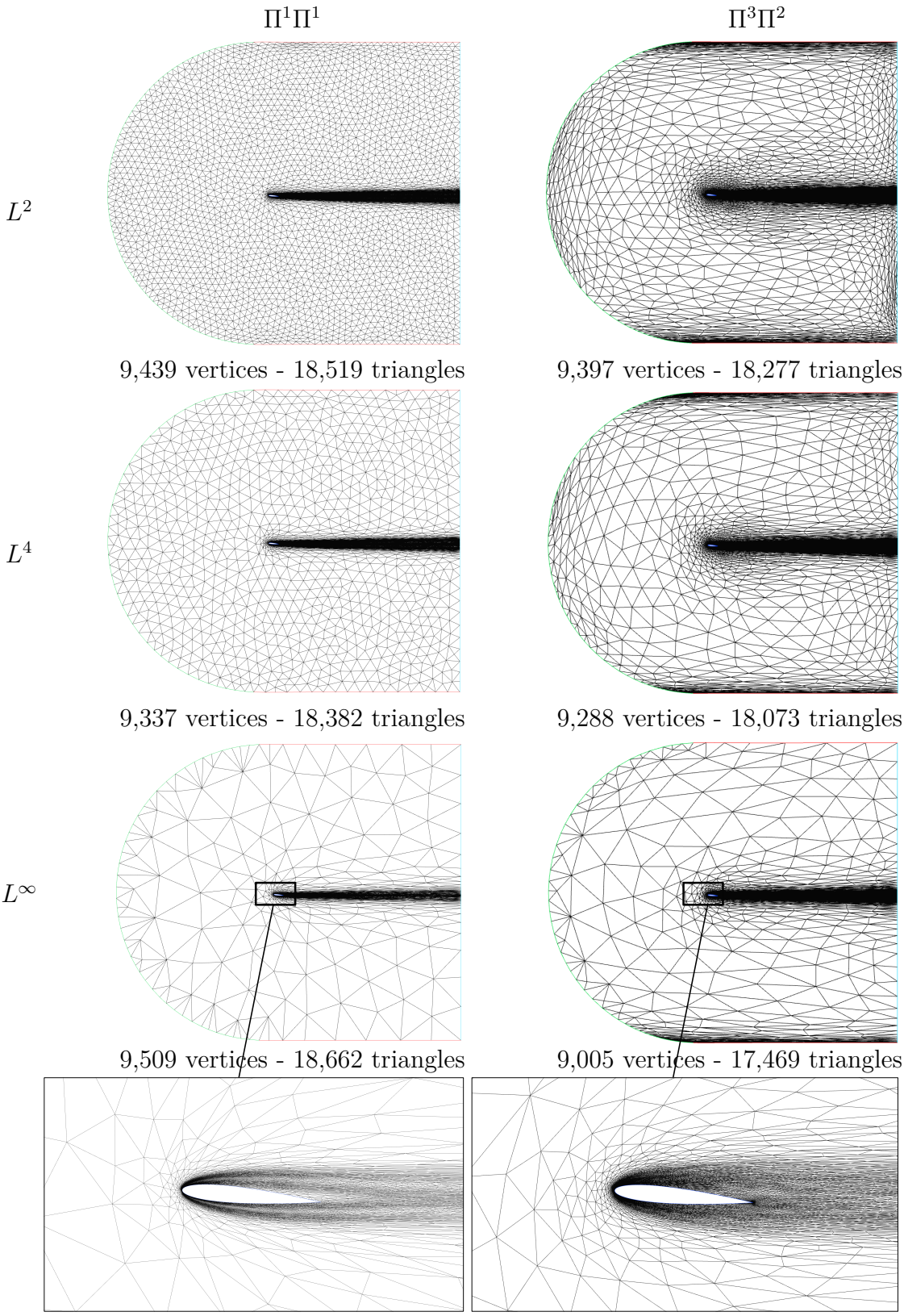


Figure 5.14 Adapted meshes for stabilized $\Pi^1\Pi^1$ (left) and Taylor-Hood $\Pi^3\Pi^2$ (right) finite elements and for a target complexity of 8,000 vertices. Minimization of the L^2 (top), L^4 (middle) and L^∞ (bottom) norm of the interpolation error for p_0.

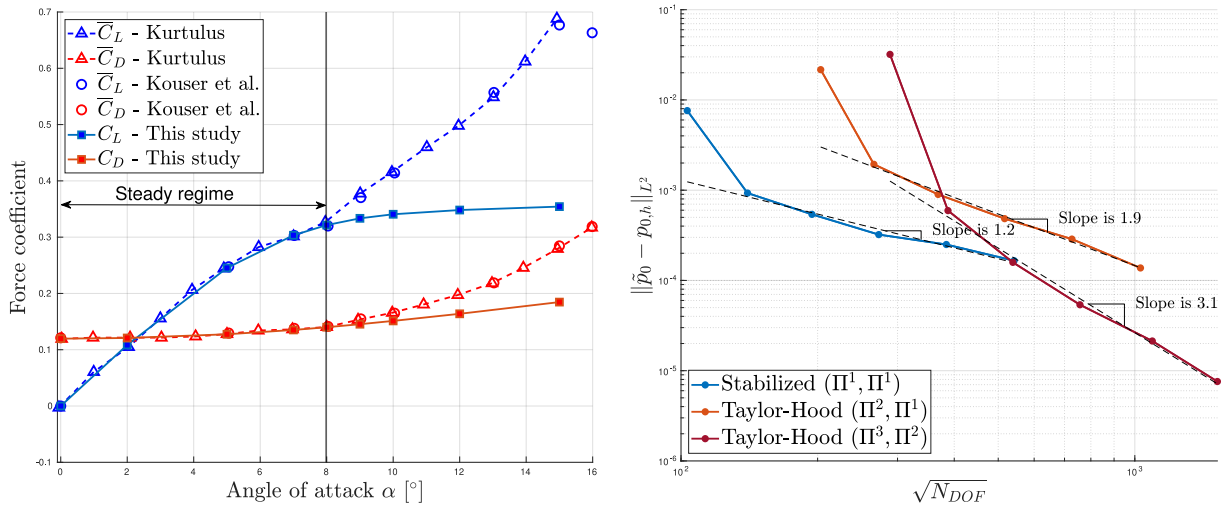


Figure 5.15 Left: Comparison of predicted lift and drag coefficients with results from reference simulations. Right: Estimated L^2 error on the total pressure versus the number of degrees of freedom.

As the meshes for validation with the literature are adapted with respect to the total pressure, we examine its convergence on the right of Fig. 5.15. For Taylor-Hood elements, the total pressure is obtained by combining the velocity field of degree k and the pressure of degree $k-1$, so that the pressure is the limiting field and we expect a convergence rate of k ($= 2$ and 3 for the considered pairs) for p_0, that is, an evolution in $\mathcal{O}(N_v^{k/2}) \sim \mathcal{O}(N_{\text{DOF}}^{k/2})$. To ease the notation, we let $h \triangleq \sqrt{N_{\text{DOF}}}$ denote the anisotropic equivalent of the characteristic meshsize, so that we expect a convergence in $\mathcal{O}(h^k)$. For $\Pi^1\Pi^1$ elements, the total pressure is linear, however we expect a convergence rate of $k = 1$ or higher, see e.g. Section 6.5. of [94] and the references therein for an account of the pressure convergence of stabilized equal-order methods. The observed rates in Fig. 5.15 are $\mathcal{O}(h^{1.9})$ and $\mathcal{O}(h^{3.1})$ for Taylor-Hood elements, and $\mathcal{O}(h^{1.2})$ for stabilized linear elements, in agreement with the predicted rates. At a low number of degrees of freedom, we also observe that stabilized linear elements achieve a reduced estimated error compared to the nonstabilized Taylor-Hood pairs.

Next, we turn to the convergence of the output functionals of interest, the lift and drag coefficients. When discretizing the Navier-Stokes equations with finite elements and when both the solution $(\boldsymbol{u}, p)$ and the solution of the adjoint problem written for the output are sufficiently smooth, superconvergence results for the outputs are documented in e.g. [95, 96] and the references therein. In particular, we can expect a convergence rate of up to $2k$ when finite elements of degree k are used. For Taylor-Hood elements, we should assume that this rate is based on the degree of the pressure, the most limiting one, however the obtained rates are close to twice the degree of the velocity discretization. The convergence for the

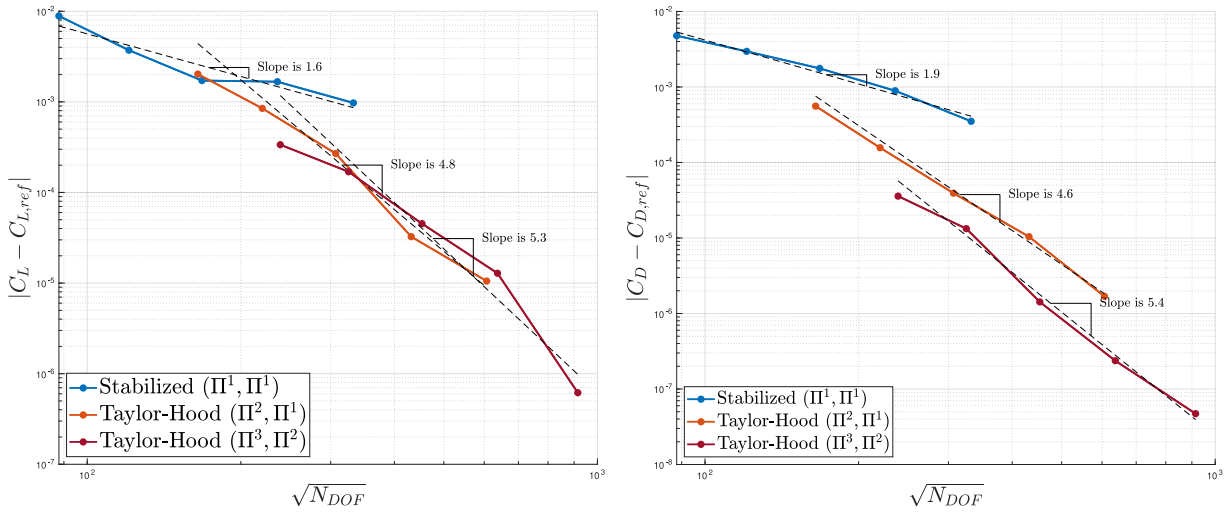


Figure 5.16 Convergence of the lift and drag coefficients with respect to the coefficients on the finest meshes versus the number of degrees of freedom.

lift and drag coefficients is shown in Fig. 5.16: for both coefficients and all finite elements configurations, the observed convergence rate is higher than the rate k obtained for the total pressure. Average rates of $\mathcal{O}(h^{1.6})$ and $\mathcal{O}(h^{1.9})$ are observed for the $\Pi^1\Pi^1$ pair, which is close to the optimal rate of $2k = 2$. For $\Pi^2\Pi^1$ elements, both coefficients converge at an average rate higher than $2k = 4$, twice the degree of the velocity discretization. Lastly, the convergence rates for $\Pi^3\Pi^2$ elements are $\mathcal{O}(h^{5.3})$ and $\mathcal{O}(h^{5.4})$, which are close to $2k = 6$. Note that some inherent uncertainty remains as the exact lift and drag are unknown, and are approximated by their values on the finest meshes.

Lastly, adapted meshes using the norm of the velocity $\|\boldsymbol{u}\|$ as indicator and targeting the H^1 seminorm are shown in Fig. 5.17. All meshes target 64,000 vertices. As feature-based adaptation relies on the higher-order derivatives of the field, we compare the adapted meshes to the Frobenius norm of the Hessian matrix and its higher-order generalizations, shown in logarithmic scale. This yields a scalar indicator of the intensity of the field derivatives around the airfoil and in the wake. The Hessian of $\|\boldsymbol{u}\|$ features three "crests" in the wake, and each additional derivatives adds crests. These crests are clearly identifiable in the meshes, especially for $\Pi^1\Pi^1$ and $\Pi^2\Pi^1$ elements.

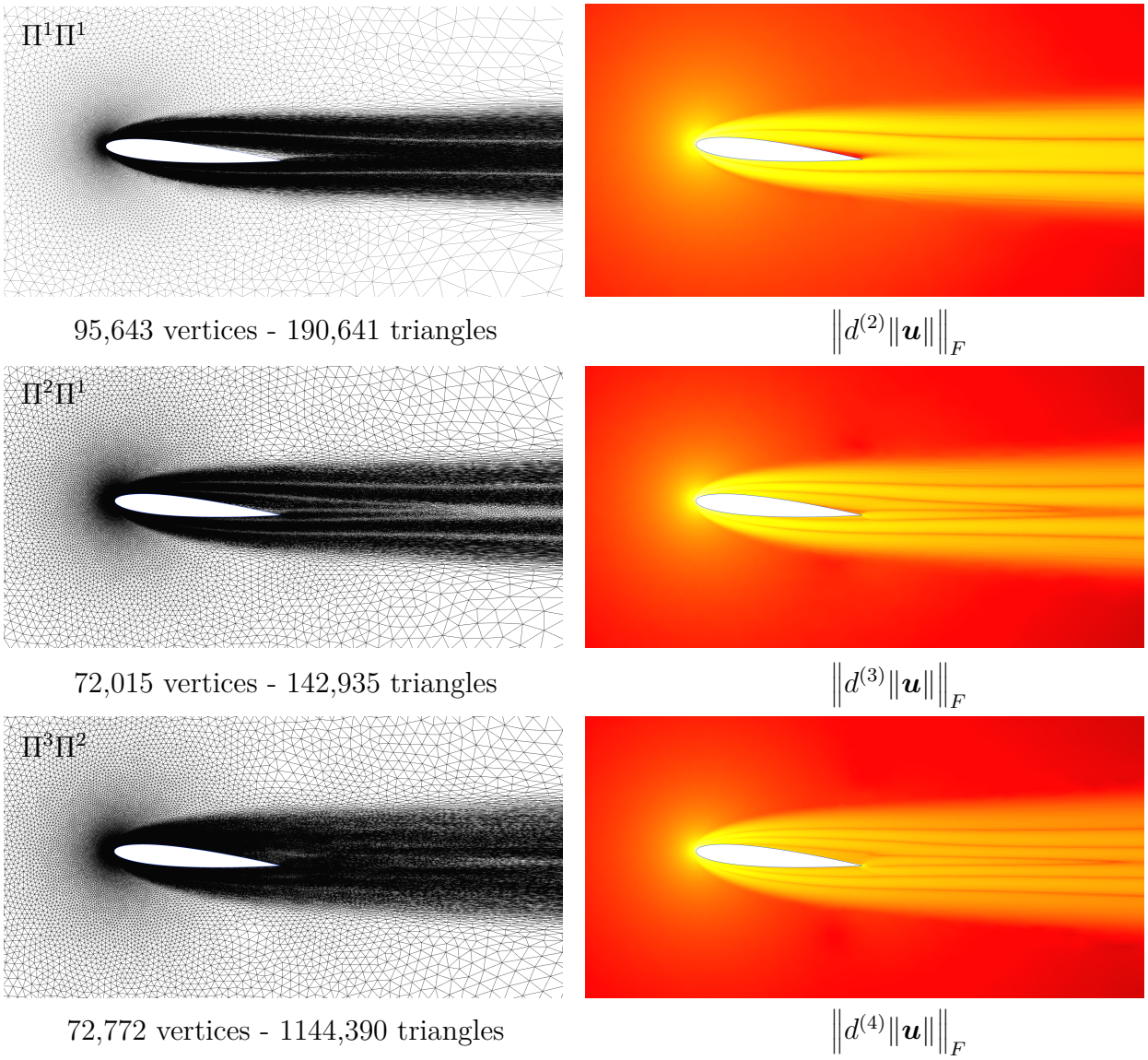


Figure 5.17 Left: Adapted meshes around the airfoil at an inclination of 5° and for $\Pi^1\Pi^1$ (top), $\Pi^2\Pi^1$ (middle) and $\Pi^3\Pi^2$ (bottom) elements. The indicator for mesh adaptation is the $H^1 = W^{1,2}$ seminorm of the interpolation error with respect to the norm of the velocity $\|\boldsymbol{u}\|$. All meshes are adapted for a target of 64,000 vertices. Right: Frobenius norm of the Hessian matrix and of the matrices of the third and fourth order derivatives around the airfoil.

5.6 Conclusion

In this chapter, we reviewed global interpolation error estimates based on both discrete triangulations and continuous Riemannian metrics. Among all triangulations with a target number of vertices or elements, the optimal triangulations are described by optimal metrics, so that we can manipulate metric fields and create an optimal mesh with any anisotropic meshing software. On linear meshes, the anisotropy is controlled by the derivatives of order $k+1$ of the interpolated field. For linear interpolation, this yields the well-known metric based on the Hessian. For higher-order interpolation, a metric $\mathcal{Q}$ mimics the role of the Hessian and bounds the higher-order derivatives. We presented the explicit formulation in terms of $\mathcal{Q}$ of the optimal Riemannian metrics to control the interpolation error in $W^{s,p}$ norm. The practical computation of $\mathcal{Q}$ is detailed for interpolation of arbitrary degree k: analytic solutions are available for $k = 1$ and 2, and a numerical approach is presented for $k \geq 3$. The interpolation properties of the resulting adapted meshes are discussed on both analytic fields and flow variables. For the considered applications, the error converges with the optimal rates, which validates our implementation.

CHAPTER 6 RIEMANNIAN METRICS FOR ANISOTROPIC ADAPTATION WITH CURVILINEAR TRIANGLES

We now turn to Riemannian metrics tailored for curved triangles, with once again the goal of controlling the interpolation error on the adapted mesh. Two Riemannian metrics for $\mathcal{P}^2$ mesh adaptation are discussed.

The first one, presented in Section 6.1, is adapted from the metric introduced in Zhang et al. [22], where for a field f, the principal directions of the metric are the isocontours and gradient curves of f. The principal sizes are devised to control the maximum error reached on second order approximations of these two families of curves, that is, on quadratic curves with matching tangent and curvature. This metric is essentially the pullback to the parameter space of the metric induced by the graph of f, but with modified eigenvalues. A recurring idea explored in this thesis is that to control the interpolation error, mesh adaptation should be performed with respect to the metric induced by a manifold derived from the higher-order derivatives of f, rather than from f itself. In this sense, the proposed metric is a compromise, as the directions are those of extreme variations of f, but the sizes reflect the error on the interpolant and involve higher-order derivatives. The anisotropy and approximation properties of this metric are studied, as well as some identified limitations. Thanks to the continuous mesh framework, its asymptotic convergence rate and, by extension, the one of similar metrics also derived from the induced metric, is evaluated. We observe that despite some shortcomings, it predicts second order convergence with respect to linear interpolation on $\mathcal{P}^2$ triangles for the tested fields and for acceptable complexities.

To derive the second metric, we tackle in Section 6.2 the problem of finding an optimal metric for $\mathcal{P}^2$ triangles with respect to the interpolation error. From Chapter 5, the optimal metric for e.g. the linear interpolation error on linear meshes is the minimizer:

$$\mathcal{M}^{\text{opt}} = \mathcal{M}^{\text{opt}}_{2,0,p} = \arg \min_{\mathcal{C}(\mathcal{M}) \leq N_v} \left(\int_\Omega \text{tr} \left(\mathcal{M}^{-\frac{1}{2}} |H_f| \mathcal{M}^{-\frac{1}{2}} \right)^p \, d\boldsymbol{x} \right)^{\frac{1}{p}}. \tag{6.1}$$

From the error estimate on higher-order meshes derived in Chapter 4, the extension of $\mathcal{M}^{\text{opt}}$ to $\mathcal{P}^2$ meshes minimizes instead:

$$\mathcal{M}^{\text{opt}} = \arg \min_{\mathcal{C}(\mathcal{M}) \leq N_v} \left(\int_\Omega \left| \text{tr} \left(\mathcal{M}^{-\frac{1}{2}} H_f \mathcal{M}^{-\frac{1}{2}} \right) + (\partial \mathcal{M}^{-\frac{1}{2}} : \mathcal{M}^{-\frac{1}{2}}) \cdot \nabla f \right|^p \, dx. \right)^{\frac{1}{p}}. \tag{6.2}$$

Finding $\mathcal{M}^{\text{opt}}$ requires minimizing over a functional in the metric and its derivatives. A

formal resolution, as presented in Loseille [1] for linear triangles, is still an open problem that was not tackled in this work. A numerical resolution should be possible within the Mesh Optimization via Error Sampling and Synthesis (MOESS) framework introduced by Yano and Darmofal [31], as it converges the numerical solution and the optimal metric together given an arbitrary error functional in $\mathcal{M}$. An implementation of this framework is not straightforward, however: it depends on the PDE and requires solving local problems on refined patches, whose boundary conditions are dictated by the frozen solution on the adjacent elements. Several aspects of the solver must be reworked, and the metric cannot be obtained by simply postprocessing a numerical solution, as is the case for the metrics considered so far. Instead, we explore in this chapter another avenue for a numerical solution to (6.2) by extending the log-simplex method discussed in Chapter 5 to curved triangles. Indeed, when the error minimization problem (6.1) takes the general form:

$$\mathcal{M}^{\text{opt}} = \arg \min_{\mathcal{C}(\mathcal{M}) \leq N_v} \left(\int_\Omega \operatorname{tr} \left(\mathcal{M}^{-\frac{1}{2}} \mathcal{Q} \mathcal{M}^{-\frac{1}{2}} \right)^{\frac{p(k+1)}{2}} d\boldsymbol{x} \right)^{\frac{1}{p}}, \tag{6.3}$$

the optimal metric is known from Mirebeau [38] and Coulaud and Loseille [11], and is given by a scaling of $\mathcal{Q}$. For linear elements, the metric $\mathcal{Q}$ bounds the derivatives of $f \circ F$ for a linear transformation $F = F_{\triangle \to K}$. If $\mathcal{Q}$ now bounds the derivatives of $f \circ F(\mathcal{M})$ and accounts for the nonlinearity of F, then $\mathcal{M}^{\text{opt}}$ is the metric minimizing the interpolation error for curved elements. To obtain $\mathcal{Q}$, we reformulate in the reference space the problem of finding an upper bound $\mathcal{Q}_\triangle$ to $D^{(k+1)}(f \circ F)$, then set the Riemannian metric of the physical space to this upper bound. This yields the nonlinear relation $\mathcal{M} = \mathcal{Q} = \mathcal{M}^{1/2} \mathcal{Q}_\triangle \mathcal{M}^{1/2}$ for $\mathcal{M}$, which can be solved with e.g. fixed-point iterations. For F linear and interpolation of degree ≤ 2, $\mathcal{Q}_\triangle$ can be solved analytically and $\mathcal{M}$ converges in a single fixed-point iteration. For interpolation with higher-order polynomials, this yields precisely the log-simplex method, as the fixed-point iterations were introduced in [11] to counteract the linearization of the log-problem for $\mathcal{Q}$. When F becomes nonlinear, the nature of the problem for $\mathcal{M}$ changes from a nonlinear equation to a system of quasilinear PDE in the metric components. As a natural, albeit naive, extension of the log-simplex method, this system is solved with fixed-point iterations involving the metric derivatives. This scheme proves hard to converge, but when it does, it converges to the upper bound $\mathcal{Q}$.

6.1 A metric to control the maximum interpolation error along the isocontours

We first introduce a Riemannian metric $\mathcal{M}$ which controls the maximum interpolation error along the isocontours and gradient curves of a target field f. This metric, which is referred

to as the **isocontours metric** in the following, was introduced in Zhang et al. [22]. The idea behind it is quite simple: if the elements are allowed to curve, some of their edges can be made to follow the curves of constant f without accumulating any error. Just as curvilinear edges are already used to better capture CAD boundaries, their goal here is to better capture the isocontours of f. The direction of the gradient completes the pair of principal directions, and the principal sizes are set to limit the interpolation error to a user-defined threshold along the integral curves of the direction field. Contrary to the metrics introduced thus far, the directions and sizes of $\mathcal{M}$ are thus set independently and the direction field is chosen beforehand. The relevant computations are detailed in Sections 6.1.1 through 6.1.3. Then, we take a step back in Section 6.1.4 and link this metric to the error estimates discussed in the previous chapters. We show that with this choice of principal directions, the error cannot be bounded by the imposed threshold in all directions around a point $\boldsymbol{x}_0$. In Section 6.1.5, the study of the asymptotic continuous error predicted by the isocontours metric on selected fields reveals that, although the error may not be *bounded* everywhere, it remains *controlled* in such a way that $\mathcal{M}$ predicts second-order convergence on the tested fields.

6.1.1 Direction field and smoothing

The SPD matrix representation of $\mathcal{M}$ decomposes as $[\mathcal{M}] = P\Lambda P^T$, with P the orthogonal matrix of its eigenvectors. Together with strictly positive eigenvalues, any two unit and orthogonal vector fields on $\mathbb{R}^2$ thus define a Riemannian metric. Given a function $f : \mathbb{R}^2 \to \mathbb{R}$, the gradient ∇f and its orthogonal $(\nabla f)^\perp$ are a natural choice of orthogonal directions which are defined at all noncritical points of f. As was done in Zhang et al. [22] and following the discussion above, we choose for $\mathcal{M}$ the directions defined by the unit vector fields:

$$\boldsymbol{g}^\perp \triangleq \frac{\nabla f^\perp}{\|\nabla f\|} = \frac{(-f_y, f_x)^T}{\|\nabla f\|}, \qquad \boldsymbol{g} \triangleq \frac{\nabla f}{\|\nabla f\|} = \frac{(f_x, f_y)^T}{\|\nabla f\|}. \tag{6.4}$$

Letting $S = (x, y, f(x, y)) \in \mathbb{R}^3$ be the graph of f, the vectors $\boldsymbol{g}^\perp$ and $\boldsymbol{g}$ are everywhere tangent to the isocontours and gradient curves of f away from critical points, They are the eigenvectors of the metric induced by S, pulled back to the parameter space $(x, y) \in \mathbb{R}^2$:

$$\begin{aligned}[\mathcal{M}_{\text{ind},\mathbb{R}^2}] = J_S^T[\overline{\mathcal{M}}_{\mathbb{R}^3}]J_S &= \begin{pmatrix} 1 & 0 & f_x \\ 0 & 1 & f_y \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ f_x & f_y \end{pmatrix} = \begin{pmatrix} 1+f_x^2 & f_x f_y \\ f_x f_y & 1+f_y^2 \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{g}^\perp & \boldsymbol{g} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1+f_x^2+f_y^2 \end{pmatrix} \begin{pmatrix} \boldsymbol{g}^\perp & \boldsymbol{g} \end{pmatrix}^T,\end{aligned}$$

and follow the directions of extreme variation of f. For the proposed metric, the sizes are modified so that $\mathcal{M}$ writes:

$$[\mathcal{M}_{\boldsymbol{x}}] = (\boldsymbol{g}^{\perp}(\boldsymbol{x}) \;\; \boldsymbol{g}(\boldsymbol{x})) \begin{pmatrix} h_{\text{iso}}^{-2}(\boldsymbol{x}) & 0 \\ 0 & h_{\text{grad}}^{-2}(\boldsymbol{x}) \end{pmatrix} (\boldsymbol{g}^{\perp}(\boldsymbol{x}) \;\; \boldsymbol{g}(\boldsymbol{x}))^T. \tag{6.5}$$

The gradient is undefined in regions of constant f. In these regions, the mesh should be isotropic and there are no privileged directions. In Chapter 7, we consider a *metric-orthogonal* approach for the generation of curvilinear meshes: it is a frontal algorithm for vertex insertion that aims at aligning the triangles with the eigenvectors of $\mathcal{M}$, see Loseille et al. [42,43]. This is facilitated by keeping smooth eigenvector fields, which is achieved with a simple Laplacian direction smoother. It is adapted from the crossfield and quadrilateral meshing literature, see for instance Beaufort [97], where a pair of orthogonal directions is identified to the angle formed by one of them with the horizontal axis, Fig. 7.5. In quadrilateral meshing, the directions are identical up to a rotation of $\pi/4$ as each of them indistinguishably creates the edge of a rectangle. Here, we distinguish between $\pm\boldsymbol{g}$ and its orthogonal, hence directions are only identical up to a rotation of $\pi/2$. In a region where the derivatives of f are undefined, the directions are averaged according to:

$$\theta(\boldsymbol{x}_i) = \frac{1}{2} \arctan 2 \left(\sum_{\text{adjacent } \boldsymbol{x}_j} \sin 2\theta_j, \sum_{\text{adjacent } \boldsymbol{x}_j} \cos 2\theta_j \right). \tag{6.6}$$

A direction pair $(\boldsymbol{g}, \boldsymbol{g}^{\perp})$ and its opposite $(-\boldsymbol{g}, -\boldsymbol{g}^{\perp})$ yield the same frame of twice their angle, see Fig. 7.5, thus averaging $2\theta_j$ ensures the fields $\boldsymbol{g}$ and $\boldsymbol{g}^{\perp}$ remain identified during the smoothing. This smoothing is performed at vertices $\boldsymbol{x}_i, \boldsymbol{x}_j$ connected by an edge where $|\boldsymbol{g}(\boldsymbol{x}_i) \cdot \boldsymbol{g}(\boldsymbol{x}_j)|$ differs by more than a threshold. In this work, this threshold is set to 0.95.

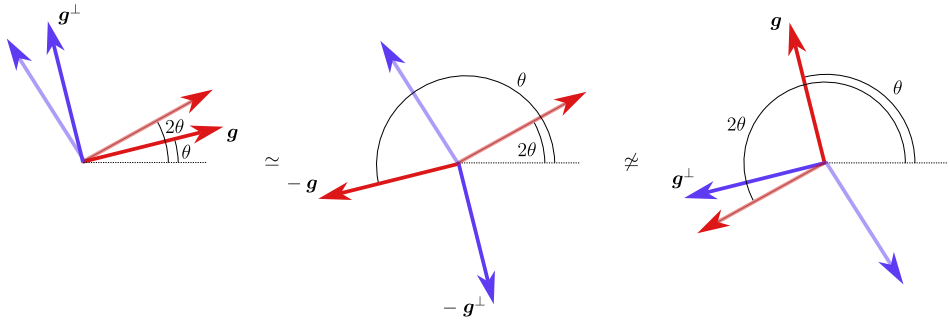


Figure 6.1 The pairs $(\boldsymbol{g}, \boldsymbol{g}^{\perp})$ and $(-\boldsymbol{g}, -\boldsymbol{g}^{\perp})$ differ by a rotation of $\frac{\pi}{2}$ and yield the same frame associated to 2θ.

6.1.2 Quadratic approximation of isoline and gradient

To define the size field, we introduce the isocontours and gradient curves, which are the integral lines of $\boldsymbol{g}^\perp$ and $\boldsymbol{g}$ respectively, and their approximation by quadratic curves. We first recall well-known properties of $\mathcal{C}^2$ curves $\gamma(t) : [a, b] \to \mathbb{R}^2$. Assuming the velocity $\gamma'(t)$ is nonzero everywhere on the curve (that is, the curve is **regular**), the **arc-length** of $\gamma(t)$ is the increasing function $s : [a, b] \to \mathbb{R}^+$:

$$s(t) \triangleq \ell(\gamma|_{[a,t]}) = \int_a^t \|\gamma'(u)\|\, du \geq 0. \tag{6.7}$$

It is the distance traveled along γ from a to t. The **arc-length (re)parameterization** of γ is the composition $\gamma(s(t))$, and is such that $\gamma(s)$ has unit **speed** $\|\gamma'(s)\|$, as we have:

$$\gamma'(s) = \gamma'(t(s))t'(s) = \frac{\gamma'(t(s))}{s'(t)} = \frac{\gamma'(t(s))}{\|\gamma'(t(s))\|} \quad \text{since} \quad s'(t) = \|\gamma'(t)\|. \tag{6.8}$$

Any curve can be reparameterized by arc-length, but it is usually not possible to derive a closed-form expression for $s(t)$, aside from very simple $\|\gamma'(t)\|$. Let now $\gamma(s)$ be a curve parameterized by arc-length. Since $\gamma(s)$ has unit speed, the tangent vector $\mathbf{T}(s) \triangleq \gamma'(s)$ is unit for all s. Similarly, we define the unit normal vector $\mathbf{N}(s) \triangleq \gamma''(s)/\|\gamma''(s)\|$. The norm of the acceleration $\|\gamma''(s)\|$, which is not unit in general, is called the **curvature**[1] κ of γ, and $\gamma'' = \kappa\mathbf{N}$ is called the **curvature vector** at s. The pair $(\mathbf{T}, \mathbf{N})$ forms an orthonormal basis of $\mathbb{R}^2$ everywhere along $\gamma(s)$, called the **Frenet frame** at s. Together with the binormal vector $\mathbf{B} \triangleq \mathbf{T} \times \mathbf{N}$, these vectors also form a right-handed orthonormal basis $(\mathbf{T}, \mathbf{N}, \mathbf{B})$ of $\mathbb{R}^3$, Fig. 6.2. For curves defined on a surface $S = (x, y, f(x, y))$ in $\mathbb{R}^3$, another useful basis, called the **Darboux frame**, is defined from the normal $\boldsymbol{n}$ to the surface at $\gamma(s)$. This normal writes:

$$\boldsymbol{n} \triangleq \frac{S_x \times S_y}{\|S_x \times S_y\|}, \tag{6.9}$$

and the Darboux frame is defined by $(\mathbf{T}, \boldsymbol{n} \times \mathbf{T}, \boldsymbol{n})$. The vector $\boldsymbol{n} \times \mathbf{T}$ lies in the tangent space $T_{\gamma(s)}S$ and is called the **tangent normal** or **geodesic normal**. The normal $\mathbf{N}$ is defined by the curve only, whereas $\boldsymbol{n}$ is defined by the surface in which lies the curve. In particular, the elements of $(\mathbf{T}, \mathbf{N}, \mathbf{B})$ do not form an orthonormal basis of $T_{\gamma(s)}S$ in general, contrary to $(\mathbf{T}, \boldsymbol{n} \times \mathbf{T})$, as the partial derivatives S_x, S_y are a coordinate basis for $T_{\gamma(s)}S$.

In the parameter space (x, y), the curve $\gamma(s)$ has the following quadratic Taylor expansion

[1] With a bit of work, this curvature can be derived from the more abstract definition of curvature introduced in Chapter 3, see e.g. Chapter 8 of Lee [56].

at 0:

$$\gamma(s) = \gamma(0) + s\gamma'(0) + \frac{s^2}{2}\gamma''(0) + \mathcal{O}(s^3). \tag{6.10}$$

Note that although $\gamma(s)$ is parameterized by arc-length, its truncated Taylor expansion is not: it only has unit speed at $s = 0$, as the acceleration $\gamma''(0) = \kappa(0)\mathbf{N}(0)$ is not unit in general. Using the Frenet frame, this expansion also writes:

$$\gamma(s) = \gamma(0) + s\mathbf{T}(0) + \frac{s^2}{2}\kappa(0)\mathbf{N}(0) + \mathcal{O}(s^3). \tag{6.11}$$

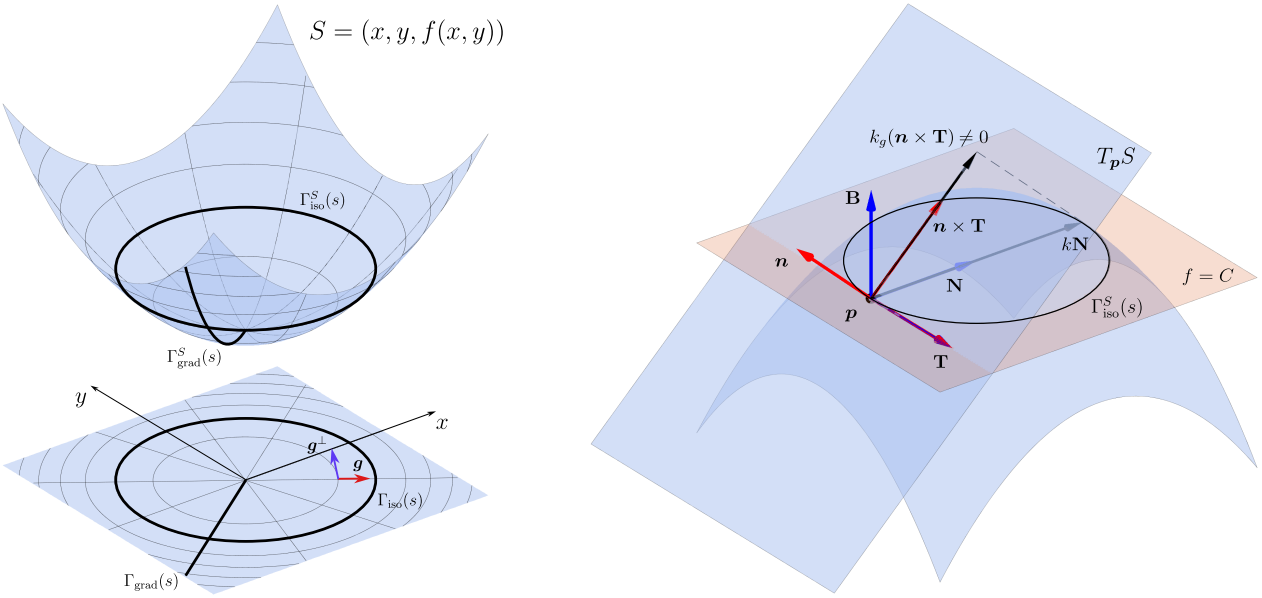


Figure 6.2 The isocontours and gradient curves of $f = x^2 + y^2$, as well as the Frenet and Darboux frames for $\Gamma^S_{\text{iso}}(s)$ lying on the graph of $-f$. The normal $\boldsymbol{n}$ points outwards from S.

At some $\boldsymbol{x}_0 \in \mathbb{R}^2$, the isocontour or level set of f is the curve described by:

$$\Gamma_{\text{iso}}(s) \triangleq \left\{ (x(s), y(s)) \mid f(x(s), y(s)) = f(\boldsymbol{x}_0) = C \right\}, \tag{6.12}$$

and the gradient curves $\Gamma_{\text{grad}}(s)$ of f are curves whose tangent vector is everywhere aligned with ∇f. The isocontours and gradient curves satisfy the initial-value problems:

$$\begin{cases} \Gamma_{\text{iso}}(0) = \boldsymbol{x}_0, \\ \Gamma'_{\text{iso}}(s) = \boldsymbol{g}^\perp(\Gamma_{\text{iso}}(s)) \end{cases}, \qquad \begin{cases} \Gamma_{\text{grad}}(0) = \boldsymbol{x}_0, \\ \Gamma'_{\text{grad}}(s) = \boldsymbol{g}(\Gamma_{\text{grad}}(s)) \end{cases}. \tag{6.13}$$

These curves are unit speed everywhere by definition of $\boldsymbol{g}$ and $\boldsymbol{g}^\perp$. Because the studied metric $\mathcal{M}$ shares the direction of the pullback of the induced metric on S, it is natural

to ask whether the images on S of these curves, noted $\Gamma^S_{\text{iso}}(s) \triangleq (\Gamma_{\text{iso}}(s), f(\Gamma_{\text{iso}}(s)))$ and $\Gamma^S_{\text{grad}}(s) \triangleq (\Gamma_{\text{grad}}(s), f(\Gamma_{\text{grad}}(s)))$, are geodesics for the induced metric on S, i.e., for the ambient Euclidean metric of $\mathbb{R}^3$ restricted to S. The answer is always negative for $\Gamma^S_{\text{iso}}(s)$. Indeed, the curvature vector of a curve on S decomposes in the Darboux frame as ([98], see also Fig. 6.2):

$$\kappa\mathbf{N} = \Big(\underbrace{\kappa\mathbf{N}\cdot(\boldsymbol{n}\times\mathbf{T})}_{\triangleq\kappa_g}\Big)(\boldsymbol{n}\times\mathbf{T}) + \Big(\underbrace{\kappa\mathbf{N}\cdot\boldsymbol{n}}_{\triangleq\kappa_n}\Big)\boldsymbol{n}. \tag{6.14}$$

The projection κ_n is the **normal curvature**, and is the curvature that the curve inherits from the surface on which it lies, whereas the tangential component κ_g is the **geodesic curvature**. It measures how a curve deviates from a straight line on S, and a curve is a geodesic iff its geodesic curvature vanishes, or equivalently if its curvature vector $\kappa\mathbf{N}$ is normal to S. As $\Gamma^S_{\text{iso}}(s)$ lies on a horizontal plane $f = C$, its normal $\mathbf{N}$ can only be aligned with $\boldsymbol{n}$ if the surface is locally vertical, which cannot happen if S is the graph of a function. Thus, $\Gamma^S_{\text{iso}}(s)$ always has a tangential component κ_g and cannot be a geodesic. The images of the gradient curves $\Gamma^S_{\text{grad}}(s)$, on the other hand, may be geodesic. From Drucker et al. [99], these curves are geodesic on the graph of $f(x,y)$ iff f satisfies the PDE $f_x f_y(f_{xx} - f_{yy}) = (f_x^2 - f_y^2)f_{xy}$ along $\Gamma_{\text{grad}}(s) \in \mathbb{R}^2$. This equation is generally not verified for arbitrary f, but it does hold for surfaces of revolution around the z-axis, such as $f = x^2 + y^2$ in Fig. 6.2.

The quadratic approximations of Γ_{iso} and Γ_{grad} match their tangent and curvature at $s = 0$. The curvature of Γ_{iso} is obtained by observing that since $f(\Gamma_{\text{iso}}(s))$ is constant, we can write [100]:

$$\frac{d}{ds}f(\Gamma_{\text{iso}}(s)) = \frac{\partial f}{\partial x^k}\frac{d\Gamma^k_{\text{iso}}}{ds} = \nabla f\cdot\Gamma'_{\text{iso}}(s) = 0, \tag{6.15}$$

thus:

$$\frac{d^2}{ds^2}f(\Gamma_{\text{iso}}(s)) = \Gamma'_{\text{iso}}\cdot H_f\cdot\Gamma'_{\text{iso}} + \nabla f\cdot\Gamma''_{\text{iso}} = 0, \tag{6.16}$$

with H_f the Hessian of $f(x,y)$. Since the acceleration Γ''_{iso} is aligned with ∇f, their dot product is $\nabla f\cdot\Gamma''_{\text{iso}} = \pm\|\nabla f\|\|\Gamma''_{\text{iso}}\|$ with $\|\Gamma''_{\text{iso}}\| = \kappa_{\text{iso}}$. Thus:

$$\kappa_{\text{iso}} = \frac{|\Gamma'_{\text{iso}}\cdot H_f\cdot\Gamma'_{\text{iso}}|}{\|\nabla f\|} = \frac{\left|\boldsymbol{g}^\perp\cdot H_f\cdot\boldsymbol{g}^\perp\right|}{\|\nabla f\|} = \frac{\left|\nabla f^\perp\cdot H_f\cdot\nabla f^\perp\right|}{\|\nabla f\|^3}. \tag{6.17}$$

explicitly, this yields:

$$\kappa_{\text{iso}} = \frac{\left|-f_y^2 f_{xx} + 2f_x f_y f_{xy} - f_x^2 f_{yy}\right|}{(f_x^2+f_y^2)^{3/2}} \quad \text{or, alternatively,} \quad \kappa_{\text{iso}} = \left|\nabla\cdot\left(\frac{\nabla f}{\|\nabla f\|}\right)\right|. \tag{6.18}$$

The curvature of Γ_{grad} is obtained by computing Γ''_{grad} directly. Since we work in the Euclidean space, the acceleration is computed by differentiating the components twice, without the need to account for the variation of basis vectors. It writes:

$$\kappa_{\text{grad}} = \|\Gamma''_{\text{grad}}\| = \frac{\left|\boldsymbol{g}^\perp \cdot H_f \cdot \boldsymbol{g}\right|}{\|\nabla f\|} = \frac{\left|f_x f_y (f_{yy} - f_{xx}) + f_{xy}(f_x^2 - f_y^2)\right|}{(f_x^2 + f_y^2)^{3/2}}. \tag{6.19}$$

When $\kappa_{\text{grad}} = 0$, the PDE above is satisfied and the projections of $\Gamma^S_{\text{grad}}(s)$ are geodesic straight lines in the parameter space. Note that by construction, $\boldsymbol{g} \times \boldsymbol{g}^\perp = \boldsymbol{e}_z$ whereas $\mathbf{T} \times \mathbf{N} = \pm \boldsymbol{e}_z$ depending on the concavity of $\gamma(s)$, and the curvature κ is positive by definition. To work exclusively with the principal directions, we define the **signed curvatures** k_{iso} and k_{grad} so that the curvature vectors of Γ_{iso} and Γ_{grad} write everywhere in terms of $\boldsymbol{g}, \boldsymbol{g}^\perp$:

$$k_{\text{iso}} \boldsymbol{g} \triangleq \kappa_{\text{iso}} \mathbf{N}, \qquad k_{\text{grad}} \boldsymbol{g}^\perp \triangleq \kappa_{\text{grad}} \mathbf{N}. \tag{6.20}$$

This is satisfied by the following expressions, whose derivation is in Appendix A.5.1:

$$k_{\text{iso}} = -\frac{\boldsymbol{g}^\perp \cdot H_f \cdot \boldsymbol{g}^\perp}{\|\nabla f\|}, \qquad k_{\text{grad}} = \frac{\boldsymbol{g}^\perp \cdot H_f \cdot \boldsymbol{g}}{\|\nabla f\|}. \tag{6.21}$$

Then, the curves $\gamma_{\text{iso}}(s)$ and $\gamma_{\text{grad}}(s)$ approximate the isocontours and gradient curves at $\boldsymbol{x}_0$ up to second order iff:

$$\begin{aligned} \gamma_{\text{iso}}(s) &= \boldsymbol{x}_0 + s\boldsymbol{g}^\perp(\boldsymbol{x}_0) + \frac{s^2}{2} k_{\text{iso}}(\boldsymbol{x}_0)\boldsymbol{g}(\boldsymbol{x}_0), \\ \gamma_{\text{grad}}(s) &= \boldsymbol{x}_0 + s\boldsymbol{g}(\boldsymbol{x}_0) \quad + \frac{s^2}{2} k_{\text{grad}}(\boldsymbol{x}_0)\boldsymbol{g}^\perp(\boldsymbol{x}_0). \end{aligned} \tag{6.22}$$

Their velocity and acceleration vectors follow:

$$\begin{aligned} \gamma'_{\text{iso}}(s) &= \boldsymbol{g}^\perp(\boldsymbol{x}_0) + s k_{\text{iso}}(\boldsymbol{x}_0)\boldsymbol{g}(\boldsymbol{x}_0), & \gamma''_{\text{iso}}(s) &= k_{\text{iso}}(\boldsymbol{x}_0)\boldsymbol{g}(\boldsymbol{x}_0), \\ \gamma'_{\text{grad}}(s) &= \boldsymbol{g}(\boldsymbol{x}_0) + s k_{\text{grad}}(\boldsymbol{x}_0)\boldsymbol{g}^\perp(\boldsymbol{x}_0), & \gamma''_{\text{grad}}(s) &= k_{\text{grad}}(\boldsymbol{x}_0)\boldsymbol{g}^\perp(\boldsymbol{x}_0). \end{aligned} \tag{6.23}$$

These parabolas are shown in Fig. 6.3, along with their curvature vector $k_{\text{iso}}\boldsymbol{g}$ or $k_{\text{grad}}\boldsymbol{g}^\perp$. These always point towards the inside of the curve. From $\boldsymbol{x}_0$, one can travel backwards along the parabolas by simply setting $s = -s$, without changing the curvatures or unit vectors.

6.1.3 Size field

The principal sizes are designed to limit the maximum interpolation error along the quadratic approximations γ_{iso} and γ_{grad} to a threshold ϵ. We propose to improve on the pointwise

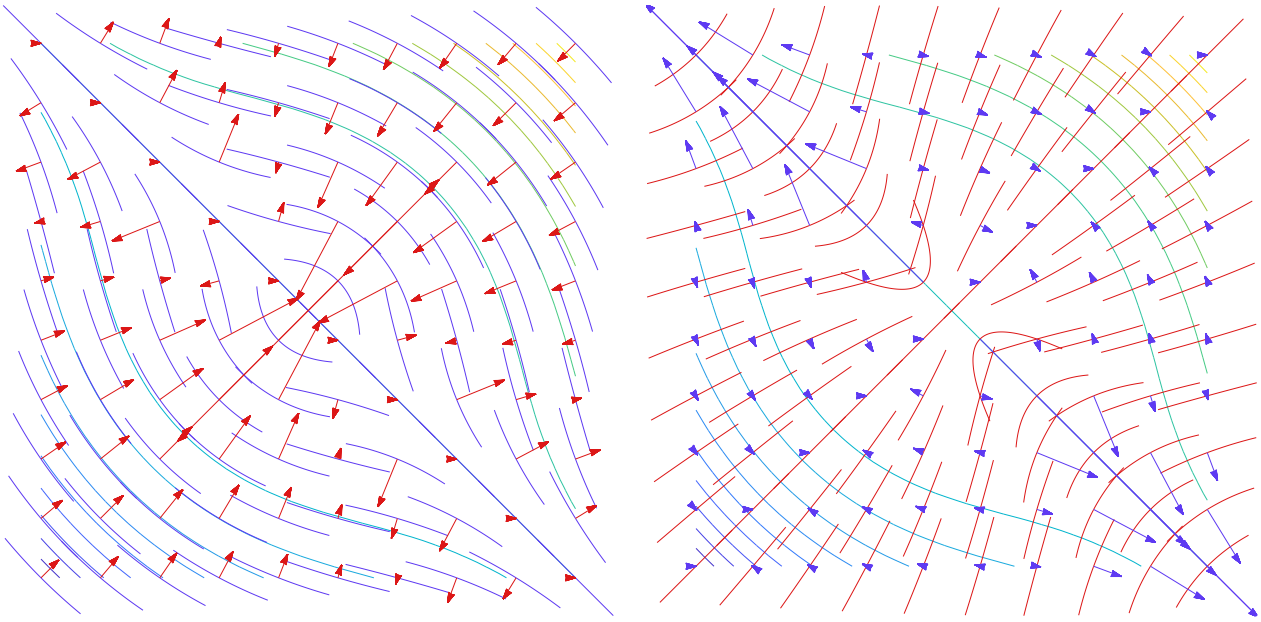


Figure 6.3 Quadratic approximations of the isocontours $\gamma_{\text{iso}}(s)$ (left) and of the gradient curves $\gamma_{\text{grad}}(s)$ (right) with their curvature vector for $f = x^3 + x^2y + xy^2 + y^3$. The true isocontours of f are traced on both figures according to their colormap. On the left, the quadratic isocontours are in blue and the signed curvature vector is a red arrow. On the right, the quadratic gradient curve is in red and the signed curvature vector is a blue arrow. The signed curvatures make the curvature vector always point in the direction of $\mathbf{N}$.

estimator proposed in Zhang et al. [22] and write instead the integral remainder of the interpolation error $e_f^k(s) = f(\gamma(s)) - \Pi_f^k(\gamma(s))$ on a curve of length $\mathcal{L}$ as:

$$|e_{\max}(\gamma)| = |e_f^k(\mathcal{L})| = \frac{1}{k!}\left|\int_0^{\mathcal{L}} (\mathcal{L}-s)^k \frac{d^{k+1} f(\gamma(s))}{ds^{k+1}}\, ds\right| \leq \epsilon. \tag{6.24}$$

The derivatives of the composition $f \circ \gamma$ are obtained by the chain rule, yielding Faà di Bruno's formula. Letting $\dot{\gamma} = \gamma', \ddot{\gamma} = \gamma''$, $G = \nabla f, H = H_f$ and C_{ijk} denote the components of the third derivatives tensor of f, we have in particular for linear and quadratic interpolation on quadratic curves:

$$\begin{aligned}
\frac{d^2 f \circ \gamma}{ds^2}(s) &= H_{ij}(\gamma(s))\dot{\gamma}^i(s)\dot{\gamma}^j(s) + G_i(\gamma(s))\ddot{\gamma}^i(s), \\
\frac{d^3 f \circ \gamma}{ds^3}(s) &= C_{ijk}(\gamma(s))\dot{\gamma}^i(s)\dot{\gamma}^j(s)\dot{\gamma}^k(s) + 3H_{ij}(\gamma(s))\dot{\gamma}^i(s)\ddot{\gamma}^j(s).
\end{aligned} \tag{6.25}$$

The third derivatives of γ always vanish as the curve is quadratic. These error estimates show the main idea behind curvilinear mesh adaptation: if $\gamma(s)$ is well represented by a curved edge, then the acceleration $\ddot{\gamma}$ does not vanish and the linear (resp. quadratic) interpolation

error is not zero even if the function f is linear (resp. quadratic) in $\boldsymbol{x}$. However, this allows for cancellation between the errors accumulated along the straight and curved parts of γ, with the sums above potentially smaller than for straight edges. The approximations $\gamma_{\text{iso}}, \gamma_{\text{grad}}$ were precisely arranged for such error reduction, as shown in the example for $f = x^2 + y^2$ further.

Suppose we want to limit the linear interpolation error to ϵ along the quadratic curve $\gamma_{\text{iso}}(s)$ given by (6.22). We write:

$$\begin{aligned} |e_{\max}(\gamma_{\text{iso}})| = |e_f^1(\mathcal{L})| &= \left| \int_0^{\mathcal{L}} (\mathcal{L} - s) \frac{d^2 f(\gamma_{\text{iso}}(s))}{ds^2} \, ds \right| \\ &= \left| \int_0^{\mathcal{L}} (\mathcal{L} - s) \left(\gamma'_{\text{iso}} \cdot H_f \cdot \gamma'_{\text{iso}} + G \cdot \gamma''_{\text{iso}} \right) \, ds \right| \\ &= \left| \int_0^{\mathcal{L}} (\mathcal{L} - s) \left[(\boldsymbol{g}^\perp + s k_{\text{iso}} \boldsymbol{g}) \cdot H_f \cdot (\boldsymbol{g}^\perp + s k_{\text{iso}} \boldsymbol{g}) + G \cdot k_{\text{iso}} \boldsymbol{g} \right] \, ds \right| \\ &= \left| c_4 \mathcal{L}^4 + c_3 \mathcal{L}^3 + c_2 \mathcal{L}^2 \right| \leq \epsilon \end{aligned} \tag{6.26}$$

where the coefficients c_i defined as follows are evaluated at $\boldsymbol{x}_0$:

$$c_4 = \frac{k_{\text{iso}}^2}{12} \boldsymbol{g} \cdot H_f \cdot \boldsymbol{g}, \quad c_3 = \frac{k_{\text{iso}}}{3} \boldsymbol{g} \cdot H_f \cdot \boldsymbol{g}^\perp, \quad c_2 = \frac{\boldsymbol{g}^\perp \cdot H_f \cdot \boldsymbol{g}^\perp + G \cdot k_{\text{iso}} \boldsymbol{g}}{2}. \tag{6.27}$$

To limit the interpolation error to ϵ along $\gamma_{\text{iso}}(s)$, the optimal size $\mathcal{L}$ should be the smallest real root of the polynomials:

$$p_1(\mathcal{L}) = c_4 \mathcal{L}^4 + c_3 \mathcal{L}^3 + c_2 \mathcal{L}^2 \pm \epsilon = 0. \tag{6.28}$$

These roots can be computed explicitly from a closed-form solution, or evaluated numerically with a root finder. The optimal size along $\gamma_{\text{grad}}(s)$ is obtained in the same way, substituting $\boldsymbol{g}^\perp$ for $\boldsymbol{g}$ and k_{grad} for k_{iso}, and the Riemannian metric (6.5) is defined by setting $h_{\text{iso}} = \mathcal{L}_{\text{iso}}$ and $h_{\text{grad}} = \mathcal{L}_{\text{grad}}$. Quasi-unit linear (straight) meshes for this metric are shown in Fig. 6.4 for analytic fields and compared with meshes adapted to the optimal metric $\mathcal{M}^{\text{opt}}$ for adaptation with linear triangles. Note that even though $\mathcal{M}$ is tailored for curved elements, the adapted meshes are straight and are shown to give an idea of the behaviour of the metric, as the generation of curved meshes is the topic of Chapter 7. The field $x^2 + y^2$ is isotropic and yields an isotropic metric $\mathcal{M}^{\text{opt}}$ and mesh. Similarly, $x^8 + y^8$ yields a mostly isotropic mesh. In comparison, the isocontours metric $\mathcal{M}$ stretches the elements along the isocontours of f up to the prescribed error threshold. In particular, the mesh density has decreased in the corners of $x^8 + y^8$, where the field makes a quarter turn.

For quadratic interpolation, the error estimate writes:

$$|e_{\max}(\gamma)| = |e_f^2(\mathcal{L})| = \frac{1}{2}\left|\int_0^{\mathcal{L}} (\mathcal{L}-s)^2 \left(C_{ijk}\dot{\gamma}^i\dot{\gamma}^j\dot{\gamma}^k + 3\dot{\gamma}\cdot H_f\cdot\ddot{\gamma}\right)\, ds\right| \leq \epsilon, \tag{6.29}$$

and the optimal lengths are the smallest real roots of the degree 6 polynomials:

$$p_2(\mathcal{L}) = c_6\mathcal{L}^6 + c_5\mathcal{L}^5 + c_4\mathcal{L}^4 + c_3\mathcal{L}^3 \pm \epsilon = 0, \tag{6.30}$$

whose coefficients for e.g. $\gamma_{\text{iso}}(s)$ are given by:

$$\begin{aligned} c_6 &= a_6 k_{\text{iso}}^3 C_{ijk} g^i g^j g^k, \qquad & c_4 &= a_4 k_{\text{iso}} C_{ijk} g^i g^{\perp,j} g^{\perp,k} + \frac{k_{\text{iso}}^2}{8}\boldsymbol{g}\cdot H_f\cdot\boldsymbol{g}, \\ c_5 &= a_5 k_{\text{iso}}^2 C_{ijk} g^i g^j g^{\perp,k}, & c_3 &= \frac{1}{6} C_{ijk} g^{\perp,i} g^{\perp,j} g^{\perp,k} \quad + \frac{k_{\text{iso}}}{2}\boldsymbol{g}\cdot H_f\cdot\boldsymbol{g}^{\perp}, \end{aligned} \tag{6.31}$$

with $a_6 = 1/120, a_5 = 1/20$ and $a_4 = 1/8$. Since the third derivatives are available, the Hessian can be improved by considering its linear expansion $H_{ij}(\gamma(s)) = H_{ij}(\boldsymbol{x}_0) + sC_{ijk}\dot{\gamma}^k + \mathcal{O}(s^2)$ along the curve, yielding instead $a_6 = 1/30, a_5 = 3/20$ and $a_4 = 1/4$.

A metric tailored for curved elements should improve on the available metrics for linear elements, in the sense that it should not prescribe curvature or increase the mesh density if it is not required by the solution. For the particular case of $f = x^2 + y^2$, the field is perfectly captured on linear triangles using reference-space interpolation: the optimal mesh elements are straight, isotropic and arbitrarily large. This is no longer true with $\mathcal{P}^2$ elements, as quadratic reference-space interpolation does not preserve quadratic polynomials on curvilinear geometries. For this f, the gradient curves are (geodesic radial) straight lines: the absence of both curvature and third derivatives of f leads to an arbitrarily large size along γ_{grad} as expected, but not along γ_{iso}, as detailed further, so the metric does not reflect the fact that the ideal mesh is large along any choice of orthogonal directions. This can be avoided by evaluating the error estimate (6.29) for both curved ($k \neq 0$) and straight ($k = 0$) parameterizations and taking the largest of both sizes, leading to the metric:

$$[\mathcal{M}_{\boldsymbol{x}}] = \begin{pmatrix} \boldsymbol{g}^{\perp}(\boldsymbol{x}) & \boldsymbol{g}(\boldsymbol{x}) \end{pmatrix} \begin{pmatrix} \max\limits_{k=0,\,k_{\text{iso}}} \mathcal{L}_{\text{iso}}^{-2}(\boldsymbol{x},k) & 0 \\ 0 & \max\limits_{k=0,\,k_{\text{grad}}} \mathcal{L}_{\text{grad}}^{-2}(\boldsymbol{x},k) \end{pmatrix} \begin{pmatrix} \boldsymbol{g}^{\perp}(\boldsymbol{x}) & \boldsymbol{g}(\boldsymbol{x}) \end{pmatrix}^T. \tag{6.32}$$

This metric yields an isotropic mesh whenever f is captured by linear elements. Adapted straight meshes for this isocontours metric and quadratic interpolation are shown in Fig. 6.5 for the same analytic fields as Fig. 6.4. With the modification above, both $\mathcal{M}^{\text{opt}}$ and $\mathcal{M}$

predict an isotropic mesh for quadratic functions whose size is bounded by a user-defined maximum mesh size.

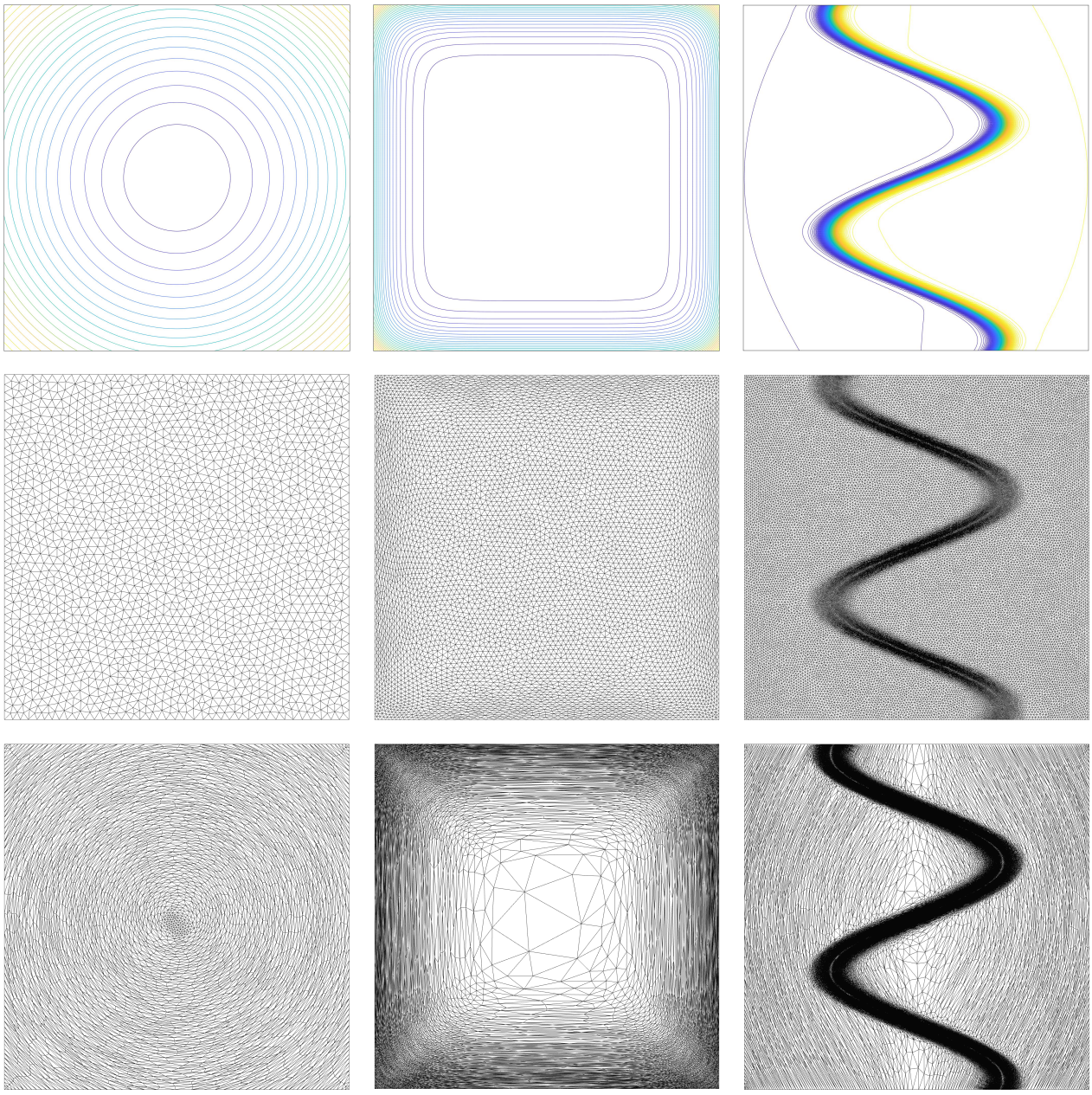


Figure 6.4 Illustration of the isoline metric for linear interpolation. Linear meshes adapted for $\mathcal{M}$ for $f = x^p + y^p$ for $p = 2$ (left) and $p = 8$ (center) and for $f = x^3 + x^2 y + \tanh[10\,(2x - \sin 5y)]$ (right). The metric is computed for linear interpolation on quadratic triangles.

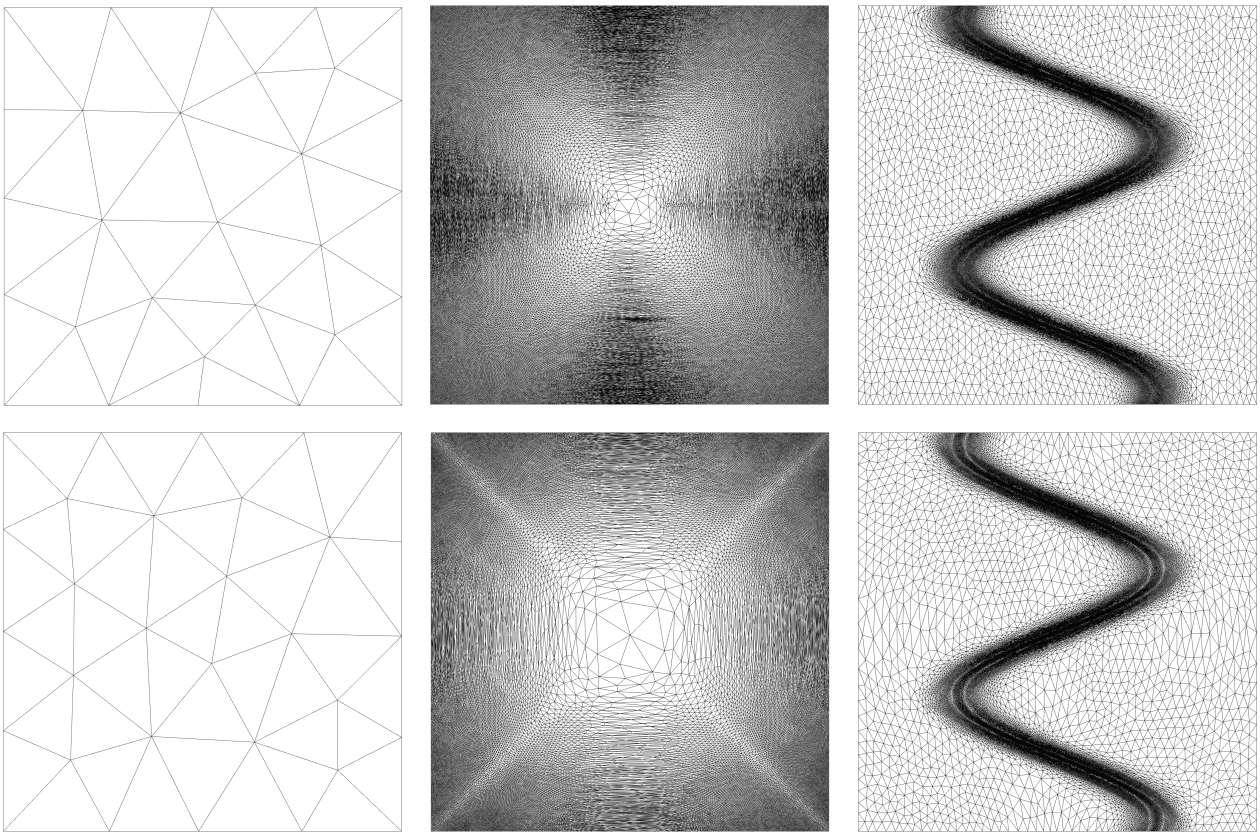


Figure 6.5 Illustration of the isoline metric for quadratic interpolation.

Example: $f(x, y) = x^2 + y^2$

To show the rationale behind the proposed metric, we consider an adaptation problem for the function $f(x, y) = x^2 + y^2 = r^2$. We first consider linear interpolation. In the standard framework of linear anisotropic mesh adaptation, using for instance the optimal Riemannian metric minimizing the L^p norm of the error from Chapter 5, the error model is controlled by the Hessian $H_f = \mathrm{diag}(2, 2)$. As H_f is constant and isotropic, this yields a fully isotropic mesh with elements of the same size. The function f is constant on the curves $r = C$, i.e., it is anisotropic when written in polar coordinates (r, θ). If we allow quadratic triangles to bend along those curves, then f is captured exactly on the edges by linear (and even constant) interpolation. The curvatures at $\boldsymbol{x}_0$ are computed from (6.18) and (6.19) and write:

$$k_{\mathrm{iso}} = -\frac{1}{\sqrt{x^2 + y^2}} = -\frac{1}{r}, \quad k_{\mathrm{grad}} = 0, \tag{6.33}$$

since the integral curves are circles and straight lines in the parameter space. Let $\boldsymbol{x} = (r, 0)$ to ease the computations, but any other point can be used because of the radial symmetry

of f. We have $G = \nabla f(\boldsymbol{x}) = (2r, 0)^T$ thus $\boldsymbol{g} = (1, 0)^T$ and the parametric curves are $\gamma_{\text{iso}}(s) = (r - s^2/2r, s)^T, \gamma_{\text{grad}}(s) = (r + s, 0)^T$. Along $\gamma_{\text{iso}}(s)$, the target function is:

$$f(\gamma_{\text{iso}}(s)) = x_{\text{iso}}(s)^2 + y_{\text{iso}}(s)^2 = \left(r - \frac{s^2}{2r}\right)^2 + s^2 = r^2 + \frac{s^4}{4r^2}, \tag{6.34}$$

that is, f is constant up to an $\mathcal{O}(s^4)$ remainder. If the constant part is captured by the interpolant $\Pi^1 f(\gamma_{\text{iso}}(s))$, then the exact pointwise error is $e^1_f(s) = s^4/4r^2 \leq \epsilon$, and the maximum error on $\gamma_{\text{iso}}(s)$ can be controlled by limiting its length to $s \simeq \mathcal{L}_{\text{iso}} \leq \sqrt[4]{4r^2\epsilon}$. Along the curve $\gamma_{\text{grad}}(s)$, we have $f = (r + s)^2 = r^2 + s^2 + 2rs$. If the linear part $r^2 + 2rs$ is captured by $\Pi^1 f(\gamma_{\text{grad}}(s))$, the exact pointwise error is $e^1_f(s) = s^2 \leq \epsilon$, which is independent of the position r and yields the optimal size $s \simeq \mathcal{L}_{\text{grad}} \leq \sqrt{\epsilon}$. In comparison, the proposed error estimate (6.26) leads to:

$$\begin{aligned} p_1(\mathcal{L}_{\text{iso}}) = \frac{\mathcal{L}^4_{\text{iso}}}{6r^2} - \epsilon = 0 \quad &\longrightarrow \quad \mathcal{L}_{\text{iso}} = \sqrt[4]{6r^2\epsilon}, \\ p_1(\mathcal{L}_{\text{grad}}) = \mathcal{L}^2_{\text{grad}} - \epsilon = 0 \quad &\longrightarrow \quad \mathcal{L}_{\text{grad}} = \sqrt{\epsilon}, \end{aligned} \tag{6.35}$$

thus an overestimation of the lengths of $\mathcal{L}_{\text{iso,model}}/\mathcal{L}_{\text{iso,exact}} = \sqrt[4]{6/4} \simeq 1.1$ along γ_{iso} and identical lengths along γ_{grad}. The ratio of anisotropy is $\mathcal{L}_{\text{iso}}/\mathcal{L}_{\text{grad}} = \sqrt[4]{6r^2/\epsilon}$, and increases with the distance from the origin. In comparison, the error estimate along a straight parameterization $\gamma_{\text{iso}} = \boldsymbol{x} + s\boldsymbol{g}^\perp$ yields $\frac{1}{2}\left|H_{f,ij} g^{\perp,i} g^{\perp,j} \mathcal{L}^2_{\text{iso}}\right| = \frac{1}{2}\left|2\mathcal{L}^2_{\text{iso}}\right| = \epsilon$, thus a constant isotropic size $\mathcal{L}_{\text{iso}} = \sqrt{\epsilon} = \mathcal{L}_{\text{grad}}$ and $h_{\text{iso}} = \max(\sqrt[4]{6r^2\epsilon}, \sqrt{\epsilon})$. Thus, h_{iso} is constant and equal to $\sqrt{\epsilon}$ close to the origin, then increases to $\sqrt[4]{6r^2\epsilon}$.

For quadratic interpolation, the error estimate yields:

$$\begin{aligned} p_2(\mathcal{L}_{\text{iso}}) = \frac{\mathcal{L}^4_{\text{iso}}}{2r^2} - \epsilon = 0 \quad &\longrightarrow \quad \mathcal{L}_{\text{iso}} = \sqrt[4]{2r^2\epsilon}, \\ p_2(\mathcal{L}_{\text{grad}}) = \quad 0 \quad - \epsilon = 0 \quad &\longrightarrow \quad \mathcal{L}_{\text{grad}} \text{ is arbitrary.} \end{aligned} \tag{6.36}$$

The estimate along a straight γ_{iso} predicts $\frac{1}{6}\left|C_{ijk} g^{\perp,i} g^{\perp,j} g^{\perp,k} \mathcal{L}^3_{\text{iso}}\right| = 0 \times \mathcal{L}^3_{\text{iso}} = \epsilon$, thus an arbitrary length as for $\mathcal{L}_{\text{grad}}$. With the correction (6.32), we obtain $h_{\text{grad}} = \max(\sqrt[4]{2r^2\epsilon}, h_{\max})$ and an isotropic mesh as expected.

6.1.4 The principal directions are not optimal

Here, we take a step back and discuss the limitations of the isocontours metric. In the context of the analysis of Chapter 4, the error estimate for the isocontours metric is written

in the physical space. Physical space estimates for unit elements require a priori knowledge of the element shape. Here, this translates in the necessity of choosing *beforehand* a set of curved parameterizations $\gamma_{\text{iso}}, \gamma_{\text{grad}}$, along which the principal sizes of the metric are defined. Although the principal directions $\boldsymbol{g}^\perp, \boldsymbol{g}$ were chosen from a reasonable observation, we show here that they do not enforce $|e_{\max}| \leq \epsilon$ along all intermediary directions around some $\boldsymbol{x}_0$, even in the absence of curvature. This case is easier to treat, as the error estimate reduces to the usual Taylor estimates along the straight line $\gamma(s) = \boldsymbol{x}_0 + s(\boldsymbol{p} - \boldsymbol{x}_0)$ with $\dot{\gamma} = \boldsymbol{p} - \boldsymbol{x}_0 \triangleq \mathcal{L}\boldsymbol{u}$ and $\|\boldsymbol{u}\| = 1$:

$$\frac{1}{2}\left|H_{ij}u^i u^j \mathcal{L}^2\right| \leq \epsilon \quad \text{and} \quad \frac{1}{6}\left|C_{ijk}u^i u^j u^k \mathcal{L}^3\right| \leq \epsilon. \tag{6.37}$$

This either is a quadratic form in $\mathcal{Q} = |H_f|$ for linear interpolation or can be bounded by a quadratic form in $\mathcal{Q}$ for higher-order polynomials, thus it is a quadratic form in $\mathcal{Q}$ in general. Note that this is only true in the absence of curvature: (6.24) cannot be identified to a quadratic form even for linear interpolation, because the curvature affects the error accumulated by travelling along $\gamma(s)$, so that the error depends on the path travelled and not only on the endpoint. The minimum and maximum values of a quadratic form are reached along the eigenvectors of its matrix, here $\mathcal{Q}$, and those eigenvectors differ from the directions $\boldsymbol{g}^\perp, \boldsymbol{g}$ of the gradient in general. The estimates above are guaranteed to hold along all directions if the principal directions are those of $\mathcal{Q}$, whereas for the proposed metric, they are only enforced along $\boldsymbol{g}, \boldsymbol{g}^\perp$, thus an error higher than the target ϵ is reached along a direction between $\boldsymbol{g}^\perp$ and $\boldsymbol{g}$. This is shown in Fig. 6.6 for $f = x^2y + xy^2 + y^3$: the blue ellipse is associated to the scaled Hessian $|H_f|/2\epsilon$, with sizes $\mathcal{L}_k = \sqrt{2\epsilon/|\lambda_{H,k}|}$ so that $\frac{1}{2}\left|H_{ij}v^i_{H,k}v^j_{H,k}\mathcal{L}_k^2\right| = \epsilon$ along each unit eigenvector $v_{H,k}$, whereas the red ellipse is the unit (tangent) sphere of $\mathcal{M}$. The error model is bounded by ϵ for all intermediary straight lines $\gamma(s) = \mathcal{L}\boldsymbol{u}$ whose extremity lies on the ellipse associated to $|H_f|/2\epsilon$, whose principal directions are those of H_f. This is not the case for the intermediary straight lines interpolated between $\mathcal{L}_{\text{iso}}\boldsymbol{g}^\perp$ and $\mathcal{L}_{\text{grad}}\boldsymbol{g}$: the maximum error is 0.12562 for a target of $\epsilon = 0.1$, thus a 25% overshoot.

When curvature is involved, we consider once again quadratic interpolation on $f = x^2 + y^2$ and the family of curves defined for $\theta \in [0, \pi]$ by:

$$\gamma_\theta(s) = \boldsymbol{x}_0 + s\boldsymbol{v}(\theta) + \frac{s^2}{2}k(\theta)(\boldsymbol{v}(\theta))^\perp, \tag{6.38}$$

whose curvature and tangent vector are interpolated from those of γ_{iso} and γ_{grad} according to:

$$\boldsymbol{v}(\theta) = \boldsymbol{g}\cos\theta + \boldsymbol{g}^\perp \sin\theta, \qquad k(\theta) = k_{\text{grad}}\cos^2\theta + k_{\text{iso}}\sin^2\theta. \tag{6.39}$$

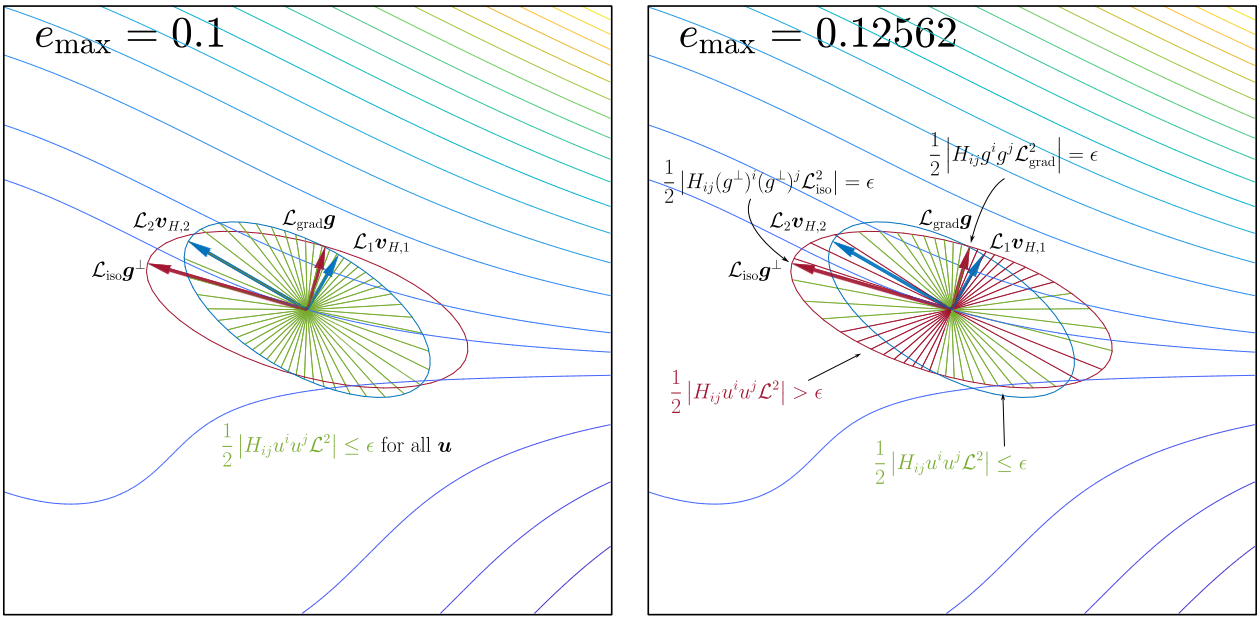


Figure 6.6 Error model evaluated at $\boldsymbol{x}_0$ along straight parameterizations $\gamma(s)$, with $\dot{\gamma} = \boldsymbol{p} - \boldsymbol{x}_0 = \mathcal{L}\boldsymbol{u}$ and $\|\boldsymbol{u}\| = 1$. Unit tangential sphere of $|H_f|/2\epsilon$ (blue) and of the isocontours metric $\mathcal{M}$ (red). On the left, the error on all intermediary straight lines is bounded by the target error $\epsilon = 0.1$. On the right, the error may exceed ϵ even though it is bounded along the principal directions of the gradient.

These curves sweep the plane between γ_{iso} and γ_{grad} with a smooth variation in their tangent and curvature, Fig 6.7. Their Euclidean length is set so that they have unit length with respect to the metric (6.5) without the size correction (6.32), with $\mathcal{L}_{\text{iso}} = (2r^2\epsilon)^{1/4}$, $\mathcal{L}_{\text{grad}}$ arbitrary and a target error $\epsilon = 10^{-3}$. Note that their extremity does not lie on the unit sphere of $\mathcal{M}$ as these curves are not geodesics in general. Rather, they are unit length interpolations of γ_{iso} and γ_{grad}. On each curve, we evaluate the error model (6.24) for quadratic interpolation with $C_{ijk} = 0$. As γ_{grad} has no curvature, f is captured exactly and the error is zero. When $\mathcal{L}_{\text{grad}}$ is small, $e_{\max} = \epsilon$ is reached along the principal direction γ_{iso}, see the green curve in Fig 6.7, so that the error along each intermediary curve is controlled by ϵ. This is no longer true when $\mathcal{L}_{\text{grad}}$ increases, and $e_{\max} > \epsilon$ can be reached on intermediary curves. The errors for $\mathcal{L}_{\text{grad}} = 0.7$ are shown in Fig 6.7: the maximum error is reached between the principal directions and is about 10% higher than the prescribed ϵ. This behaviour is amplified further as $\mathcal{L}_{\text{grad}}$ increases.

To summarize, the isocontour metric cannot bound the error to the target ϵ in the vicinity of $\boldsymbol{x}_0$. This comes from the fact that (6.24) is defined in the physical space on parameterized curves that are set beforehand, and the computations of the directions and sizes are decou-

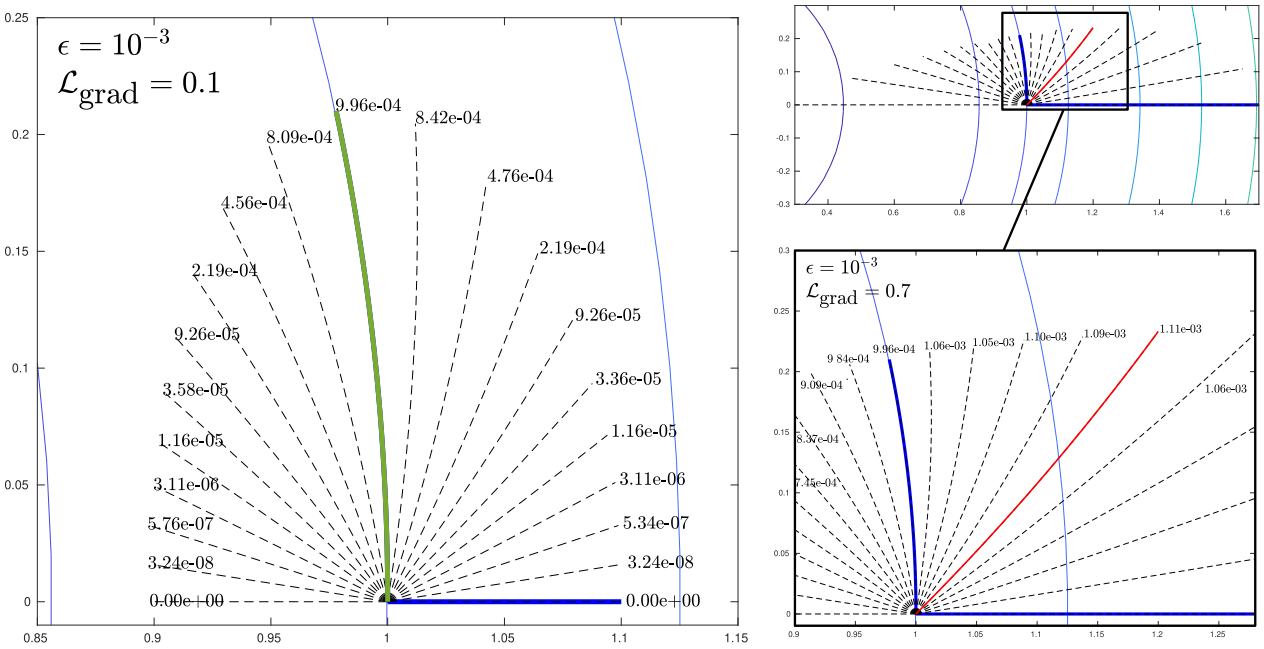


Figure 6.7 Quadratic approximations of the gradient γ_{grad} and of the isocontour γ_{iso} at $\boldsymbol{x}_0 = (1, 0)$ (in thick blue). Left: the maximum of the error model is the target ϵ and is reached on γ_{iso}. The error on the intermediary curves are below ϵ. Right: as $\mathcal{L}_{\text{grad}}$ increases, the maximum error exceeds ϵ and is reached on an arbitrary curve.

pled. In reality, these parameterizations are dictated by the metric and its derivatives, and should be properly solved for together with the sizes. To ensure a control of the error in each direction around $\boldsymbol{x}_0$, we proposed in Bawin et al. [41] a first extension of the log-simplex method, where an ellipse is fitted inside the level set $\Lambda(e_f^k) \equiv |e_f^k(\mathcal{L})| = 1$. Each point of this level set is obtained by solving the polynomials (6.26) or (6.29) for a target error $\epsilon = 1$. A major obstacle is that e_f^k is no longer a homogeneous polynomial, thus Proposition 5.1 no longer holds and we lose the guarantee that the metric associated to the ellipse inscribed in $\Lambda(e_f^k)$ remains an upper bound for the level sets $e_f^k = C \neq 1$. In Section 6.2, we propose to improve on this method by considering the error model in the reference space presented in Chapter 4, which is a quadratic form. As this model involves the metric derivatives, the upper bound $\mathcal{Q}$ satisfies a system of quasilinear PDE in each of its components. The metric derivatives are solved for together with the metric, which means that the parameterized curves given to (6.24) become a part of the problem, and not set beforehand anymore. Before that, however, we investigate the asymptotic behaviour of the isocontour metric. Nonetheless, we observe in the next section that although it does not bound the error in all directions, the error may remain controlled by the density parameter ϵ.

6.1.5 Asymptotic convergence of the isocontour metric

One way of assessing the asymptotic behaviour of a given Riemannian metric is to compute the convergence rate of a target error on its associated quasi-unit meshes. To this end, quasi-unit meshes should be reliably generated, as the error should measure the adequacy between the metric and the solution, and not the adequacy between the meshing process and the metric. This assumes that both unitness and its practical implementation are well-defined, so that one can rule out the error from mesh generation per se. In the context of curvilinear mesh adaptation, a definition of quasi-unit has yet to be widely accepted, although we gave convincing arguments in favor of Riemannian (quasi-)isometries in Chapter 4. Moreover, an implementation of quasi-unit $\mathcal{P}^2$ triangulations was not yet discussed and is the topic of Chapter 7.

Alternatively, the continuous mesh framework provides local continuous error estimates $e(\mathcal{M})$ allowing to perform convergence studies based solely on parameterized families of Riemannian metrics, see e.g. Loseille [1, 9]. The complexity $\mathcal{C}(\mathcal{M})$ is analogous to the number of vertices N_v in a unit mesh, so that the convergence rate k can be derived from a relation $||e(\mathcal{M})|| = \mathcal{O}(\mathcal{C}^k)$. This is done in [1, 9] for the continuous error model for linear interpolation on linear triangles:

$$e(\mathcal{M}) = \frac{1}{8} \operatorname{tr}\left(\mathcal{M}^{-1/2}|H_f|\mathcal{M}^{-1/2}\right). \tag{6.40}$$

We propose to assess the asymptotic behaviour of the metric $\mathcal{M}_{\text{iso}}$ given by (6.5) on the error models for both linear and quadratic triangles, where the latter is given by (4.100) and writes:

$$e(\mathcal{M}) = C\left|\operatorname{tr}\left(\mathcal{M}^{-1/2}H_f\mathcal{M}^{-1/2}\right) + (\partial\mathcal{M}^{-1/2} : \mathcal{M}^{-1/2}) \cdot \nabla f\right|. \tag{6.41}$$

The constant C is set to $1/8$ to be consistent with the linear model in the absence of curvature. As was done in [9], we compute the L^1 norm[2] of both error models against the square root of the metric complexity and deduce the convergence rate of the error on $\mathcal{M}_{\text{iso}}$ and its unit meshes. This analysis is performed for linear interpolation. To put the metric $\mathcal{M}_{\text{iso}}$ in perspective, it is compared with the following other metrics:

- The optimal metric $\mathcal{M}^{\text{opt}} = \mathcal{M}^{\text{opt}}_{2,0,1}$ for anisotropic adaptation on linear triangles given by (5.48) and minimizing the L^1 norm of the interpolation error.
- For $f = x^2 + y^2$ only, the metric $\mathcal{M}_{1,\alpha}$ proposed in [9] and used in Sanjaya et al. [29]

[2]The continuous error model has the simple form (6.40) which can be integrated to derive exact L^1 error estimates in terms of the metric complexity, as done in [9]. Moreover, the isocontours metric controls the maximum error along γ_{iso} and γ_{grad}, but does so throughout the domain, making the L^1 norm a somewhat natural comparison.

for $\mathcal{P}^2$ adaptation. It is given for a density parameter $\alpha > 0$ by:

$$\mathcal{M}_{1,\alpha}(x,y) = P \begin{pmatrix} \alpha^2\, h_1^{-2}(x,y) & 0 \\ 0 & \alpha\, h_2^{-2}(x,y) \end{pmatrix} P^T, \text{ with } P = \frac{1}{\sqrt{x^2+y^2}} \begin{pmatrix} x & -y \\ y & x \end{pmatrix} \quad (6.42)$$

and

$$h_1^{-2}(x,y) = 4(x^2+y^2), \qquad h_2^{-2}(x,y) = \frac{1}{2\sqrt{x^2+y^2}}. \quad (6.43)$$

With our notation, this metric also writes after inverting the eigenvectors:

$$\mathcal{M}_{1,\alpha}(x,y) = (\boldsymbol{g}^\perp \;\; \boldsymbol{g}) \begin{pmatrix} \alpha \left[8(x^2+y^2)^{3/2}\right]^{-1} & 0 \\ 0 & \alpha^2 \end{pmatrix} (\boldsymbol{g}^\perp \;\; \boldsymbol{g})^T. \quad (6.44)$$

Letting $h_{\text{iso}}^{-2} = \alpha h_2^{-2}$ and $h_{\text{grad}}^{-2} = \alpha^2 h_1^{-2}$, the anisotropy $\text{An} \triangleq h_{\text{iso}}/h_{\text{grad}}$ scales as $\text{An}(\mathcal{M}_{1,\alpha}) \sim r^{3/2}\sqrt{\alpha}$. In comparison, setting $\epsilon \sim \alpha^{-2}$, the anisotropy of $\mathcal{M}$ scales as $\text{An}(\mathcal{M}) = (6r^2\alpha^2)^{1/4} \sim \sqrt{r\alpha}$, thus slower as the radius increases.

- The induced metric $\mathcal{M}_{\text{ind},1,\alpha}$ whose principal sizes are scaled nonuniformly as $\mathcal{M}_{1,\alpha}$:

$$\mathcal{M}_{\text{ind},1,\alpha}(x,y) = (\boldsymbol{g}^\perp \;\; \boldsymbol{g}) \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^2(1+4x^2+4y^2)) \end{pmatrix} (\boldsymbol{g}^\perp \;\; \boldsymbol{g})^T. \quad (6.45)$$

  Its anisotropy scales as $\text{An}(\mathcal{M}_{\text{ind},1,\alpha}) = \sqrt{1+4r^2}\sqrt{\alpha}$, thus $\sim r\sqrt{\alpha}$ as r becomes large.

- The induced metric $\mathcal{M}_{\text{ind},2,\alpha}$ with sizes independent of the position:

$$\mathcal{M}_{\text{ind},2,\alpha}(x,y) = (\boldsymbol{g}^\perp \;\; \boldsymbol{g}) \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^2 \end{pmatrix} (\boldsymbol{g}^\perp \;\; \boldsymbol{g})^T. \quad (6.46)$$

  Its anisotropy is constant and simply scales as $\text{An}(\mathcal{M}_{\text{ind},2,\alpha}) = \sqrt{\alpha}$.

The metrics $\mathcal{M}_{\text{iso}}, \mathcal{M}_{1,\alpha}, \mathcal{M}_{\text{ind},1,\alpha}$ and $\mathcal{M}_{\text{ind},2,\alpha}$ are all derived from the induced metric of f and differ on how their anisotropy scales with the position. Note that setting the same exponents for α in the metrics above yields a unique scaling coefficient for both sizes, resulting in embedded metrics whose linear continuous error always converges at order 2. The optimal metric $\mathcal{M}^{\text{opt}}$ is the gold standard, as it minimizes the interpolation error in L^1 norm and converges at order 2 as the complexity increases [1]. Thus, other metrics are expected to exhibit an asymptotic convergence at least as good with respect to the error model (6.41) to be competitive for curvilinear mesh adaptation, to provide a gain compared to standard anisotropic mesh adaptation.

We first consider $f = x^2 + y^2$. For this field, the optimal metric is constant and given by

$\mathcal{M}^{\text{opt}} = c|H_f| = c\,\text{diag}(2,2)$ where c controls the mesh density, thus the mesh is isotropic and of constant size. Straight meshes adapted to each metric are shown in Fig. 6.8 for $\alpha = 64$ and $\epsilon = \alpha^{-2}$, and by setting c so that $\mathcal{C}(\mathcal{M}^{\text{opt}}) = \mathcal{C}(\mathcal{M}_{\text{iso}})$.

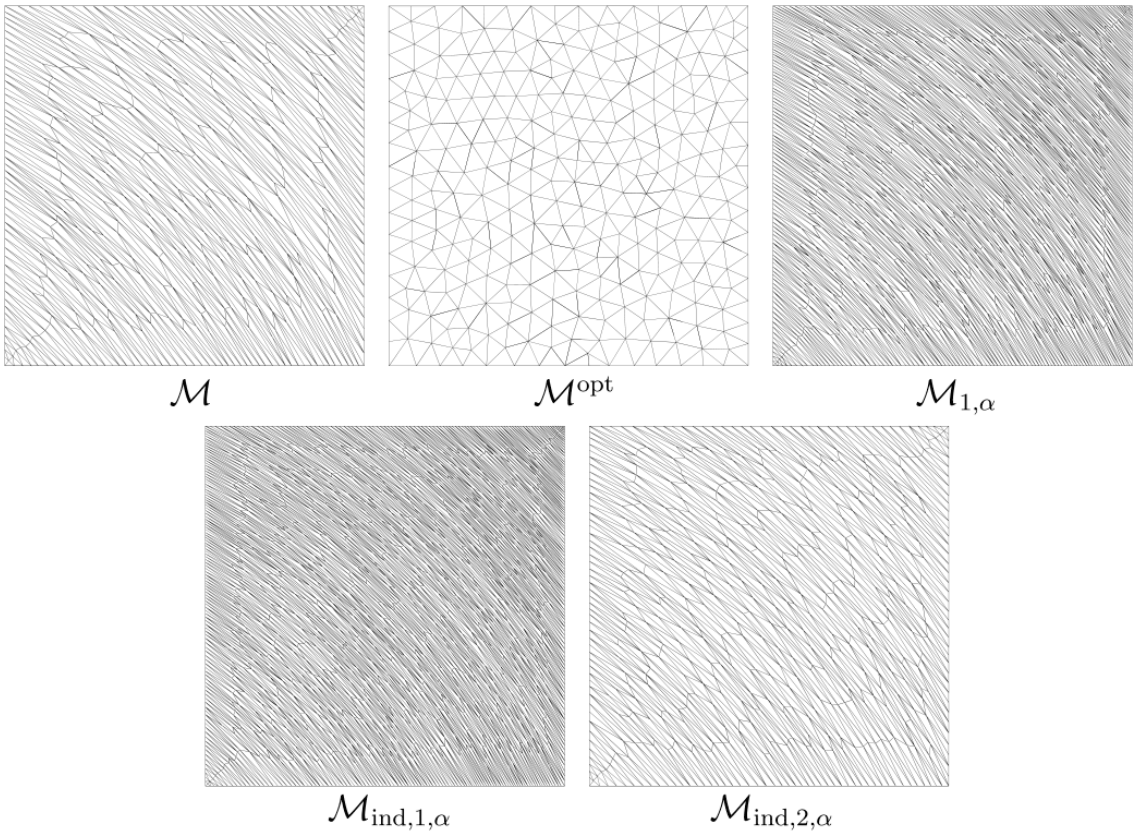


Figure 6.8 Quasi-unit straight meshes adapted to $f = x^2 + y^2$.

For analytic metrics, the exact derivatives $\partial_m \mathcal{M}^{-1/2}$ are used to evaluate (6.41). For $\mathcal{M}_{\text{iso}}$ and $\mathcal{M}^{\text{opt}}$, $\partial_m \mathcal{M}^{-1/2}$ is interpolated at the quadrature nodes from the area-weighted derivatives computed at the vertices using the gradient of the finite element shape functions. At a mesh vertex $\boldsymbol{x}_i$, these derivatives are obtained as:

$$\partial_m \mathcal{M}^{-1/2}(\boldsymbol{x}_i) = \frac{\sum\limits_{K \in \omega(\boldsymbol{x}_i)} |K|_{\overline{\mathcal{M}}} \left(\sum\limits_j \mathcal{M}^{-1/2}(\boldsymbol{x}_{K,j}) \dfrac{\partial \phi_j}{\partial x^m}(\boldsymbol{x}_i) \right)}{\sum\limits_{K \in \omega(\boldsymbol{x}_i)} |K|_{\overline{\mathcal{M}}}}, \tag{6.47}$$

where $\omega(\boldsymbol{x}_i)$ is the patch of triangles directly adjacent to $\boldsymbol{x}_i$. The nodal derivatives are then interpolated using the shape functions. Note that since the metric derivatives are symmetric but not positive-definite in general, the log-Euclidean interpolation cannot be

used to interpolate the $\partial_m \mathcal{M}^{-1/2}(\boldsymbol{x}_i)$. The derivatives of the log-Euclidean interpolation described in Section 3.6.1, however, can be used by including a factor -1/2 in front of the logarithms, as suggested by Rochery [33], to interpolate $\mathcal{M}^{-1/2}$ instead of $\mathcal{M}$. The evolution with the metric complexity $\mathcal{C}$ of both continuous error models for f is shown in Fig. 6.9: with the striking exception of $\mathcal{M}_{\text{ind},1,\alpha}$, all metrics exhibit second order convergence for the curved error model. As $\mathcal{M}^{\text{opt}}$ is constant, its derivatives vanish and the curved error trivially reduces to the linear error. $\mathcal{M}_{\text{iso}}$, $\mathcal{M}_{1,\alpha}$ and $\mathcal{M}_{\text{ind},2,\alpha}$ yield a smaller error than the optimal metric with straight elements with a factor of 2-3, which is encouraging, but still far from the accelerated convergence of the motivating examples of Section 2.7.

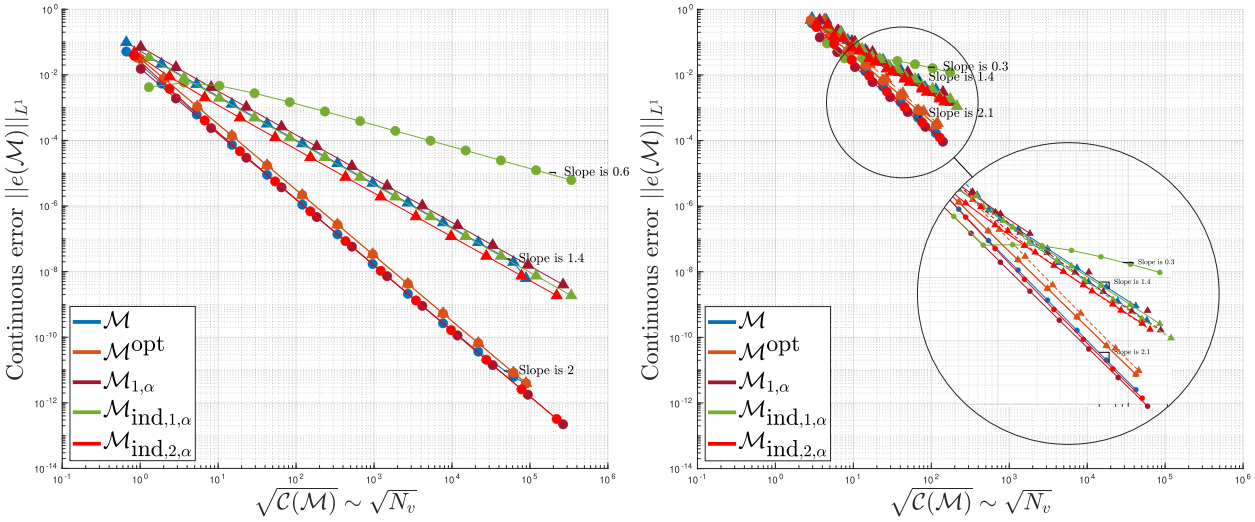


Figure 6.9 Continuous error for $f = x^2 + y^2$ versus metric complexity. Left: error model for linear interpolation on linear (triangles) and quadratic (circles) elements evaluated on a uniform grid and for increasing density coefficient α. Right: continuous error evaluated on adapted straight meshes and discrete L^1 error on these meshes (dashed lines with triangles).

The proposed metric $\mathcal{M}_{\text{iso}}$ deviates from second order starting around $\sqrt{\mathcal{C}} = 10^5$, thus past an order of magnitude of 10 billion vertices, which is way beyond the density of the considered applications. Thus, $\mathcal{M}_{\text{iso}}$ can be considered to yield second order convergence with curved triangles for acceptable complexities $\mathcal{C} \in [1, 10^{10}]$. All the metrics based on the induced metric are suboptimal for adaptation with straight elements, with a predicted rate of 1.4. This was first observed for $\mathcal{M}_{1,\alpha}$ by Loseille and Alauzet in [9]. This rate is observed in the discrete error measured on quasi-unit straight meshes adapted for these metrics using `mmg2d` without gradation and shown in dashed lines on the right figure. A good match is observed between the discrete and continuous linear errors (plain versus dashed lines with triangle markers), indicating that the straight meshes are adequate discrete counterparts for each metric. These observations also match those of Sanjaya et al. [29], although, in our

opinion, their conclusion should be tempered. Indeed, they conclude that curving becomes more beneficial as the mesh is refined, emphasizing an improvement in the convergence rate from 1.4 to 2 for $\mathcal{M}_{1,\alpha}$. However, $\mathcal{M}_{1,\alpha}$ is suboptimal for linear adaptation, so that curving with this metric merely recovers the optimal rate of $\mathcal{M}^{\text{opt}}$, with an improvement of a constant factor 2-3. In comparison, the examples of Section 2.7 exceed this optimal rate, making them particularly appealing.

We also consider the fields $f = x^3 + y^3$ on $[0.5, 1]^2$ and $f = x^3 + x^2y + \tanh[10\,(2x - \sin 5y)]$ on $[-1, 1]^2$, for which quasi-unit straight meshes are shown in Fig. 6.10. The proposed isoline metric $\mathcal{M}_{\text{iso}}$ yields second order convergence for the first field only up to $\mathcal{C} \simeq 10^4$, after which it becomes suboptimal, Fig. 6.11. A similar behaviour is observed for the tanh field for reasonable complexities ($< 10^5$). The behaviour of the other metrics is surprising. The optimal metric $\mathcal{M}^{\text{opt}}$ behaves equally well for the linear and curved error models, even predicting a slightly smaller curved error. Aside from $\mathcal{M}^{\text{opt}}$, the only metric that behaves well on all cases is $\mathcal{M}_{\text{ind},2,\alpha}$, which has uniform principal sizes everywhere and uniform anisotropy that only scales with α. This metric is suboptimal for linear adaptation, but always yields second order convergence with curved elements. On the contrary, the induced metric $\mathcal{M}_{\text{ind},1,\alpha}$ with sizes scaled by α along the isocontours and α^2 along the gradient yields terrible convergence for all tested fields. We currently do not have an explanation for these results.

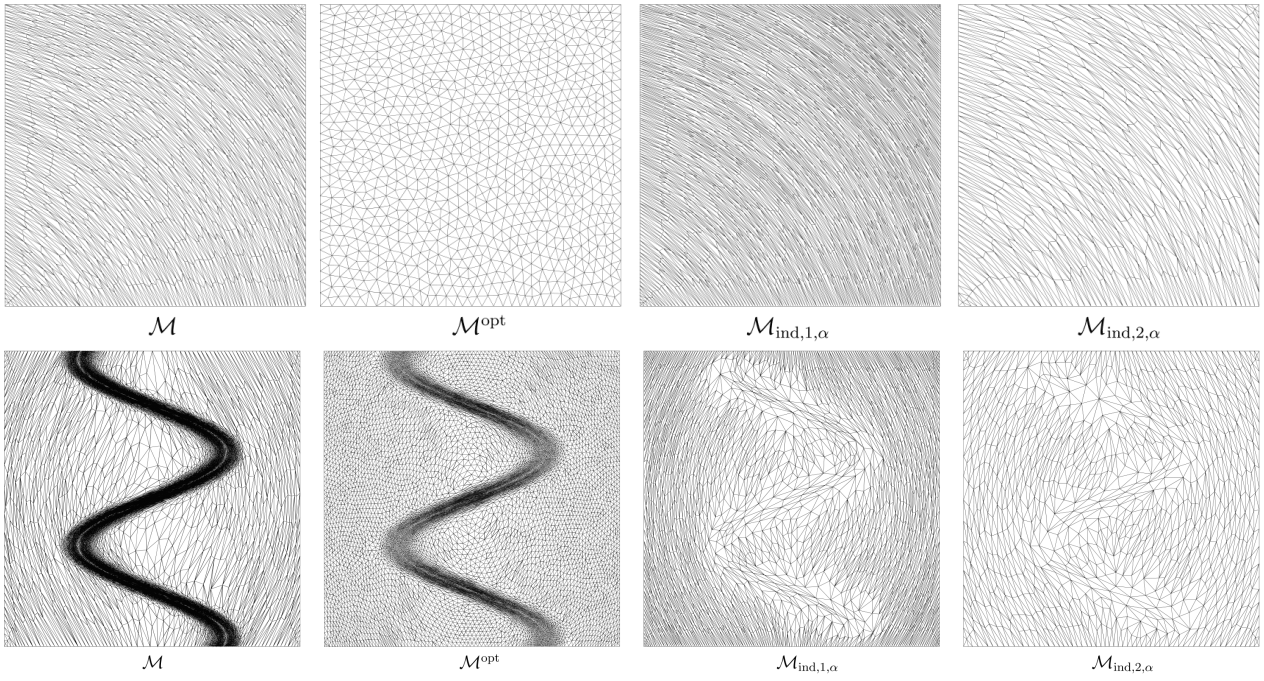


Figure 6.10 Quasi-unit straight meshes for $f = x^3+y^3$ and $f = x^3+x^2y+\tanh[10\,(2x-\sin 5y)]$.

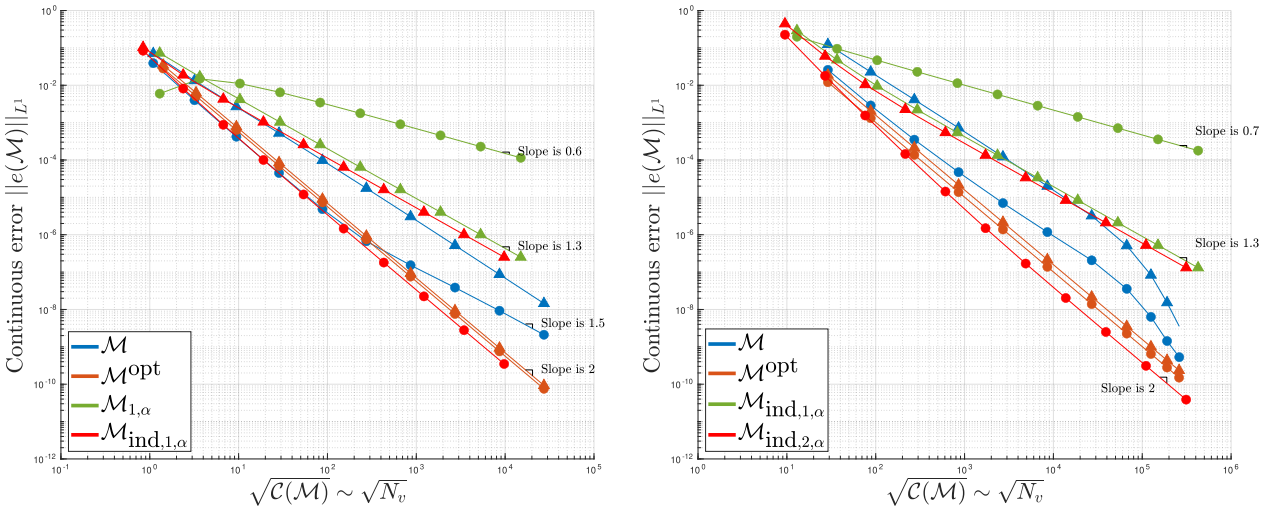


Figure 6.11 Continuous error versus metric complexity for $f = x^3 + y^3$ (left) and $f = x^3 + x^2y + \tanh[10\,(2x - \sin 5y)]$ (right). Error model for linear interpolation on linear (triangles) and quadratic (circles) elements evaluated on a uniform grid and for increasing density coefficient α.

6.2 Extension of the log-simplex method to curved triangles

We now propose to improve on the isocontour metric of Section 6.1 and on the metric from Bawin et al. [41] by considering a fixed-point method in the reference space for the composition $f \circ F(\mathcal{M})$. This is motivated by the comparison hereafter of the metric computation schemes discussed so far, according to whether the computation is performed in the reference or physical space. We also provide preliminary results illustrating the Riemannian metrics computed with this method.

6.2.1 Reference and physical space metric computation

The derivation of the optimal Riemannian metric minimizing the interpolation error on **linear meshes**, reviewed in Chapter 5, requires finding an upper bound on the derivatives of order $k+1$ on the field f of interest. This upper bound is the field of SPD matrices of minimum determinant that dominates in all directions the directional derivatives induced by $D^\alpha f$ with $|\alpha| = k+1$. In the physical space, identified to the Riemannian manifold M, these constraints at $\boldsymbol{p} \in M$ have already been written as:

$$\left| \sum_{|\alpha|=k+1} D^\alpha f(\boldsymbol{p})\, \boldsymbol{\xi}^\alpha \right| = |\pi_{\boldsymbol{p}}(\boldsymbol{\xi})| \leq \left(\boldsymbol{\xi}^T \mathcal{Q}(\boldsymbol{p}) \boldsymbol{\xi} \right)^{\frac{k+1}{2}}, \qquad \forall\, \boldsymbol{\xi} \in T_{\boldsymbol{p}} M, \tag{PS}$$

whereas we have in the reference space at the point $\boldsymbol{y} = F^{-1}(\boldsymbol{p}) \in \mathbb{R}^2$:

$$\left| \sum_{|\alpha|=k+1} D^\alpha (f \circ F(\mathcal{M}))(\boldsymbol{y})\, \boldsymbol{\zeta}^\alpha \right| = |\pi_{\boldsymbol{y}}(\boldsymbol{\zeta})| \leq \left(\boldsymbol{\zeta}^T \mathcal{Q}_\triangle(\boldsymbol{y}) \boldsymbol{\zeta} \right)^{\frac{k+1}{2}}, \qquad \forall\, \boldsymbol{\zeta} \in T_{\boldsymbol{y}}\mathbb{R}^2 \cong \mathbb{R}^2. \tag{RS}$$

The equivalence of these two constraints has been shown for linear elements in Section 4.5.2: if $\mathcal{Q}_\triangle$ is SPD of minimum determinant bounding $D^\alpha(f \circ F)$, then its pushforward $\mathcal{Q} = \mathcal{M}^{1/2}\mathcal{Q}_\triangle\mathcal{M}^{1/2}$ is SPD of minimum determinant and bounds $D^\alpha f$ in all directions in the tangent space at $\boldsymbol{p}$. The equivalence only holds for linear unit elements: as the transformation F is linear and its Jacobian matrix is $J_F = \mathcal{M}^{-1/2}R$, the $k+1$ instances of $\mathcal{M}^{-1/2}$ originating from the chain rule $D^\alpha(f \circ F)$ are paired with the $k+1$ tangent vectors $\boldsymbol{\zeta}$ to yield their pushforward $dF_{\boldsymbol{p}}(\boldsymbol{\zeta}) = J_F\boldsymbol{\zeta} = \mathcal{M}^{-1/2}R\boldsymbol{\zeta} = \boldsymbol{\xi} \in T_{\boldsymbol{p}}M$. For higher-order elements, the higher-order derivatives of F are not accounted for in the physical space formulation (PS).

Both formulations (RS) and (PS) describe a manifold $(M, \mathcal{M})$ isometric to the reference space. In (PS), we place ourselves directly on M. The metric $\mathcal{Q}$ bounds the physical-space derivatives and the Riemannian metric is set to[3] $\mathcal{M} = \mathcal{Q}$. In other words, we solve (PS) on a manifold with unknown metric, then set the metric to $\mathcal{Q}$. The lengths with respect to this metric bound the derivatives of f at each $\boldsymbol{p}$, as the left-hand side of (PS) is $(\boldsymbol{\xi}^T\mathcal{Q}\boldsymbol{\xi})^{(k+1)/2} = \ell_{\mathcal{Q}}^{k+1}(\boldsymbol{\xi})$. As error estimates for linear elements can be formulated directly on M and do not involve $\mathcal{M}$, the metric is explicitly obtained from $\mathcal{Q}$. For linear interpolation on linear elements, for instance, $\mathcal{M} = \mathcal{Q} = |H_f|$, and a subsequent scaling allows to recover the optimal metric of Chapter 5.

On the other hand, (RS) requires the metric from the start as the transformation F is involved. This metric now satisfies the nonlinear relation:

$$\mathcal{M} = \mathcal{Q} = g(\mathcal{M}) \triangleq \mathcal{M}^{1/2}\mathcal{Q}_\triangle(\mathcal{M})\mathcal{M}^{1/2}, \tag{6.48}$$

which can be solved with e.g. fixed-point iterations:

$$\mathcal{M}_{i+1} = \mathcal{Q}_i = \mathcal{M}_i^{1/2}\mathcal{Q}_{\triangle,i}(\mathcal{M}_i)\mathcal{M}_i^{1/2}. \tag{6.49}$$

In this approach, the manifold $(M, \mathcal{M})$ is constructed iteratively by solving (RS) in the reference space and mapping the bound $\mathcal{Q}_\triangle$ to M by $\mathcal{M}^{1/2}$. In other words, we construct a sequence of tangent spaces which characterize the manifold as the number of mesh vertices

[3] In Chapter 5, the metric is set to $\mathcal{M} = c_{\mathcal{M}}(\det \mathcal{Q})^\beta \mathcal{Q}$ to account for both a target number of vertices in the final mesh, through the constant $c_{\mathcal{M}}$, and for the equidistribution of the interpolation error, through a scaling by the determinant. Here, this amounts to consider only the "raw" metric, before any scaling.

increases. At convergence, we have $\mathcal{Q}_\triangle = I$, that is, $\mathcal{M}$ bounds the derivatives of f in the physical space. The convergence of this fixed-point method is guaranteed if g is a contraction, that is, if the distance between the images of any $\mathcal{M}, \mathcal{N} \in \mathrm{Sym}_2^+$ satisfies $d(g(\mathcal{M}), g(\mathcal{N})) \leq \mu\, d(\mathcal{M}, \mathcal{N})$ for $0 \leq \mu < 1$. It is trivially the case for linear interpolation, as $g(\mathcal{M}) = |H_f|$ so that $d(g(\mathcal{M}), g(\mathcal{N})) = |H_f| - |H_f| = 0$, see below. For quadratic interpolation, $g(\mathcal{M})$ depends on $\mathcal{M}$ through the coefficients of the polynomial $\pi_{\boldsymbol{p}}(\boldsymbol{\xi})$. Its exact form from Mirebeau's solution (Algo. 2) depends on the classification of the polynomial, which makes it arduous to determine whether g is a contraction. Similarly, we did not check for higher-order interpolation, as g is not an explicit function of $\mathcal{M}$.

For linear interpolation on linear elements, (RS) reduces to:

$$\left|D^2(f \circ F)\zeta^i\zeta^j\right| = \left|\left(\mathcal{M}^{-1/2} H_f \mathcal{M}^{-1/2}\right)_{ij} \zeta^i\zeta^j\right| = \left|\boldsymbol{\zeta}^T \left(\mathcal{M}^{-1/2} H_f \mathcal{M}^{-1/2}\right) \boldsymbol{\zeta}\right| \leq \boldsymbol{\zeta}^T \mathcal{Q}_\triangle \boldsymbol{\zeta}, \quad (6.50)$$

for which $\mathcal{Q}_\triangle = |\mathcal{M}^{-\frac{1}{2}} H_f \mathcal{M}^{-\frac{1}{2}}| = \mathcal{M}^{-\frac{1}{2}} |H_f| \mathcal{M}^{-\frac{1}{2}}$ satisfies the equality. It is then easy to see that the fixed-point method converges in a single iteration for any choice of initial $\mathcal{M}$:

$$\mathcal{M}_1 = \mathcal{Q}_1 = \mathcal{M}_0^{1/2} \mathcal{Q}_{\triangle,0} \mathcal{M}_0^{1/2} = \mathcal{M}_0^{1/2} \mathcal{M}_0^{-1/2} |H_f| \mathcal{M}_0^{-1/2} \mathcal{M}_0^{1/2} = |H_f|, \quad (6.51)$$

that is, $g(\mathcal{M}) = |H_f|$ and the subsequent iterations yield:

$$\mathcal{Q}_{\triangle,i+1} = |H_f|^{-1/2} |H_f| |H_f|^{-1/2} = I, \qquad \mathcal{M}_{i+1} = |H_f|^{1/2} I |H_f|^{1/2} = |H_f|. \quad (6.52)$$

Similarly, for quadratic interpolation on linear elements, an ellipse of maximum area associated to $\mathcal{Q}_\triangle$ can be inscribed in $\Lambda(\pi_{\boldsymbol{y}})$, the level set 1 of $\pi_{\boldsymbol{y}}$, using Mirebeau's analytical solution described in Algorithm 2. As $\mathcal{Q}_\triangle$ is the exact upper bound, $\mathcal{M}$ is also obtained in a single fixed-point iteration. The log-simplex method described in Section 5.4.2 tackles higher-order interpolants on linear elements. However, the matrix $\mathcal{Q}_\triangle$ is no longer exact as it is obtained from a linearization of (RS) written for the log-matrix $\mathcal{L}_\triangle = \log \mathcal{Q}_\triangle$. Thus, $\mathcal{M}$ is no longer obtained in a single iteration as $\mathcal{Q} \neq \mathcal{M}^{1/2} (\exp \mathcal{L}_\triangle) \mathcal{M}^{1/2}$, but if the sequence of fixed-point iterates converges, then it converges to $\mathcal{Q}$, as shown in Coulaud and Loseille [11] and in Section 5.4.2. In other words, the fixed-point method corrects the effect of the linearization.

6.2.2 Application to curved triangles

For curved triangles, only the reference space formulation (RS) has a straightforward extension by considering the composition $D^\alpha(f \circ F(\mathcal{M}))$. Higher-order derivatives of F, thus

derivatives of $\mathcal{M}^{-1/2}$, are now involved in addition to powers of $\mathcal{M}$. As a result, the nature of the problem changes from the nonlinear relation (6.48) to a system of PDE in the components of $\mathcal{M}$. For linear interpolation on $\mathcal{P}^2$ triangles, for instance, the chain rule yields the following reference space constraints for any $\boldsymbol{\zeta} \in \mathbb{R}^2$:

$$
\begin{aligned}
&\left| \left(\left.\frac{\partial^2 f}{\partial x^k \partial x^\ell}\right|_{F(\boldsymbol{y})} \frac{\partial F^\ell}{\partial y^j}\frac{\partial F^k}{\partial y^i} + \left.\frac{\partial f}{\partial x^k}\right|_{F(\boldsymbol{y})} \left.\frac{\partial^2 F^k}{\partial y^i \partial y^j}\right|_{F(\boldsymbol{y})} \right) \zeta^i \zeta^j \right| \leq \boldsymbol{\zeta}^T \mathcal{Q}_\triangle \boldsymbol{\zeta} \\
&\qquad\qquad \Big\downarrow \quad J_F = \mathcal{M}^{-1/2} R,\ \widetilde{\boldsymbol{\zeta}} = R\boldsymbol{\zeta} \\
&\Bigg| \underbrace{\left(\left.\frac{\partial^2 f}{\partial x^k \partial x^\ell}\right|_{F(\boldsymbol{y})} \mathcal{M}^{-1/2}_{\ell j}\mathcal{M}^{-1/2}_{ki} + \left.\frac{\partial f}{\partial x^k}\right|_{F(\boldsymbol{y})} \left.\frac{\partial \mathcal{M}^{-1/2}_{ki}}{\partial x^m}\right|_{F(\boldsymbol{y})} \mathcal{M}^{-1/2}_{mj} \right)}_{\triangleq\ T_{ij}(\mathcal{M})} \widetilde{\zeta}^i \widetilde{\zeta}^j \Bigg| \leq \widetilde{\boldsymbol{\zeta}}^T R \mathcal{Q}_\triangle R^T \widetilde{\boldsymbol{\zeta}} \\
&\qquad\qquad \Big\downarrow \quad \widetilde{\mathcal{Q}}_\triangle = R\mathcal{Q}_\triangle R^T \\
&\left| \left(\mathcal{M}^{-1/2} H_f \mathcal{M}^{-1/2} + \nabla f \cdot \partial_m \mathcal{M}^{-1/2} \cdot \mathcal{M}^{-1/2} \right) \widetilde{\zeta}^i \widetilde{\zeta}^j \right| \leq \widetilde{\boldsymbol{\zeta}}^T \widetilde{\mathcal{Q}}_\triangle \widetilde{\boldsymbol{\zeta}} \qquad (6.53) \\
&\qquad\qquad \Big\downarrow \quad \text{The } \widetilde{\boldsymbol{\zeta}} \text{ are arbitrary} \\
&\qquad\qquad |\widetilde{\boldsymbol{\zeta}}^T T \widetilde{\boldsymbol{\zeta}}| \leq \widetilde{\boldsymbol{\zeta}}^T \widetilde{\mathcal{Q}}_\triangle \widetilde{\boldsymbol{\zeta}}.
\end{aligned}
$$

where $\widetilde{\boldsymbol{\zeta}} = R\boldsymbol{\zeta}$ includes the rotation matrix appearing in the derivatives of F. The pullback of the Hessian is symmetric but not positive-definite in general, and the additional term accounting for curvature is not symmetric nor definite, so neither is T. The left-hand side is still a homogeneous quadratic polynomial in $\boldsymbol{\zeta}$, whose coefficients are also described by the symmetric part of T:

$$
\pi_{\boldsymbol{y}}(\boldsymbol{\zeta}) = \boldsymbol{\zeta}^T T \boldsymbol{\zeta} = \boldsymbol{\zeta}^T \left(\frac{T + T^T}{2} \right) \boldsymbol{\zeta}, \qquad \forall\, \boldsymbol{\xi} \in \mathbb{R}^2. \tag{6.54}
$$

An SPD upper bound on $\pi_{\boldsymbol{y}}$ is thus obtained by setting $\mathcal{Q}_\triangle = |(T + T^T)/2|$. Letting Sym(.) denote the symmetric part of a matrix, the components of $\mathcal{M}$ satisfy the PDE:

$$
\begin{aligned}
\mathcal{M} = \mathcal{Q} &= \mathcal{M}^{1/2} \mathcal{Q}_\triangle \mathcal{M}^{1/2} \\
&= \mathcal{M}^{1/2} \left| \mathcal{M}^{-1/2} H_f \mathcal{M}^{-1/2} + \mathrm{Sym}\left(\nabla f \cdot \partial_m \mathcal{M}^{-1/2} \cdot \mathcal{M}^{-1/2} \right) \right| \mathcal{M}^{1/2},
\end{aligned} \tag{6.55}
$$

or equivalently:

$$
\left| \mathcal{M}^{-1/2} H_f \mathcal{M}^{-1/2} + \mathrm{Sym}\left(\nabla f \cdot \partial_m \mathcal{M}^{-1/2} \cdot \mathcal{M}^{-1/2} \right) \right| = I. \tag{6.56}
$$

As all matrices are symmetrized, this is, in two dimensions, a system of 3 coupled and quasilinear first-order PDE in the components of $\mathcal{M}$. For $\mathcal{P}^q$ elements, this is a system of

$(q-1)$-th order PDE instead, as derivatives of order q of F remain. The derivatives of f are known and act as source terms. An exact resolution of this system of PDE, if at all possible depending on f, seems arduous. Before commenting further, we observe that for quadratic interpolation, the homogeneous cubic polynomial $\pi_{\boldsymbol{y}}(\boldsymbol{\zeta})$ in (RS) is obtained from (4.109):

$$\pi_{\boldsymbol{y}}(\boldsymbol{\zeta}) = \underbrace{\left(C_{f,mno}(F(\boldsymbol{y}))\mathcal{M}_{mi}^{-\frac{1}{2}}\mathcal{M}_{nj}^{-\frac{1}{2}}\mathcal{M}_{ok}^{-\frac{1}{2}} + \sum_{\substack{\text{cyclic} \\ i,j,k}} H_{f,mn}(F(\boldsymbol{y}))\mathcal{M}_{nk}^{-\frac{1}{2}}\mathcal{M}_{oj}^{-\frac{1}{2}}\partial_o\mathcal{M}_{mi}^{-\frac{1}{2}} \right)}_{\triangleq\ T_{ijk}} \zeta^i\zeta^j\zeta^k. \tag{6.57}$$

The upper bound $\mathcal{Q}_{\triangle}(\mathcal{M}_i, \partial_m\mathcal{M}_i^{-1/2})$ can still be solved with Mirebeau's solution as $\pi_{\boldsymbol{y}}(\boldsymbol{\zeta})$ is still homogeneous, however, it is much harder to determine the influence of the metric and its derivatives on $\mathcal{Q}_{\triangle}$, as it is obtained from geometric considerations depending on the coefficients of the cubic polynomial. In other words, it may no longer be possible to derive a PDE similar to (6.56) starting with quadratic interpolation. For higher-order interpolation, the ellipse associated to $\mathcal{Q}_{\triangle}$ and inscribed in the level curve 1 of $\pi_{\boldsymbol{y}}$ of degree $k+1$ is approximated by the solution of the linearized log-problem (5.76), so that the link between $\mathcal{Q}_{\triangle}$ and the metric and its derivatives becomes even more tenuous. It seems thus reasonable to claim that $\mathcal{M}$ should be determined numerically as soon as curvature is involved, even for the simplest case of linear interpolation, either with an ad hoc numerical scheme to be devised in the future, or by constructing the metric field iteratively within the MOESS framework [31].

6.2.3 Naive resolution of the metric PDE

To understand the behaviour of the metric for linear interpolation on a very simple test case, we consider here, as a simple extension of the approach for linear elements, the fixed-point iteration:

$$\begin{aligned} \mathcal{M}_{i+1} &= \mathcal{M}_i^{1/2}\left|\mathcal{M}_i^{-1/2}H_f\mathcal{M}_i^{-1/2} + \mathrm{Sym}\left(\nabla f\cdot\partial_m\mathcal{M}_i^{-1/2}\cdot\mathcal{M}_i^{-1/2}\right)\right|\mathcal{M}_i^{1/2}, \\ \partial_m\mathcal{M}_{i+1}^{-1/2} &= \partial_m(\mathcal{M}_{i+1}^{-1/2}). \end{aligned} \tag{6.58}$$

The fixed-point iterates $\mathcal{M}_{i+1}$ are computed for a fixed metric derivative, which are then updated from $\mathcal{M}_{i+1}$. The conditions under which this scheme converges have not been investigated, however, it is clear that if the sequence $\mathcal{M}_i$ converges, it converges to a solution of (6.55). For higher-order interpolation, the fixed-point iterations would instead take the

general form:

$$\begin{aligned} \mathcal{M}_{i+1} &= \mathcal{M}_i^{1/2}\mathcal{Q}_{\triangle}(\mathcal{M}_i, \partial_m\mathcal{M}_i^{-1/2})\mathcal{M}_i^{1/2}, \\ \partial_m\mathcal{M}_{i+1}^{-1/2} &= \partial_m(\mathcal{M}_{i+1}^{-1/2}). \end{aligned} \tag{6.59}$$

The major difference with linear elements, covered by the log-simplex method, is that the derivatives $\partial\mathcal{M}^{-1/2}$ depend on the neighbouring values of $\mathcal{M}^{-1/2}$, so it is no longer possible to solve the fixed-point problem independently at each mesh vertex; instead, the metric and its derivatives must be converged together on the whole mesh. The derivatives of $\mathcal{M}^{-1/2}$ must be evaluated at the mesh vertices. This can be achieved either by recovering each component with Zhang & Naga's procedure, or with the weighted average (6.47). We chose the latter and more local option, as the information is taken from directly adjacent elements to average the gradients. The metrics at neighbouring vertices may not converge at the same rate, meaning that they may not require the same number of fixed-point iterations. Consequently, recomputing their derivatives after each iteration leads to a nonsmooth field, resulting in inaccurate metrics at the next iteration. To mitigate this issue, the metrics $\mathcal{M}_j$ can be converged at all vertices in an inner loop with fixed derivatives $\partial_m\mathcal{M}_i^{-1/2}$, ensuring that the next derivatives are recovered from a smooth metric field:

While $\|\partial_m\mathcal{M}_{i+1}^{-1/2} - \partial_m\mathcal{M}_i^{-1/2}\|_F > \epsilon_{\text{derivatives}}$:
- For each vertex :
  - While $\|\mathcal{M}_{j+1} - \mathcal{M}_j\|_F > \epsilon_{\text{metrics}}$:
    - $\mathcal{M}_{j+1} = \mathcal{M}_j^{1/2}\mathcal{Q}_{\triangle}(\mathcal{M}_j, \partial_m\mathcal{M}_i^{-1/2})\mathcal{M}_j^{1/2}$
- $\partial_m\mathcal{M}_{i+1}^{-1/2} = \partial_m(\mathcal{M}_j^{-1/2})$ for $m = 1, 2$

This is summarized in Algorithm 4. Computing the first outer iteration with $\partial_m\mathcal{M}_0^{-1/2} = 0$ yields the metric field of the original log-simplex method, as all metrics are converged without additional term to account for the curvature. One immediately notices that if the first iteration yields a constant metric field, then $\partial_m\mathcal{M}_1^{-1/2} = 0$ and convergence is reached within a single iteration. This happens when the order $k+1$ derivatives of f are constant, as then the upper bound is the same for all vertices. For instance, and contrary to the metric of Section 6.1, $f = x^2+y^2$ with Hessian $H_f = \text{diag}(2, 2)$ yields the constant metric field $\mathcal{M} = |H_f|$. Note that the method proposed in [41] is essentially a single outer loop iteration of the fixed-point scheme above, where the metric derivatives are set beforehand by considering the family of curves described by (6.38) and setting their curvature.

Example: $f(x,y) = x^3 + y^3$

Preliminary results are presented to illustrate the method. Since $f = x^2 + y^2$ has constant Hessian and yields zero metric derivatives, as discussed above, we consider instead $f = x^3+y^3$ in $[0.1, 1]^2$, and apply the fixed-point method described in Algorithm 4 to derive a field $\mathcal{Q} = \mathcal{M}$ satisfying (RS) on a starting uniform mesh. Because a fixed-point scheme is not best suited to solve PDEs, the metric derivatives prove difficult to converge, contrary to the linear case which typically requires a few fixed-point iterations. To facilitate the convergence, the derivatives are under-relaxed according to:

$$\partial_m \mathcal{M}_{i+1}^{-1/2} = \omega \partial_m (\mathcal{M}_{i+1}^{-1/2}) + (1-\omega)\partial_m \mathcal{M}_i^{-1/2}, \tag{6.60}$$

with a relaxation factor $\omega = 0.1$. In each inner loop of Algorithm 4, the metric tensors are converged for the current derivatives up to 10^{-10}, and the outer loop stops when two successive derivatives fields differ by less than 10^{-6} in max norm. The ellipses representing the resulting Riemannian metrics are shown in Fig. 6.12 for both linear (in black) and quadratic (in blue) triangles. The optimal metric is essentially a scaled version of the metric for linear elements. In particular, it does not follow the isocontours of f as would do $\mathcal{M}_{\text{iso}}$.

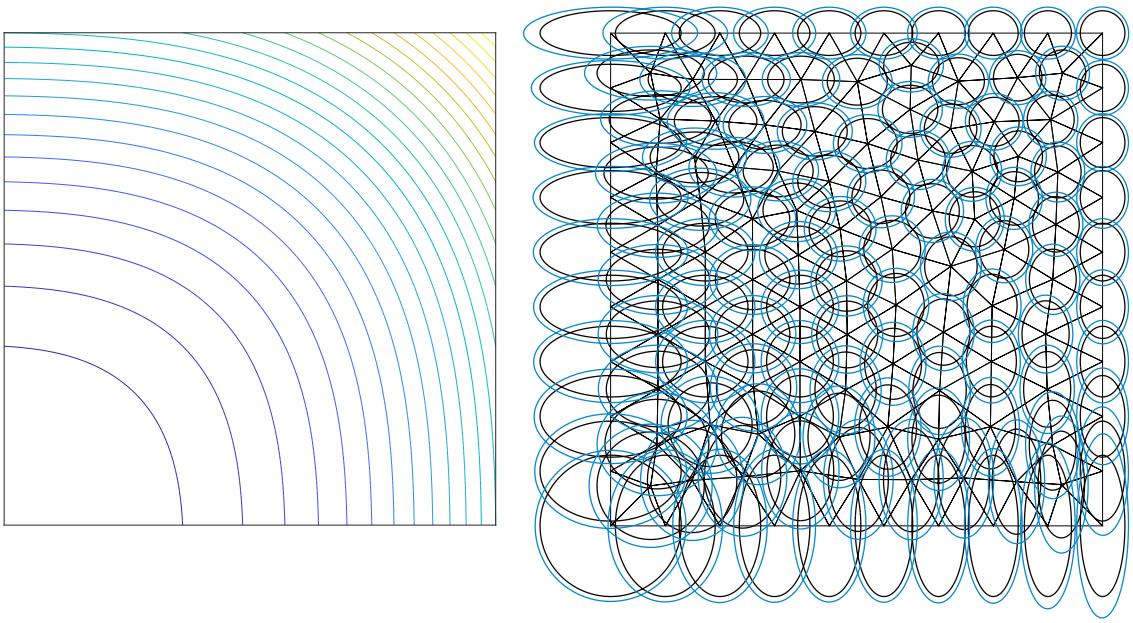


Figure 6.12 Unit tangential spheres ∂B_1 of the metric field $\mathcal{M}$ computed with Algorithm 4 for $f = x^3 + y^3$. Solution for linear interpolation on linear ($\mathcal{P}^1$, in black) and quadratic ($\mathcal{P}^2$, in blue) triangles.

Algorithm 4 Log-simplex method for adaptation with degree k interpolation on $\mathcal{P}^q$ triangles. The boxed section applied to $f \circ F$ is the log-simplex method for linear triangles discussed in Section 5.4.2.

Input: Polynomial degree $m = k + 1$, derivatives of the field of order $k + 1$ through $k - q + 1$, stopping tolerance ϵ, maximum number of iterations `maxIter`.

Output: Anisotropic measure $\mathcal{Q}$ (= Riemannian metric $\mathcal{M}$ before scaling) for Π^k interpolation on $\mathcal{P}^q$ triangles.

function LogSimplexCurved(`m`, ϵ, `maxIter`)
  Set $i = j = 0$, $\mathcal{M}_j = I$ and $\partial\mathcal{M}_i^{-1/2} = 0$
  Set `derivativesConverged` = `false`
  while `not derivativesConverged` and $i <$ `maxIter` **do** ▷ Outer derivatives loop

    for all vertices $\boldsymbol{x}$ **do** ▷ Log-simplex method
      Set `metricConverged` = `false`
      while `not metricConverged` and $j <$ `maxIter` **do** ▷ Inner metrics loop
        Compute derivatives tensor $T(\mathcal{M}_j, \partial\mathcal{M}_i^{-1/2}) \triangleq D^\alpha(f \circ F(\mathcal{M}_j))$ at $\boldsymbol{x}$

        if `m` = 2 **then** ▷ Linear interpolation
          $\mathcal{Q}_{\triangle,j}(\boldsymbol{x}) = |(T + T^T)/2|$
        else if `m` = 3 **then** ▷ Quadratic interpolation
          `coeff`$(\pi_{\boldsymbol{y}}) = [T_{111}, T_{112} + T_{121} + T_{211}, T_{122} + T_{212} + T_{221}, T_{222}]$
          $\mathcal{Q}_{\triangle,j}(\boldsymbol{x}) =$ AnisotropicMeasureQuadratic(`coeff`$(\pi_{\boldsymbol{y}})$) ▷ Algo. 2
        else
          Set `coeff`$(\pi_{\boldsymbol{y}})$ from T
          $\mathcal{Q}_{\triangle,j}(\boldsymbol{x}) = \exp\left(\mathcal{L}_\triangle(\Lambda(\pi_{\boldsymbol{y}}))\right)$ ▷ Solve log-problem for $\pi_{\boldsymbol{y}}$
        end if

        Recover the physical-space solution $\mathcal{Q}_j(\boldsymbol{x}) = \mathcal{M}_j^{1/2}(\boldsymbol{x})\mathcal{Q}_{\triangle,j}(\boldsymbol{x})\mathcal{M}_j^{1/2}(\boldsymbol{x})$
        Set $\mathcal{M}_{j+1}(\boldsymbol{x}) = \mathcal{Q}_j(\boldsymbol{x})$
        Compute stopping criterion `diff` $= \|\mathcal{M}_{j+1}(\boldsymbol{x}) - \mathcal{M}_j(\boldsymbol{x})\|_F / \|\mathcal{M}_j(\boldsymbol{x})\|_F$
        if `diff` $< \epsilon$ or $j >$ `maxIter` **then**
          `metricConverged` = `true`
        end if
        Increment j
      end while
    end for

    Recompute metric derivatives $\partial\mathcal{M}_{i+1}^{-1/2} = \partial(\mathcal{M}_j^{-1/2})$ at mesh vertices
    Compute stopping criterion `diff2` $= \max_{\boldsymbol{x},k}(\|\partial_k\mathcal{M}_{i+1} - \partial_k\mathcal{M}_i\|_F / \|\partial_k\mathcal{M}_i\|_F)$
    if `diff2` $< \epsilon$ or $i >$ `maxIter` **then**
      `derivativesConverged` = `true`
    end if
    Increment i
  end while
end function

6.3 Conclusion

In this chapter, we discussed Riemannian metrics for adaptation with curvilinear triangles, with the emphasis on (i) a metric controlling the maximum error on quadratic approximation of the isocontours and gradient curves of a field, and (ii) a fixed-point scheme to tackle the problem of finding the best upper bound on the derivatives of the composition $f \circ F$ for nonlinear transformations $F(\mathcal{M})$. The former metric is the metric induced by the graph of f, with principal sizes modified to limit the interpolation error on specific curves. Despite some shortcomings, it predicts a second-order convergence with respect to the continuous linear interpolation error on curved elements for the considered test cases, for varying ranges of complexity. The second metric is the scaled solution to (ii), and is the generalization of the optimal metric for linear simplices, minimizing the interpolation error on nonlinear elements. It is derived from a natural extension of the log-simplex method and is computed in the reference space. Because of the nonlinearity in $\mathcal{M}$ of the transformation $F(\mathcal{M})$, the nature of the problem to solve for $\mathcal{M}$ changes from a nonlinear equation to a system of quasilinear PDEs in the components of the metric. An efficient numerical resolution of this PDE, which is yet to derive, seems crucial to provide a complete methodology to make curvilinear mesh adaptation competitive. In this chapter, we merely generalized the log-simplex fixed-point iteration to account for the metric derivatives. When the proposed fixed-point method converges, it is a solution to this system, but it proves hard to converge in practice.

CHAPTER 7 GENERATION OF QUASI-UNIT CURVILINEAR TRIANGULATIONS

This chapter tackles the last ingredient of high-order metric-based anisotropic meshing, namely the practical generation of optimal curvilinear triangulations with respect to a Riemannian metric. The objective is to generate metric-conforming and quasi-unit elements in the sense of definitions *QU1p* and *QU2p* of Chapter 4, which are recalled here:

Definition. *(QU1p) A $\mathcal{P}^q$* ***simplex*** *K is* ***quasi-unit*** *with respect to $\mathcal{M}$ and K_0 if its quality Q_{K_0} is in $[a, 1]$ and if homologous edges e and e_0 satisfy*

$$\frac{1}{\sqrt{2}}\ell_{\overline{\mathcal{M}}}(e_0) \leq \ell_{\mathcal{M}}(e) \leq \sqrt{2}\ell_{\overline{\mathcal{M}}}(e_0). \tag{7.1}$$

Definition. *(QU2p) A $\mathcal{P}^q$* ***simplex*** *K is* ***quasi-unit*** *with respect to $\mathcal{M}$ and K_0 if*

- *it is valid;*
- *its edges approximate the geodesics, thus are of minimal length;*
- *homologous inner angles θ and θ_0 satisfy $\theta_0 - c \leq \theta \leq \theta_0 + c$;*
- *homologous edges e and e_0 satisfy* (7.1)*.*

In addition to controlling the edge length, as is done for linear quasi-unit simplices, both definitions yield triangles of controlled edge parameterization and inner angles. In *QU1p*, these are set as a by-product of the distortion-based quality, whereas *QU2p* enforces the angles and explicitly requires the edge parameterization to be length-minimizing. In the following, we describe a global remeshing method where 2D curvilinear $\mathcal{P}^2$ triangulations are obtained in two steps, described in Sections 7.1 and 7.2 respectively.

Firstly, a valid and high-quality linear mesh is generated with a frontal approach, using a front of vertices instead of a traditional front of faces. We build on existing methods proposed by e.g. Loseille et al. [42, 43], Marcum and Alauzet [101, 102] and Baudouin et al. [103], where the edges follow the principal directions of the metric, that is, are aligned with its eigenvectors. However, instead of creating right angles for the Euclidean metric by aligning the edges with two orthogonal directions, as is done in the *metric-orthogonal* approach, angles are controlled for the given metric $\mathcal{M}$. Moreover, in addition to the integral curves of the metric eigenvectors, we explore the creation of unit edges along the geodesics of $\mathcal{M}$, in agreement with *QU2p*. These two families of curves are similar for the considered applications, but, to our knowledge, are not formally linked. At the end of this step, the

length and angles of the future edges are roughly determined and the positions of the $\mathcal{P}^1$ vertices are fixed until the end of the procedure.

Secondly, the linear mesh is curved by moving the $\mathcal{P}^2$ edge vertices to minimize either the distortion, yielding high-quality *QU1p* triangles, or the Riemannian length to create quadratic approximations of the geodesics. The former, distortion minimization, is a global and expensive problem, where the mesh is improved by minimizing a nonlinear functional over all interior edge vertices [25–27]. As such, generating large-scale meshes of quasi-unit simplices by explicitly controlling their distortion is not realistic. Rather, it can be used as a gold standard against which the weaker definition *QU2p* of simplices can be compared. In contrast, length minimization is purely local and very fast. It is a straightforward approach to edge curving that has been explored in recent works on curvilinear mesh adaptation [21, 23, 24, 29, 41]. Together with a control of the inner angles, it proves an efficient way of generating quasi-unit triangles with quality indicators comparable to quasi-unit triangles in the sense of *QU1p*. We detail in Section 7.2 how each quantity is minimized, then present illustrative examples of curved meshes adapted to analytical fields and CFD solutions.

All operations (vertex sampling, edge swaps, edge curving) heavily rely on interpolating the metric and its derivatives over a background mesh. In fact, due to its constant use, metric interpolation, and especially point localization in the background mesh, proves to be the bottlenecks of the presented pipeline. Metric-based mesh generation is an iterative process, where both the mesh and the Riemannian metric are successively improved with the goal of converging them together. Here, the background mesh and its metric are the ones obtained after a few (typically 5) iterations of linear anisotropic adaptation, and a single curved mesh is generated for each test case. As curvilinear mesh generation is still exploratory, this allows to start from a known and robust pipeline. However, future advancements are expected to iterate directly over curved meshes.

The approach presented in this chapter can be applied to generate quasi-unit $\mathcal{P}^2$ triangulations for any input Riemannian metric. Naturally, the metric must be tailored to the field of interest and adequately account for element curvature to provide a gain in interpolation error. Since the optimal metric for interpolation on curved meshes, described in Chapter 6, is not yet available, we do not expect appreciable gains of interpolation error with curved elements, compared to linear meshes adapted to the optimal metric for linear elements. Nevertheless, it is a step in the right direction as we describe how ideal elements can be generated.

All methods discussed were implemented in Gmsh. However, there remains a close integration between Gmsh and our finite element solver in some steps. For instance, the background mesh/metric pair uses data structures from our code, which consequently manages mesh lo-

calization and metric interpolation. Once these routines are fully integrated into Gmsh, the entire pipeline for metric-based curved meshing will be publicly accessible.

7.1 Initial straight mesh

We start by generating a valid linear mesh with the required spacing and angles. As discussed in the introduction, a major difference with quasi-unit linear elements, where only the edge lengths matter, is that curved triangles require controlling the inner angles with respect to the Riemannian metric. Linear elements need not follow any particular direction, as long as they exhibit a satisfying quality and their edges are quasi-unit. As a result, and to our best knowledge, publicly available metric-based meshing libraries, such as `mmg` or `BAMG`, do not control the inner angles of their quasi-unit linear meshes. As in [21, 23, 41], we choose to create a linear mesh with a pipeline similar to Loseille et al. [42, 43]. This is done as follows:

1. Create the **background mesh and metric field** from the current mesh and its Riemannian metric. The background mesh is kept throughout the process for metric queries with the log-Euclidean interpolation presented in Chapter 3, as originally proposed by Aparicio-Estrems et al. [26] and Rochery [17].
2. Generate a metric-conforming linear mesh with an off-the-shelf meshing library (in this work, `mmg2d`). The interior vertices are discarded to keep only an **empty mesh**, that is, a metric-conforming discretization of the boundary.
3. Initialize a **front of vertices** containing the boundary vertices.
4. Spawn the **interior mesh vertices** at unit distance from one another until the front becomes empty. This is discussed in detail in Sections 7.1.1 and 7.1.2.
5. **Triangulate** the vertices using the isotropic Delaunay kernel, followed by edge swaps to improve the distortion-based quality. This is discussed in Section 7.1.3.

7.1.1 Alignment with the metric and control of the inner angles

Generating anisotropic meshes with enforced directions was tackled by Loseille et al. [42, 43], Marcum and Alauzet [101, 102] and more recently Xiao et al. [104], leading to the definitions of **metric-orthogonal** and **metric-aligned** meshes, with similar ideas applied in Baudouin et al. [103] in the context of hex-dominant meshing. The **metric-orthogonal** approach, which is the one on which the works above focus, generates right triangles in the Euclidean space by sampling the vertices along the integral curves of the metric eigenvectors. This

yields block-structured meshes whose edges follow the directions of the metric and with few obtuse angles, which are desirable e.g. for boundary layer simulations or to better control the interpolation error in H^1 norm [38]. This approach has been used since the first papers of Zhang et al. [21–23] for curvilinear mesh adaptation, as well as in our continuation of their work [41]. Indeed, the chosen metric in those works is the isocontour metric $\mathcal{M}_{\text{iso}}$ discussed in Chapter 6, designed to limit the interpolation error on curves aligned with the directions of the metric. It is then natural to enforce these directions in the curved mesh, thus the choice of metric-orthogonal meshing.

The **metric-aligned** approach, on the other hand, creates equilateral triangles with respect to the metric, but, unlike standard mesh generators, aligns one of their medians with the direction of highest anisotropy. Thus, it creates quasi-unit triangles in the usual sense with controlled orientation. This is achieved in [43, 101] by advancing a front of edges, where each edge proposes a new vertex in the direction of the largest metric eigenvector. The angles in the metric are not explicitly controlled, as new vertices are proposed along the eigenvectors of the metric, independently of the orientation of the active front edge.

The metric-orthogonal approach controls the orientation and the inner angles for the Euclidean metric $\overline{\mathcal{M}}$. In general, angles which are controlled for the Euclidean metric can be arbitrary with respect to $\mathcal{M}$. But because the Jacobian matrix $J_{K_0 \to K} = \mathcal{M}^{-1/2} R_0$ can also be viewed as a linear endomorphism $\mathcal{M}^{-1/2} R_0 : \mathbb{R}^2 \to \mathbb{R}^2$ from the reference space $(\mathbb{R}^2, \overline{\mathcal{M}})$ to the parameter space $(\mathbb{R}^2, \mathcal{M})$, two vectors in the reference space aligned with the eigenvectors $\boldsymbol{v}_i$ of $\mathcal{M}^{-1/2}$ (thus also of $\mathcal{M}$) are simply scaled by the linear map $\mathcal{M}^{-1/2}$ and remain aligned with $\boldsymbol{v}_i$ in the parameter space, as shown in Fig. 7.1. Conversely, aligning two edges with $\boldsymbol{v}_i$ directly in the parameter space, where the mesh is constructed, aligns them in the reference space and controls their metric angle. Because the metric-orthogonal approach aligns the edges with *both* the metric eigenvectors, their preimage in the reference space can only form a right angle. In other words, it creates right angles simultaneously for the input metric and for the Euclidean metric. Following Chapter 4, the metric-orthogonal approach hence yields angles suitable for the creation of triangles (quasi-)isometric to $K_0 = K_{\triangleright}$. In [21–23, 41], we use a metric-orthogonal vertex sampling and curve the edges to approximate geodesics, which amounts to creating unit triangles w.r.t. $K_0 = K_{\triangleright}$.

For other target simplices, for instance $K_0 = K_{\triangle}$, it is in general not possible to control the inner angles with respect to both $\mathcal{M}$ and $\overline{\mathcal{M}}$. In fact, the angles *can* be controlled for both metrics only if $\mathcal{M}$ is "stretched enough" to allow the triangles to rotate to form e.g. right angles. In that case, the dot product of mapped unit vectors w.r.t. $\overline{\mathcal{M}}$ can be set

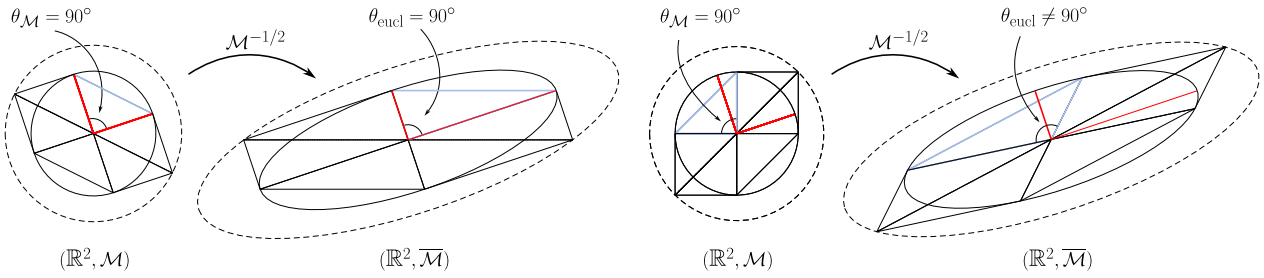


Figure 7.1 Angle control with the metric-orthogonal approach. On the left, the pattern of right triangles (copies of $K_{\triangle}$) has two edges aligned with the eigenvectors $\boldsymbol{v}_i$ of $\mathcal{M}^{-1/2}$, in red. The 90° angles are preserved by $\mathcal{M}^{-1/2}$, but not the 45° angles. On the right, the reference triangles are not aligned with the $\boldsymbol{v}_i$ and the triangles mapped by $\mathcal{M}^{-1/2}$ have arbitrary Euclidean angles. The ellipse $\|u\|_{\mathcal{M}} = 1$ is in plain and the ellipse $\|u\|_{\mathcal{M}} = \sqrt{2}$ in dashed.

to the cosine of the required angles. For instance, take $u = (1, 0)$ and $v = (1/2, \sqrt{3}/2)$ forming a 60° angle in the reference space $(\mathbb{R}^2, \overline{\mathcal{M}})$. The mapped vectors $\mathcal{M}^{-\frac{1}{2}} R_0 u$ and $\mathcal{M}^{-\frac{1}{2}} R_0 v$ in the parameter space preserve the metric angle $\theta_{\mathcal{M}} = 60°$. The leftmost subfigure of Fig. 7.2 shows triangles associated to both u, v in the reference space and to their images in the metric space $(\mathbb{R}^2, \mathcal{M})$, as they are isometric and are seen as equilateral triangles in both spaces. The mapped tangent vectors will make an Euclidean angle of 90° if there is a rotation matrix $R_0 = R(\theta_0)$ such that:

$$\cos \theta_{\text{eucl}} = 0 = \frac{\langle \mathcal{M}^{-\frac{1}{2}} R_0 u, \mathcal{M}^{-\frac{1}{2}} R_0 v \rangle}{\|\mathcal{M}^{-\frac{1}{2}} R_0 u\| \|\mathcal{M}^{-\frac{1}{2}} R_0 v\|} \quad \Longrightarrow \quad u^T R_0^T \mathcal{M}^{-1} R_0 v = 0. \tag{7.2}$$

Depending on the metric, this may or may not have solutions. If this has a solution, then mapping the 6 directions by $\mathcal{M}^{-\frac{1}{2}} R_0$ yields triangles with Euclidean inner angles of 90° and 45°, see Fig. 7.2.

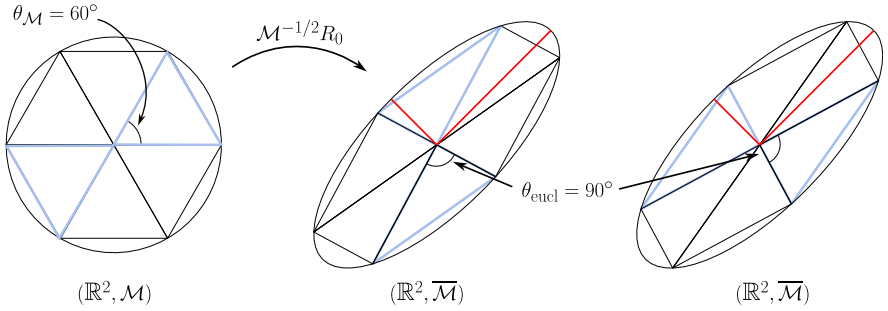


Figure 7.2 Angle control with the metric-aligned approach. For this particular metric and rotation R_0, the triangles mapped by $\mathcal{M}^{-1/2} R_0$ form right angles in the Euclidean space, but are not aligned with the eigenvectors.

Contrary to the metric-orthogonal approach, the mapped directions may not coincide with the metric eigenvectors (in red in Fig. 7.2), hence for a constant Riemannian metric such as the one of Fig. 7.2, it is possible to create a block-structured mesh that is not aligned with the principal directions of anisotropy, but whose angles are controlled for both $\mathcal{M}$ and the Euclidean metric, and is thus unit for both lengths and angles.

To create unit triangles for $K_\triangle$, we propose to start from the metric-aligned approach and explicitly control the inner angles with respect to $\mathcal{M}$. At a point $\boldsymbol{x}$ in the parameter space $(\mathbb{R}^2, \mathcal{M})$, 6 inner angles are defined in the tangent space with constant metric tensor $\mathcal{M}_{\boldsymbol{x}}$. Given a first direction u_0 (for instance one of the eigenvectors of $[\mathcal{M}_{\boldsymbol{x}}]$), the next direction v is chosen so that the Euclidean angle θ_{eucl} between them sets the dot product w.r.t. $\mathcal{M}$ to a target. For instance, to obtain 60° angles in the metric, θ_{eucl} satisfies:

$$\cos\theta_{\mathcal{M}_{\boldsymbol{x}}} = \frac{\langle u_0, v(\theta_{\text{eucl}})\rangle_{\mathcal{M}_{\boldsymbol{x}}}}{\|u_0\|_{\mathcal{M}}\|v(\theta_{\text{eucl}})\|_{\mathcal{M}}} = \frac{1}{2} \quad \Longrightarrow \quad \frac{\left(u_0^T[\mathcal{M}_{\boldsymbol{x}}]v(\theta_{\text{eucl}})\right)^2}{(u_0^T[\mathcal{M}_{\boldsymbol{x}}]u_0)(v^T(\theta_{\text{eucl}})[\mathcal{M}_{\boldsymbol{x}}]v(\theta_{\text{eucl}}))} = \frac{1}{4}. \quad (7.3)$$

From among the 4 exact solutions to the above, θ_{eucl} is the one that yields a positive dot product and keeps rotating in the same direction as the previously chosen angles. On the leftmost part of Fig. 7.3, u_0 is aligned with the smallest metric eigenvector $\lambda_2^{-1/2}\boldsymbol{v}_2$, and the 6 Euclidean angles obtained this way form metric angles $\theta_{\mathcal{M}} = 60°$. Alternatively, a global rotation R_0 can be determined so that $\mathcal{M}^{-1/2}R_0u = u_0 = \boldsymbol{v}_2$, then used to map the remaining 5 reference vectors forming equilateral triangles in the reference space.

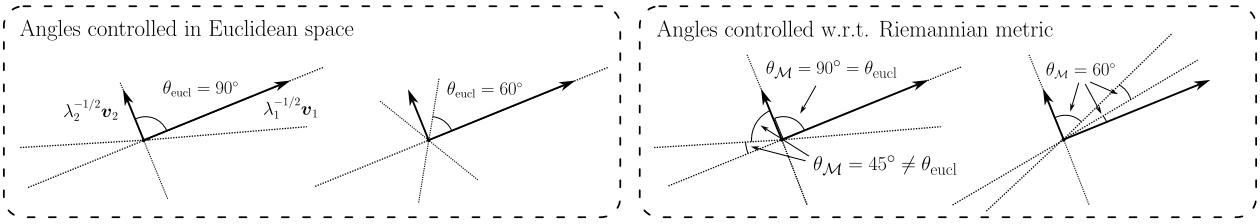


Figure 7.3 Control of the inner angles for the Euclidean metric $\overline{\mathcal{M}}$ (left) or for the input metric $\mathcal{M}$ (right).

Both the metric-orthogonal and metric-aligned approaches use the integral curves of the **eigenvectors** of the metric for alignment, whereas *QU2p* defines quasi-unit elements in terms of **geodesic** parameterizations. Although both trajectories are very similar for the presented test cases, we did not find nor established a formal link between eigenvectors and geodesics in general[1]. In the following, we compare the meshes obtained by spawning vertices

[1]Strictly speaking, eigenvectors are not defined for Riemannian metrics, which are covariant 2-tensors and

along both trajectories.

7.1.2 Vertex insertion

Following [21, 22, 41–43, 101], interior vertices are added to the empty mesh using a front of vertices. Unlike the usual frontal methods using a front of edges or faces, new vertices are proposed along trajectories originating from an active front vertex. This front is stored in a queue and initially consists of the boundary vertices constituting the empty mesh. The active vertex proposes $N = 4$ to 6 neighbours at unit distance along the chosen trajectories (eigenvectors integral curves or geodesics), depending on the choice of angles in the initial directions, then is removed from the queue. Different vertices will propose close or identical neighbours, so the vertices must be filtered to avoid creating short edges. In this work, as in [21, 22, 41], this is done by updating an RTree [63] with the accepted vertices and their metric. The accepted neighbours are then added to the front, and in turn propose new vertices until the queue empties. In the following, the creation of new candidate vertices and the filtering step are described. At the end of this operation, a cloud of points is obtained and is triangulated in the next section.

Integration along the chosen trajectories

Integrating along either the eigenvectors or geodesics requires solving an initial-value problem for the trajectory $\gamma(t)$. Let λ_i and $\boldsymbol{v}_i$ be the eigenvalues and eigenvectors of $[\mathcal{M}]$ and let $\boldsymbol{v}_{\min}$ be associated to $\lambda_{\max}^{-1/2} = h_{\min}$. N candidate vertices are obtained by integrating the chosen ODE with initial velocity $\boldsymbol{w} = h\boldsymbol{u}_j$, where $\boldsymbol{u}_j$ is unit and makes an angle with $\boldsymbol{v}_{\min}$ controlled for either $\mathcal{M}$ or the Euclidean metric, and h is the length from $\boldsymbol{x} = \gamma(t)$ to the unit ellipse in the direction $\boldsymbol{u}_j$. If $\boldsymbol{u}_j$ makes an angle θ with $\boldsymbol{v}_1$, this length is:

$$h(\theta) = \frac{h_1 h_2}{\sqrt{h_1^2 \sin^2\theta + h_2^2 \cos^2\theta}} \quad \text{with} \quad h_i = \lambda_i^{-\frac{1}{2}}. \tag{7.4}$$

For instance, with the metric-orthogonal approach, the $\boldsymbol{u}_j$ form a right-angle with each other for the Euclidean metric and the 4 initial velocities are simply the 4 eigendirections $\pm\lambda_j^{-\frac{1}{2}}\boldsymbol{v}_j$.

not linear transformations or $(1,1)$-tensors, but they can be used to determine the metric, as discussed in Chapter 3. Nonetheless, this is more subtle that it seems. For instance, in the hyperbolic plane $\{(x,y) \in \mathbb{R}^2 \mid y > 0\}$ characterized by the metric $[\mathcal{M}] = \mathrm{diag}(1/y^2, 1/y^2)$, the geodesics are vertical lines and half-circles centered on the x-axis, see e.g. Shifrin [98]. The horizontal lines have constant but nonzero geodesic curvature, thus are not geodesics. As the metric is diagonal, any two orthogonal vector fields can be taken to be its eigenvectors, but we cannot conclude that integrating along any set of eigenvectors yields geodesics. On the contrary, each point is the intersection of a vertical line with a half-circle, and these geodesics do not intersect orthogonally in general.

With the metric-aligned approach, the first velocity vector is set to $\boldsymbol{v}_{\min}$ and the remaining angles are controlled for the metric $\mathcal{M}$ and obtained by solving (7.3). Integrating along the **eigenvectors** thus yields the first-order ODE:

$$\begin{cases} \gamma(0) = \boldsymbol{x}_0, \\ \gamma'(t) = h(\gamma(t))\, \boldsymbol{u}_j(\gamma(t)) \triangleq \boldsymbol{w}(\gamma(t)), \end{cases} \tag{7.5}$$

whereas integrating the **geodesic** equation requires solving the second-order ODE:

$$\frac{d^2\gamma^k}{dt^2} + \frac{d\gamma^i}{dt}\frac{d\gamma^j}{dt}\Gamma^k_{ij}(\gamma(t)) = 0, \quad \gamma(0) = \boldsymbol{x}_0, \quad \gamma'(0) = h(\boldsymbol{x}_0)\boldsymbol{u}_j(\boldsymbol{x}_0). \tag{7.6}$$

The former only requires interpolating the background metric along the trajectories, whereas the Christoffel symbols in the geodesic equation require both the metric and its derivatives. The derivatives are more expensive to interpolate, thus compute the geodesics comes with an inherent cost increase. After recasting the geodesic equation into two first-order ODE, either (7.5) or (7.6) can be integrated with e.g. a standard RK4 method until a point $\boldsymbol{x}_j$ at unit distance from $\boldsymbol{x}_0$. This is shown in Fig. 7.4 for the hyperbolic tangent function and for the metric-orthogonal approach. Each dot in this figure is a neighbour proposed by the active vertex of the front.

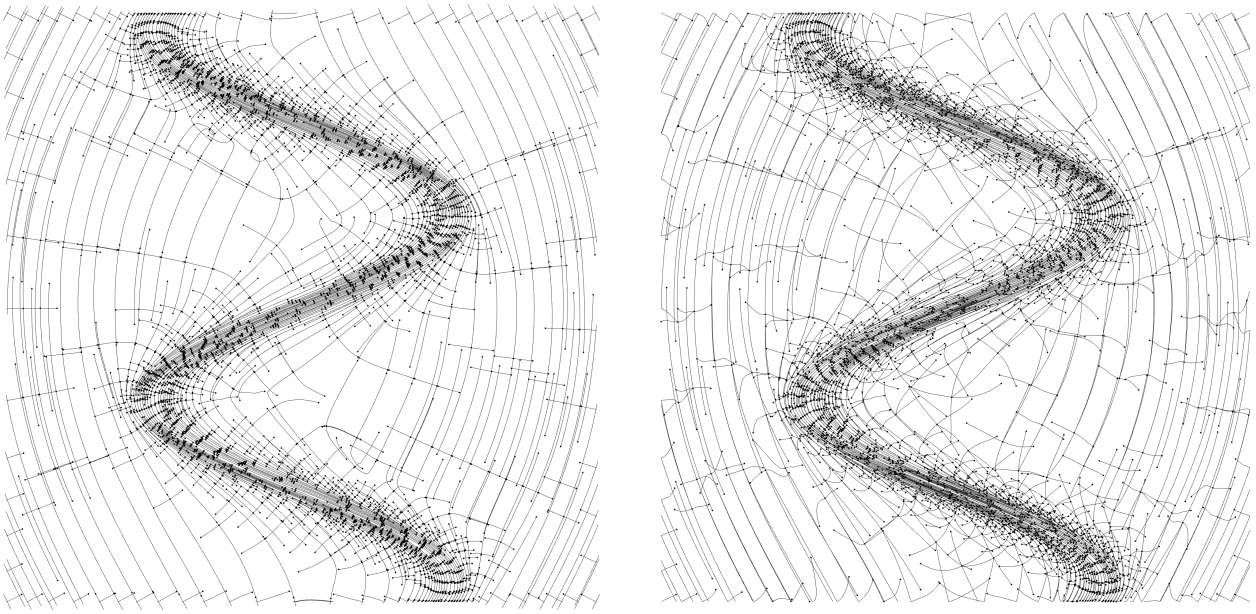


Figure 7.4 Integral curves of the eigenvector fields (left) and geodesics of the isocontour metric (right).

The right-hand side of the geodesic equation is always uniquely defined, but the one of (7.5)

is chosen from among N directions. To avoid sudden direction changes, the next velocity vector at each substep is taken as the one the most aligned with the current velocity, yielding the following scheme similar to [43]:

$$\begin{aligned}
\boldsymbol{x}_1 &= \boldsymbol{x}_n, & \boldsymbol{w}_1 &= h(\boldsymbol{x}_1)\boldsymbol{u}_j(\boldsymbol{x}_1), & & \\
\boldsymbol{x}_2 &= \boldsymbol{x}_n + \frac{\Delta t}{2}\boldsymbol{w}_1, & \boldsymbol{w}_2 &= h(\boldsymbol{x}_2)\boldsymbol{u}_{j_2}(\boldsymbol{x}_2), & j_2 &= \arg\max_{k=1,N} \boldsymbol{u}_k(\boldsymbol{x}_2)\cdot\boldsymbol{u}_j(\boldsymbol{x}_1), \\
\boldsymbol{x}_3 &= \boldsymbol{x}_n + \frac{\Delta t}{2}\boldsymbol{w}_2, & \boldsymbol{w}_3 &= h(\boldsymbol{x}_3)\boldsymbol{u}_{j_3}(\boldsymbol{x}_3), & j_3 &= \arg\max_{k=1,N} \boldsymbol{u}_k(\boldsymbol{x}_3)\cdot\boldsymbol{u}_{j_2}(\boldsymbol{x}_2), \\
\boldsymbol{x}_4 &= \boldsymbol{x}_n + \Delta t\boldsymbol{w}_3, & \boldsymbol{w}_4 &= h(\boldsymbol{x}_4)\boldsymbol{u}_{j_4}(\boldsymbol{x}_4), & j_4 &= \arg\max_{k=1,N} \boldsymbol{u}_k(\boldsymbol{x}_4)\cdot\boldsymbol{u}_{j_3}(\boldsymbol{x}_3), \\
\boldsymbol{x}_{n+1} &= \boldsymbol{x}_n + \frac{\Delta t}{6}(\boldsymbol{w}_1 + 2 & \boldsymbol{w}_2 &+ 2\boldsymbol{w}_3 + \boldsymbol{w}_4), & & \\
\ell &= \ell + d\ell, & d\ell &= \sqrt{(\boldsymbol{x}_n\boldsymbol{x}_{n+1})^T\,\mathcal{M}(\boldsymbol{x}_n)\,\boldsymbol{x}_n\boldsymbol{x}_{n+1}}. & &
\end{aligned} \tag{7.7}$$

This spawns a candidate vertex at unit distance and along the trajectory of the eigenvectors, but requires 4 metric interpolation per step to produce the velocity vectors $\boldsymbol{u}_k$. As metric queries are the costliest operations of the meshing procedure, a cheaper alternative such as the one-step Runge-Kutta-like scheme from [43] can be considered. A direction $\boldsymbol{x}_0\boldsymbol{x}_4$ is estimated following:

$$\begin{aligned}
\boldsymbol{x}_1 &= \boldsymbol{x}_0 + \frac{1}{4}h(\boldsymbol{x}_0)\boldsymbol{u}_j(\boldsymbol{x}_0), & & \\
\boldsymbol{x}_2 &= \boldsymbol{x}_1 + \frac{1}{4}h(\boldsymbol{x}_1)\boldsymbol{u}_{j_1}(\boldsymbol{x}_1), & j_1 &= \arg\max_{k=1,N} \boldsymbol{u}_k(\boldsymbol{x}_1)\cdot\boldsymbol{u}_j(\boldsymbol{x}_0), \\
\boldsymbol{x}_3 &= \boldsymbol{x}_2 + \frac{1}{4}h(\boldsymbol{x}_2)\boldsymbol{u}_{j_2}(\boldsymbol{x}_2), & j_2 &= \arg\max_{k=1,N} \boldsymbol{u}_k(\boldsymbol{x}_2)\cdot\boldsymbol{u}_{j_1}(\boldsymbol{x}_1), \\
\boldsymbol{x}_4 &= \boldsymbol{x}_3 + \frac{1}{4}h(\boldsymbol{x}_3)\boldsymbol{u}_{j_3}(\boldsymbol{x}_3), & j_3 &= \arg\max_{k=1,N} \boldsymbol{u}_k(\boldsymbol{x}_3)\cdot\boldsymbol{u}_{j_2}(\boldsymbol{x}_2),
\end{aligned} \tag{7.8}$$

then the candidate $\boldsymbol{x}_j$ is localized by dichotomy on the segment $\boldsymbol{x}_0\boldsymbol{x}_4$ so that:

$$\ell_{\mathcal{M}}(\boldsymbol{x}_0\boldsymbol{x}_j) = \int_0^1 \|\boldsymbol{x}_0\boldsymbol{x}_j\|_{\mathcal{M}}\,dt = \int_0^1 \sqrt{(\boldsymbol{x}_0\boldsymbol{x}_j)^T\,\mathcal{M}(\boldsymbol{x}_0 + t\boldsymbol{x}_0\boldsymbol{x}_j)\,\boldsymbol{x}_0\boldsymbol{x}_j}\,dt = 1. \tag{7.9}$$

The full integration is accurate, but is typically one order of magnitude more expensive, depending on the number of RK4 steps. This is the approach that was taken to obtain the prototype meshes of this chapter, however.

Filtering

As we aim at generating quasi-unit edges, vertices should not be allowed to create edges of length smaller than e.g. $\ell_{\min} = 1/\sqrt{2}$ if the chosen reference simplex is $K_0 = K_\triangle$. To this end, the N candidate vertices $\boldsymbol{x}_j$ proposed by the active vertex in the front are filtered based on their distance to the previously accepted vertices. A candidate is rejected if its distance to any vertex is smaller than $\ell_{\min}$. Fast neighbours detection is achieved with an RTree spatial search structure, storing the accepted vertices together with a rectangular box. Two vertices are considered neighbours if their boxes overlap, and distance checks are only carried out between such neighbours. This procedure is illustrated in Fig. 7.5. As always, identifying neighbours and computing lengths between them is a trade-off between speed and accuracy. Here, to reflect the exclusion zone of a vertex $\boldsymbol{x}$ and its metric, the box is the axis-aligned bounding box of the ellipse $E \equiv \|\boldsymbol{u}\|_{\mathcal{M}} = \sqrt{\boldsymbol{u}^T \mathcal{M} \boldsymbol{u}} = \ell_{\min}$, see the bottom right of Fig. 7.5. Because the Riemannian metric is not uniform in general, the sphere of radius $\ell_{\min}$ centered at $\boldsymbol{x}$ has an arbitrary shape and differs from this ellipse, see Section 3.4.1. However, estimating this sphere precisely is too expensive and E is its most straightforward, albeit crude, approximation. The bounding box of E is obtained from its extreme points with respect to the x, y-axes, obtained by cancelling the derivatives of the parameterized ellipse of semi-axes a, b making an angle ϕ with the x-axis:

$$\begin{cases} x(t) = x_0 + a\cos t\cos\phi - b\sin t\sin\phi \\ y(t) = y_0 + b\sin t\cos\phi + a\cos t\sin\phi \end{cases} \quad \Longrightarrow \quad \begin{cases} t_x^* = \operatorname{atan}\left({}^{b}\!/_{a}\,\tan\phi\right), \\ t_y^* = \operatorname{atan}\left(-{}^{b}\!/_{a}\operatorname{cotan}\phi\right), \end{cases} \tag{7.10}$$

The extreme points are $x^* = x(t_x^*)$, $y^* = y(t_y^*)$, see Fig. 7.5, or simply the semi-axes when the ellipse is aligned with either axis.

Before a new vertex $\boldsymbol{x}_j$ is inserted in the RTree, its metric-weighted distance to all other stored vertices $\boldsymbol{x}_s$ with overlapping bounding box is checked. As these vertices do not lie on a particular trajectory in general, the only natural curves between them are the straight line $\boldsymbol{x}_j\boldsymbol{x}_s$ (because they lie in the parameter space, a subset of $\mathbb{R}^2$) or a connecting geodesic. Computing the length of the straight line is clearly the most viable option as this check is performed a large number of times. Note that since the length of the straight line overestimates the Riemannian distance between the two vertices, i.e., $\ell_{\mathcal{M}}(\boldsymbol{x}_j\boldsymbol{x}_s) \geq d_{\mathcal{M}}(\boldsymbol{x}_j, \boldsymbol{x}_s)$, it is only conservative to reject a vertex, but never to accept it, as $\ell_{\mathcal{M}}(\boldsymbol{x}_j\boldsymbol{x}_s) \geq \ell_{\min} \not\Rightarrow d_{\mathcal{M}}(\boldsymbol{x}_j\boldsymbol{x}_s) \geq \ell_{\min}$.

Long edges are not controlled with this method. However, if the mesh is not too coarse and the metric is graded, enough vertices should be placed so that few long edges are cre-

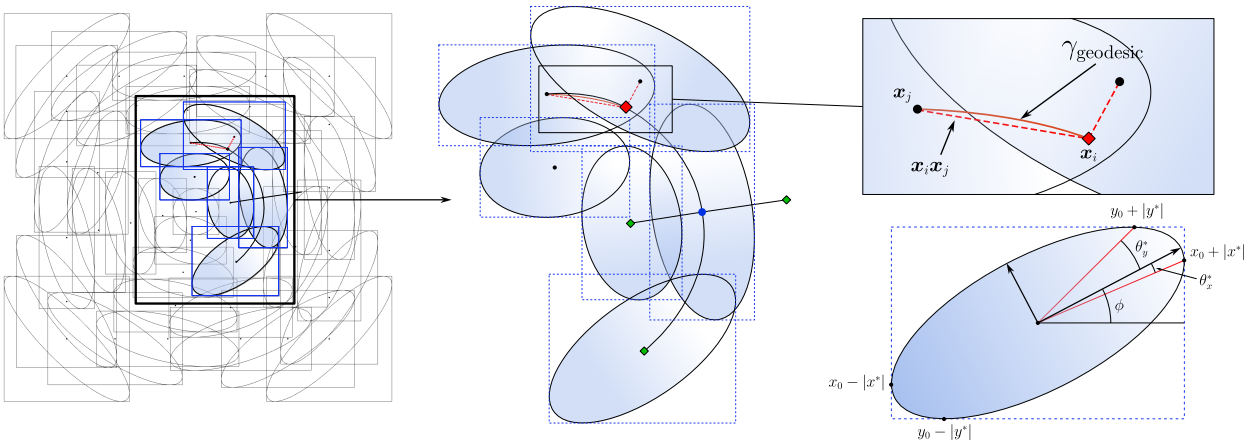


Figure 7.5 RTree for the insertion of new interior vertices. In this example, the blue vertex proposes 4 neighbours along the eigenvectors with the metric-orthogonal approach. The 3 green vertices were accepted and added to the tree with their metric. Before inserting the red vertex, distance checks are performed with other vertices whose ellipse bounding box intersects its own (not shown). The red vertex is rejected as it is too close to two existing vertices. Bottom right: computation of the axis-aligned bounding box of an ellipse.

ated. In Zhang et al. [22], long edges are eliminated with the so-called Curvilinear Small Polygon/Polyhedron Reconnection (CSPR), a curvilinear variant of the SPR cavity operator [105, 106], combined with the Growing Cavity proposed by Marot et al. [106]. Cavities are built around remaining long edges after curving, then all their possible curved triangulations are enumerated. The triangulation maximizing the minimum quality is selected and replaces the existing edges. Re-triangulating macrocavities this way allows to escape local quality maxima by reaching configurations not reachable with standard swaps. This formulation only accounts for quality, however, and not edge length explicitly, and indeed, the CSPR operator cannot remove all long edges in the curved mesh [22]. Moreover, the original SPR operator is appealing because computing the quality of linear elements for the Euclidean metric is very cheap, whereas the CSPR computes quality for a varying metric on $\mathcal{P}^2$ triangles which comes at higher cost. The addition of the CSPR, or a variant thereof, should be explored in the future to eliminate long edges in the final mesh and improve the present method.

7.1.3 Triangulation and optimization

Next, the cloud of vertices is triangulated to provide a valid and metric-conforming linear mesh. In [42, 43, 107], new vertices are inserted and triangulated on the fly within a cavity operator. This operator implements the anisotropic Delaunay kernel and performs standard

mesh operations, such as vertex insertions, edge collapses, edge swaps and, more recently, edge curving [24,32], in a unified framework. This cavity operator only creates valid elements and ensures the mesh is valid at all times during vertex insertion. As the meshing routines were implemented in Gmsh where there is no such cavity operator, we triangulate the point cloud with the standard isotropic Delaunay kernel, then swap the straight edges to improve conformity with the metric through a quality function. The vertices are not relocated as they were sampled at quasi-unit distance of one another. To guide the edge swaps, we maximize the minimum elementwise distortion-based quality Q_{K_0}, defined in (4.48) by:

$$Q_{K_0}(K) = \left(\frac{1}{|K_0|_{\overline{\mathcal{M}}}} \int_{K_0} \eta_{K_0}^n(\boldsymbol{y})\, d\boldsymbol{y} \right)^{-1/n} = n\, c_n |K_0|_{\overline{\mathcal{M}}}^{-2/n} \frac{|K|_{\mathcal{M}}^{2/n}}{\sum_{i=1}^{N_e} \ell_{\mathcal{M}}(e_i)^2} \in [0,1], \tag{7.11}$$

with n the dimension of space, $c_n = 3/2$ in 2D or 2 in 3D and e_i the edges of K. That is, an interior straight edge shared by two triangles K_1, K_2 is flipped to form the triangles K_1', K_2' if:

$$\min_{K'=K_1',K_2'} Q_{K_0}(K') > \min_{K=K_1,K_2} Q_{K_0}(K). \tag{7.12}$$

Since Q_{K_0} measures how far K deviates from a conformal transformation of K_0, the swaps aim at optimizing the straight mesh with respect to the inner angles for $\mathcal{M}$. Note that if the reference simplex K_0 is not equilateral, then the distortion depends on the ordering of the vertices through the Jacobian matrix J_K. This ordering must be accounted for to swap the edges consistently. Quality is improved by applying several passes of edge swaps. At each pass, only the edges touching a previously swapped triangles are tested. Edges are swapped this way until a stable configuration is found. These swaps are also performed as a final optimization step, to improve the $\mathcal{P}^2$ quality after curving to minimize the edge length.

The creation of the straight mesh for two analytical functions, following a metric-orthogonal point placement along the eigenvectors, is illustrated on Figs. 7.6 and 7.7: vertex sampling at controlled angles and distances starting from the empty mesh, connection with the isotropic Delaunay kernel and edge swaps. In regions where the elements should be curved, the apparent angles of the straight mesh are not 90° as vertices are connected by straight lines, while right angles are formed along the integral curves and will be recovered after curving. Linear meshes for 4 combinations of trajectories (eigenvectors versus geodesics) and angle control (metric-orthogonal vs metric-aligned) are shown in Fig. 7.8 for the hyperbolic tangent function. Quality indicators for all 4 combinations of approaches and for both $f = x^2+y^2$ and the hyperbolic tangent are presented in Figs. 7.9 and 7.10. For each mesh, the distribution of inner angles with respect to the metric $\theta_{\mathcal{M}}$ and to the Euclidean metric θ_{eucl} are presented,

together with the distortion-based quality $Q_{K_\triangle}$ and the Riemannian edge length $\ell\mathcal{M}(e)$. These histograms summarize the remarks made so far. They should be read together with the histograms of Figs. 7.12 through 7.15, showing the same indicators after curving the meshes. Specifically, the inner angles are better recovered after curving. In all histograms, red and blue bars are the initial unswapped and final swapped straight mesh, respectively, and green bars correspond to the final curved mesh. The blue bars in Figs. 7.9 and 7.10 and in Figs. 7.12 through 7.15 are the same set of data.

The metric-orthogonal approach (first and third rows) yields 45° and 90° angles w.r.t. $\mathcal{M}$ in proportion close to 2:1. This translates in a similar distribution of 90° angles in the Euclidean space, but 45° angles are mapped to arbitrary angles in $]0°, 90°[$. This illustrates the fact that 90° are preserved when the edges are aligned with both metric eigenvectors. The metric-aligned approach (second and fourth rows) creates a Gaussian-shaped distribution of metric angles centered around 60°, as expected. Those angles can be associated to very small or larger Euclidean angles. Slivers are not yet filtered in the meshing procedure. This translates in very obtuse Euclidean angles in very anisotropic regions, as seen in Figs. 7.14 and 7.15 for the hyperbolic tangent. Very small angles in these regions, however, are not a problem.

For all meshes, there are no short edges save for a few boundary edges, whose position is fixed by the empty mesh. A small proportion of long edges remains, however, as they were not explicitly treated. A majority of edges have almost exactly unit length. The metric-orthogonal approach yields right triangles for the metric, and a peak close to $\ell_{\mathcal{M}}(e) = \sqrt{2}$ can be observed. It also creates more long edges, as these edges aim a length of $\ell_0 = \sqrt{2}$ and can theoretically lie in $[1/\sqrt{2}\ell_0, \sqrt{2}\ell_0] = [1, 2]$. Conversely, the metric-aligned approach creates equilateral triangles with a higher concentration around $\ell_{\mathcal{M}}(e) = 1$.

Lastly, the quality of the swapped linear meshes is excellent and slightly improves after curving. Note that all meshes were improved for $Q_{K_\triangle}$, whereas metric-orthogonal sampling creates unit triangles for $K_\triangle$.

Aside from creating less right angles in the Euclidean space for geodesics, the indicators for following the eigenvectors (top two rows) or the geodesics (bottom two rows) are very similar. As they are cheaper to compute and yield comparable results, the subsequent meshes are computed along the **metric eigenvectors**, as in [21–23, 41].

7.2 Edge curving

Then, the linear mesh is curved to obtain quasi-unit $\mathcal{P}^2$ triangles in the sense of *QU1p* or *QU2p*. To this end, the $\mathcal{P}^1$ (corner) vertices are kept fixed and only the interior edges vertices

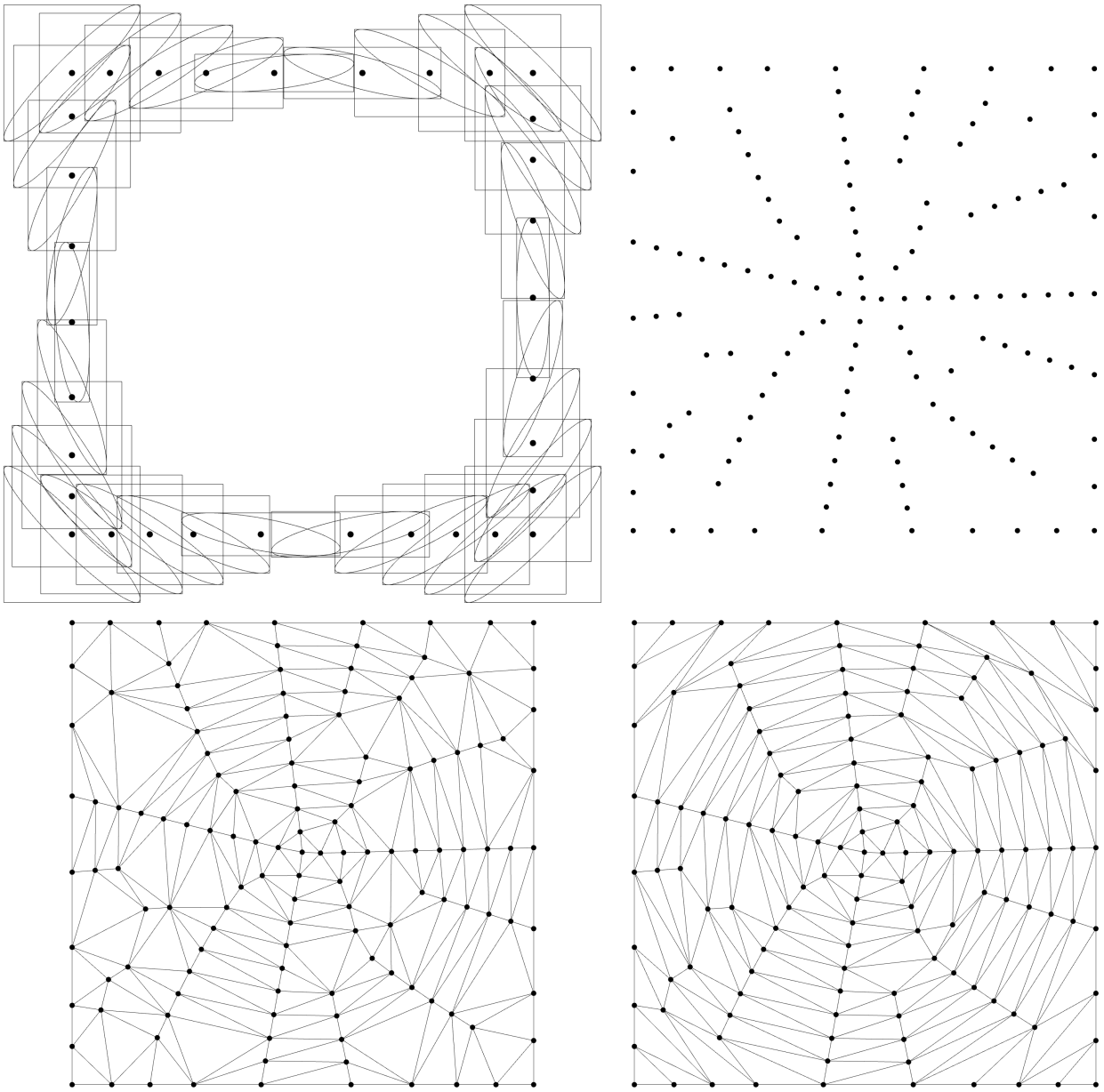


Figure 7.6 Generation of a metric-orthogonal linear mesh for $f = x^2 + y^2$ and the isocontour metric of Chapter 6. Top: boundary vertices with their ellipse satisfying $\|u\|^2_{\mathcal{M}} = u^T \mathcal{M} u = \ell^2_{\min}$ and bounding box, and final front of vertices. Bottom: initial isotropic Delaunay triangulation and triangulation after edge swaps to improve the distortion-based $\mathcal{P}^1$ quality.

are moved according to the chosen definition of quasi-unit. *QU1p* triangles have controlled quality whereas *QU2p* approximate geodesic triangles, hence edge curving is achieved by minimizing either the mesh distortion or the edge length w.r.t. the Riemannian metric, respectively. The former is a global optimization problem that was proposed by Aparicio-Estrems et al. [25–27]. It can be viewed as a *target matrix* approach with respect to a

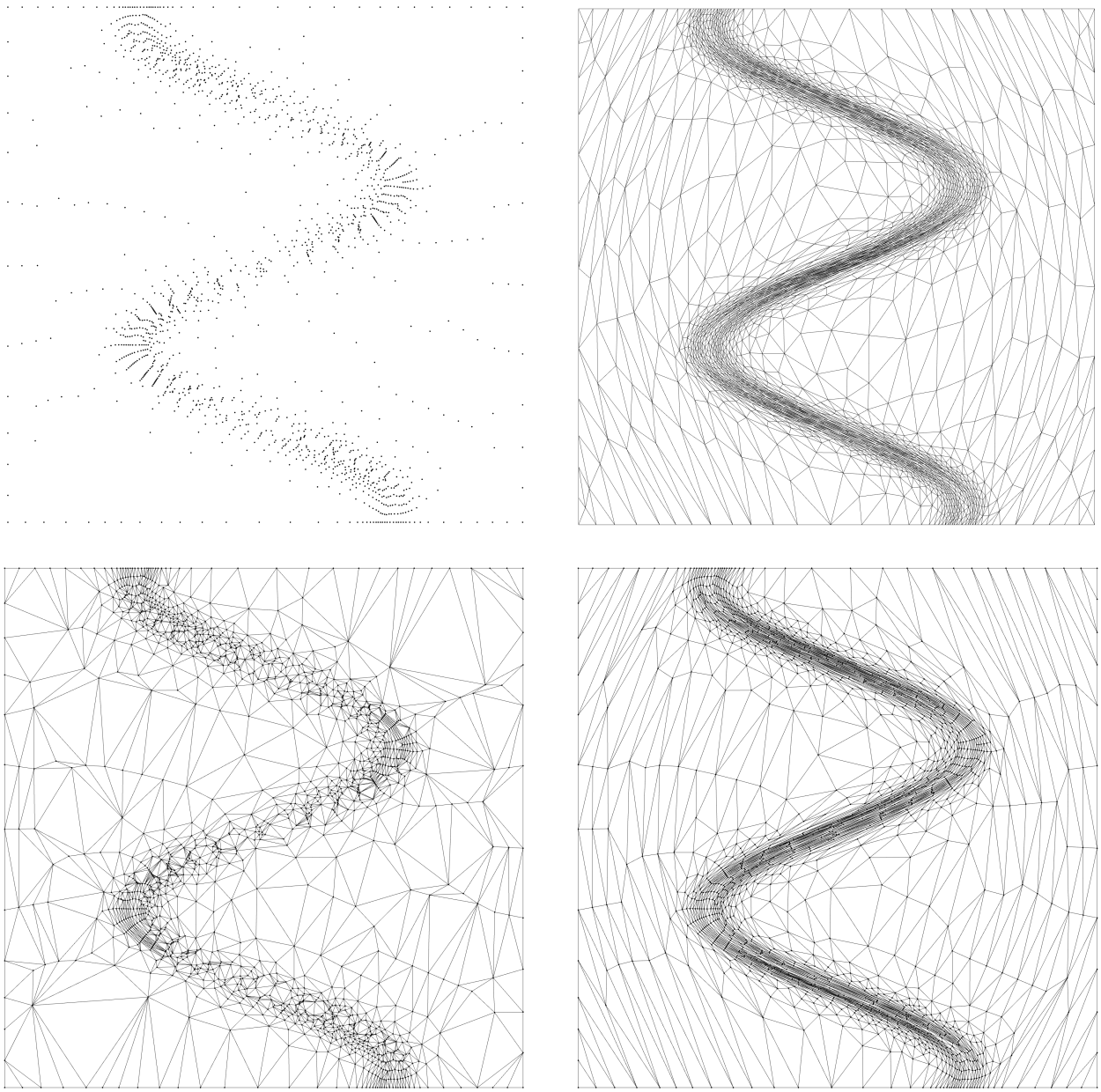


Figure 7.7 Generation of a metric-orthogonal linear mesh for $f = x^3 + x^2y + \tanh[10\,(2x - \sin 5y)]$ and the isocontour metric. Top left: sampled vertices. Bottom left: initial triangulation connected with the isotropic Delaunay kernel. Bottom right: swapped mesh. Comparison with standard quasi-unit linear mesh generated with `mmg2d` (top right).

Riemannian metric [108, 109]. The edge vertices are moved to minimize a nonlinear mesh distortion functional, which is regularized to detect invalid elements. In this work, the minimization is carried out on the nonregularized functional, with an additional smooth term enforcing validity as done in Toulorge et al. [16] and in Chapter 4. As an alternative, Rochery et al. [33] recently proposed a promising one-step curving method to enforce unitness with

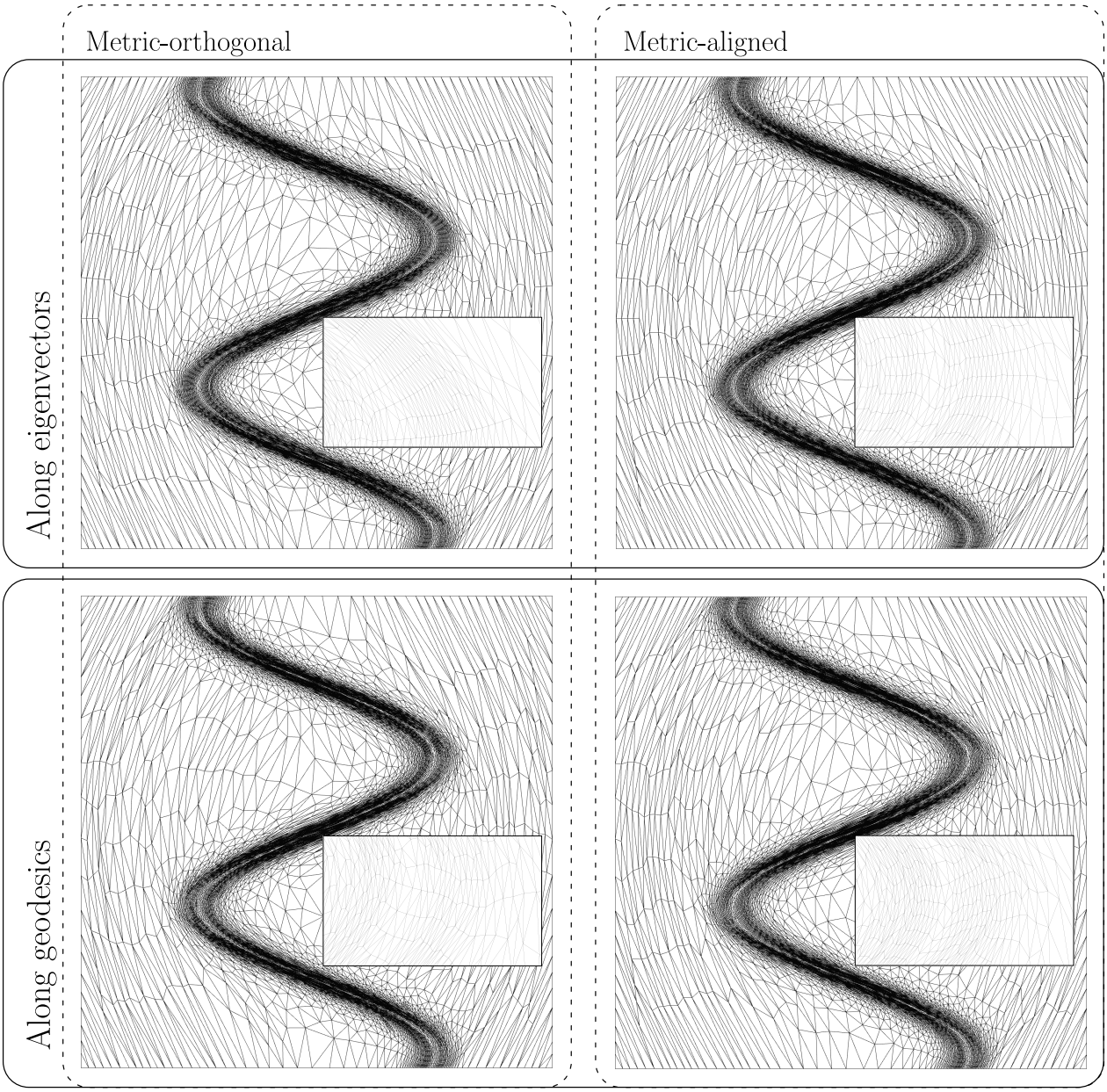


Figure 7.8 Initial linear meshes for the four considered approaches.

respect to the Jacobian matrix evaluated at the triangles' barycenters. Turning the global and expensive optimization problem into cheap elementwise operations makes this approach particularly appealing. It was not implemented in this thesis, but its ability to generate *QU1p* triangles should be investigated in the future.

The latter, edge length minimization, is a local problem formulated independently on each interior $\mathcal{P}^2$ edge. This is an n-parameters unconstrained optimization problem for each edge, which is fast and easily parallelizable. There is no a priori link between the geodesics and

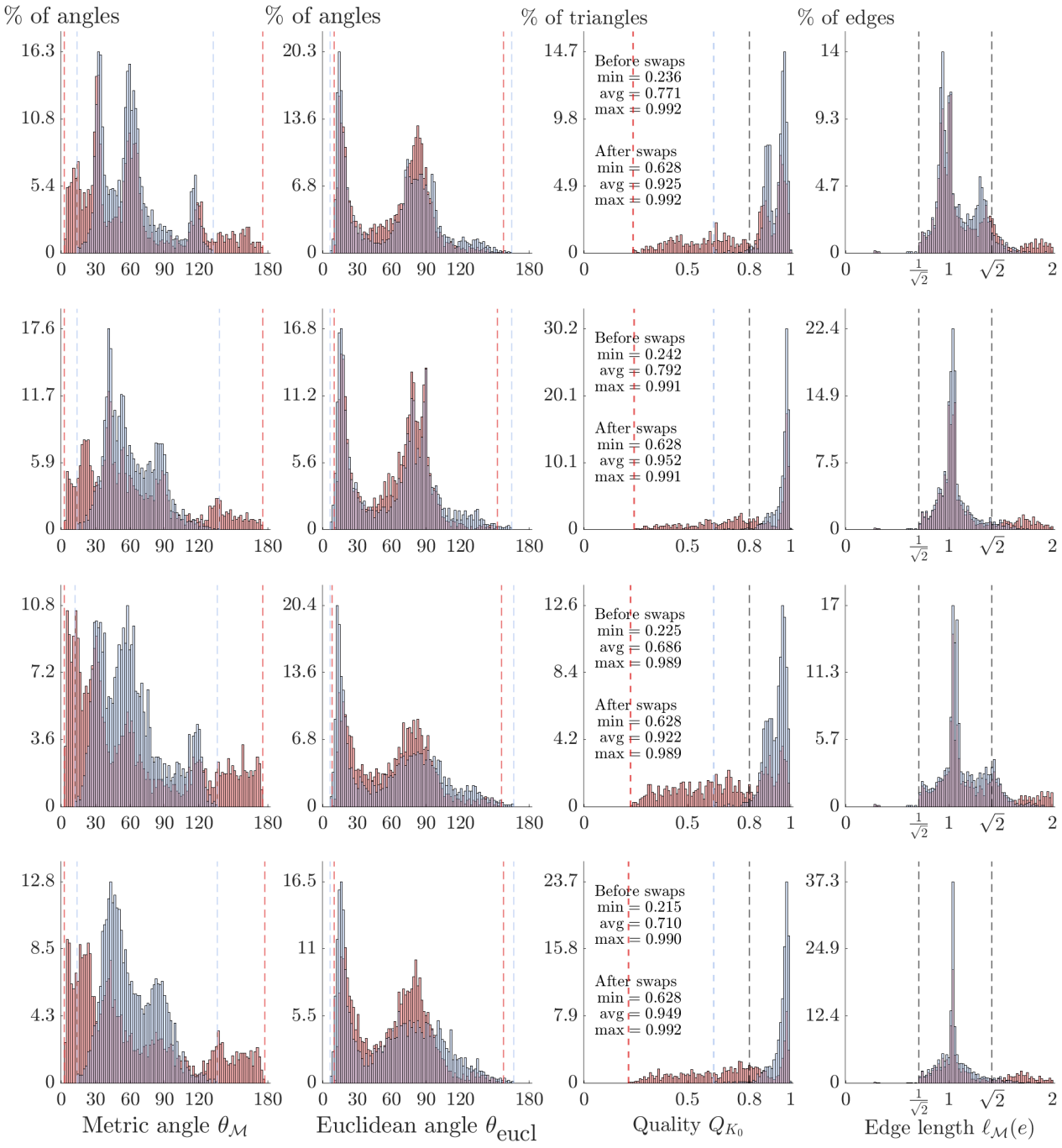


Figure 7.9 Quality indicators for the linear mesh adapted to $f = x^2 + y^2$ before (in red) and after (in blue) swapping the edges. Top two rows: vertices sampled along the eigenvectors. Bottom two rows: along the geodesics. Rows 1 and 3: metric-orthogonal approach. Rows 2 and 4: metric-aligned approach.

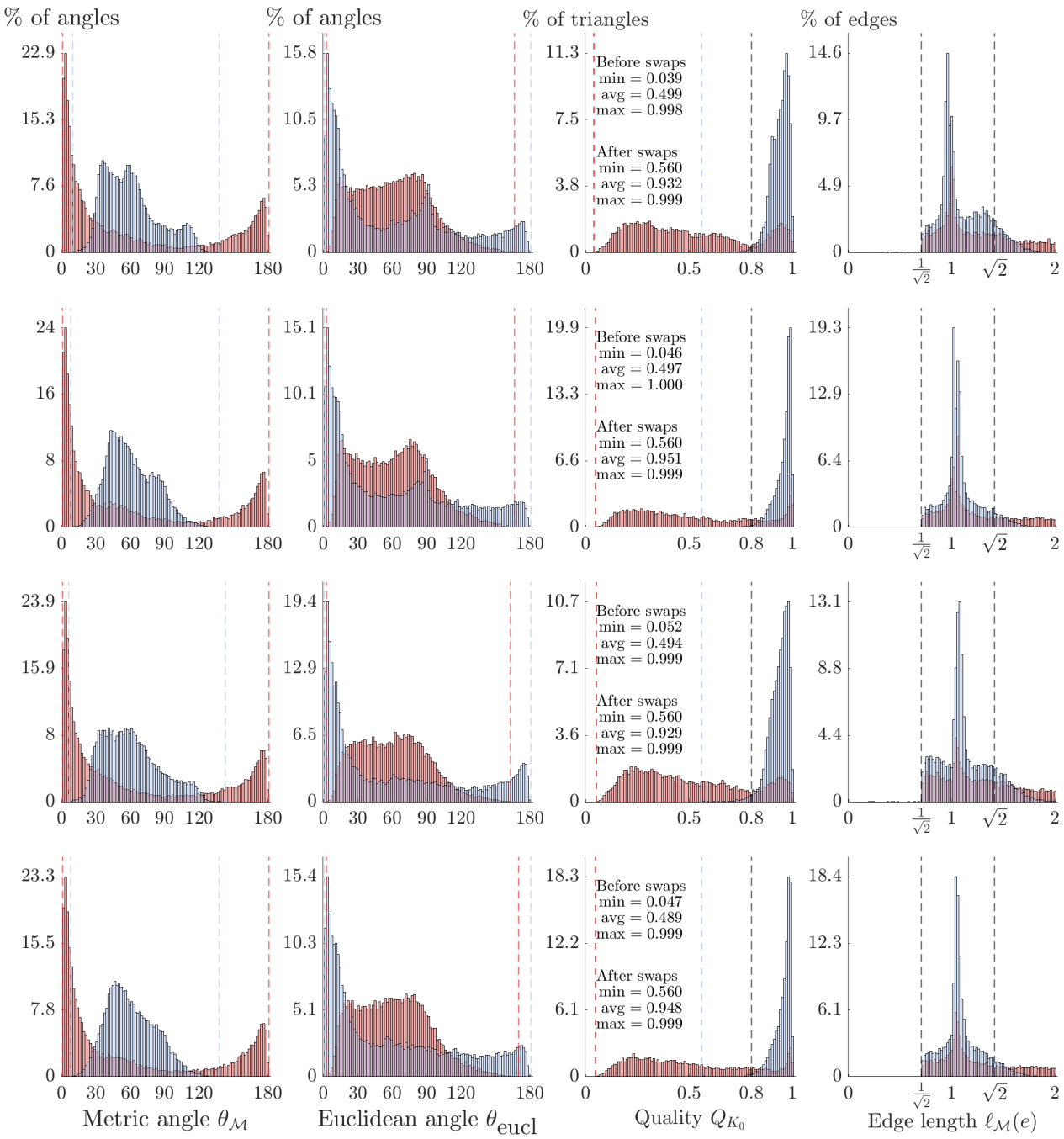


Figure 7.10 Quality indicators for the linear mesh adapted to $f = x^3 + x^2y + \tanh[10\,(2x - \sin 5y)]$ before (in red) and after (in blue) swapping the edges. Top two rows: vertices sampled along the eigenvectors. Bottom two rows: along the geodesics. Rows 1 and 3: metric-orthogonal approach. Rows 2 and 4: metric-aligned approach.

their $\mathcal{P}^2$ approximation, that is, placing the edge vertex at half the geodesic parameterization does not yield a length-minimizing parabola in general, as the optimal vertex is not even required to lie on the geodesic. Thus, minimization is always required to obtain the optimal edges. Because it is a natural and straightforward approach, length minimization has been the default curving method in most works on metric-based curvilinear mesh adaptation [21,23,24,29,41]. Note that unless the vertices are sampled along geodesic curves, one cannot both minimize and control the length within $[1/\sqrt{2}\ell(e_0), \sqrt{2}\ell(e_0)]$ in general, as short nonminimizing curves can become shorter than $1/\sqrt{2}\ell(e_0)$ after minimization.

Each minimization is detailed in this section and illustrative examples are shown. The quality indicators already presented for the linear meshes are shown after curving, and indicate that quasi-unit meshes are obtained.

7.2.1 Distortion minimization

We first recall the definition of the pointwise distortion w.r.t. a reference simplex K_0. Letting $J = J_{K_0 \to K}(\boldsymbol{y})$ and $\mathcal{M} = [\mathcal{M}_{F_{\triangle} \to K}(\boldsymbol{y})]$:

$$\eta(\boldsymbol{y}) = \frac{1}{n} \frac{\operatorname{tr}\left(J^T \mathcal{M} J\right)}{\left[\det\left(J^T \mathcal{M} J\right)\right]^{1/n}} \geq 1. \tag{7.13}$$

Distortion minimization was tackled in [25–27] by minimizing the *regularized* distortion functional:

$$\min_{\boldsymbol{x}_i} \sum_{K \in \mathcal{T}} \int_{K_0} \eta_0^2(\boldsymbol{y})\, d\boldsymbol{y}, \tag{7.14}$$

where the regularized pointwise distortion is defined as follows to detect inverted elements:

$$\eta_0(\boldsymbol{y}) = \frac{\operatorname{tr}\left(J^T \mathcal{M} J\right)}{n \sigma_0^{2/n}} \quad \text{with} \quad \sigma_0 = \frac{\sigma + |\sigma|}{2}, \quad \sigma = \det J \sqrt{\det \mathcal{M}}. \tag{7.15}$$

This function blows up when $\mathcal{J} = \det J \leq 0$. It varies smoothly for $\mathcal{J} > 0$ but is not differentiable at $\mathcal{J} = 0$. Here, similarly to Chapter 4, we minimize instead the smooth functional $E = \mathcal{E} + \mathcal{F}$, where $\mathcal{E}$ is the nonregularized distortion functional obtained by changing η_0 to η in (7.14), and $\mathcal{F}$ enforces the validity of the $\mathcal{P}^2$ triangles as described in Toulorge et al. [16]. It writes:

$$\mathcal{F} = \sum_{K \in \mathcal{T}} \sum_{j=1}^{6} F_\epsilon\left(\frac{N_j(\boldsymbol{x}_i)}{\mathcal{J}_0}\right) = \sum_{K \in \mathcal{T}} \sum_{j=1}^{6} \left[\ln\left(\frac{N_j(\boldsymbol{x}_i) - \epsilon \mathcal{J}_0}{\mathcal{J}_0 - \epsilon \mathcal{J}_0}\right)\right]^2 + \left(\frac{N_j(\boldsymbol{x}_i)}{\mathcal{J}_0} - 1\right)^2, \tag{7.16}$$

where F_ϵ is the log-barrier [] described in Chapter 4, the N_j are the 6 Bézier control coefficients described in Chapter 2 and such that the degree $n(q-1) = 2$ polynomial Jacobian determinant of the $\mathcal{P}^2$ triangle K writes $\mathcal{J}_K(\boldsymbol{\xi}) = \sum_j N_j B_j(\boldsymbol{\xi})$, and $\mathcal{J}_0$ is the Jacobian determinant of the underlying $\mathcal{P}^1$ triangle associated to K. The sum $\mathcal{F}$ blows up as soon as any control coefficients of the mesh becomes negative, which is a sufficient but not necessary condition for the associated determinant $\mathcal{J}_K$ to become negative. The objective function is thus overly conservative and forbids valid configurations with $\min N_j \leq 0$. In the following, we simply set $\epsilon = 0$ to ensure that all triangles remain valid during the minimization.

The minimization is performed with a quasi-Newton L-BFGS method via the library Ceres [54] by providing the exact gradient ∇E w.r.t. the edge vertices. The minimization terminates when either the relative decrease in the cost function, the maximum norm of its gradient or the change in the problem parameters $\|\Delta \boldsymbol{x}_i\|$ passes below the default threshold of Ceres, which are $10^{-6}, 10^{-10}$ and 10^{-8} respectively.

7.2.2 Length minimization

Recall the quadratic edge parameterization for $t \in [0, 1]$:

$$\begin{aligned} e_{12}(t) &= (1-t)(1-2t)\boldsymbol{x}_1 + 4t(1-t)\boldsymbol{x}_{12} + t(2t-1)\boldsymbol{x}_2, \\ e'_{12}(t) &= (4t-3)\boldsymbol{x}_1 + 4(1-2t)\boldsymbol{x}_{12} + (4t-1)\boldsymbol{x}_2. \end{aligned} \tag{7.17}$$

Each edge $e_{12}(t)$ is curved by moving the middle vertex to minimize the Riemannian length:

$$\min_{\boldsymbol{x}_{12}} \ell_{\mathcal{M}}(e_{12}) = \min_{\boldsymbol{x}_{12}} \int_0^1 \sqrt{(e'_{12}(t))^T \mathcal{M}(e_{12}(t)) e'_{12}(t)}\, dt. \tag{7.18}$$

Letting $e' = e'_{12}(t)$, the length gradient requires computing:

$$\frac{\partial}{\partial x^i_{12}} \left((e')^T \mathcal{M}(e)\, e' \right) = 2\, \frac{\partial e'}{\partial x^i_{12}} \mathcal{M} e' + (e')^T \frac{\partial \mathcal{M}(e)}{\partial x^i_{12}} e'. \tag{7.19}$$

The derivative of e' w.r.t. each component of the middle vertex is simply $4(1-2t)\boldsymbol{e}_i$. The metric derivative w.r.t. $\boldsymbol{x}_{12}$ writes:

$$\frac{\partial \mathcal{M}(e)}{\partial x^i_{12}} = \left.\frac{\partial \mathcal{M}}{\partial x^m}\right|_{e(t)} \frac{\partial e^m}{\partial x^i_{12}} = \left.\frac{\partial \mathcal{M}}{\partial x^m}\right|_{e(t)} 4t(1-t)\delta^m_i = \left.\frac{\partial \mathcal{M}}{\partial x^i}\right|_{e(t)} 4t(1-t). \tag{7.20}$$

The metric and its gradient are interpolated from the background mesh[2], following the pro-

[2]The current implementation interpolates the metric from the background mesh at each quadrature node,

cedure proposed by Aparicio-Estrems et al. [26] relying on the derivatives of the eigenvalues and eigenvectors of $\mathcal{M}$ and described in Section 3.6.1.

Contrary to the distortion functional, the validity of the elements is not accounted for while minimizing and must be enforced a posteriori. A simple backtracking procedure was applied: if a triangle is invalid after curving its edges, the curvature of each edge is linearly reduced along its displacement vector $\boldsymbol{a} = \boldsymbol{x}_{ij}\overline{\boldsymbol{x}}_{ij}$ with $\overline{\boldsymbol{x}}_{ij} = (\boldsymbol{x}_i + \boldsymbol{x}_j)/2$, until all of its Bézier control coefficients become positive.

Curved meshes for the running examples are shown in Fig. 7.11, and the quality indicators are presented in Figs. 7.12 through 7.15. These were commented in Section 7.1.3.

7.3 Illustrative examples

7.3.1 Lagrangian Coherent Structures

In fluid mechanics, Lagrangian coherent structures (LCS) are material separatrices revealing the dynamics of a flow, such as vortices, stagnation and separation/attachment lines (in 2D flows) and surfaces (in 3D). They were introduced by Haller [110, 111] to describe fluid flows through the theory of dynamic systems. LCS may be described as $n-1$-dimensional manifolds locally maximizing (i.e., the *ridges* of) a scalar field called the finite-time Lyapunov exponent (FTLE), a measure of the maximum growth rate of the distance between two initially close fluid particles over a finite time interval. These manifolds have locally the strongest influence on nearby trajectories and help visualize the flow structures. Depending on the Eulerian velocity field, these ridges can be curved and/or exhibit a pronounced anisotropy, making them prime candidates for anisotropic mesh adaptation, see for instance Miron et al. [5] and Fortin et al. [6, 37] for examples of straight-sided adaptation with and without Riemannian metrics respectively. We briefly describe the computation of the FTLE field and of the Hessian matrix of the *flow map* driving the adaptation, and refer to these works for more details. Given a velocity field $u(\boldsymbol{x}, t)$ and a time interval $[t_0, t_0+T]$ with positive or negative T, a Lagrangian particle initially at some $\boldsymbol{x}_0$ follows a trajectory determined by $\boldsymbol{x}'(t) = u(\boldsymbol{x}, t)$. The position $\boldsymbol{x}(t)$ is the current configuration from continuum mechanics, and the **flow map** is defined in [5, 6, 37] as $\boldsymbol{x}(\boldsymbol{x}_0, T)$, the final position of the particle. The evolution of this particle at $t_0 + T$ is reflected by the right Cauchy-Green strain tensor $\boldsymbol{C} = (\nabla \boldsymbol{x}(\boldsymbol{x}_0, T))^T \cdot (\nabla \boldsymbol{x}(\boldsymbol{x}_0, T))$, where the gradient is taken w.r.t. the position $\boldsymbol{X} = \boldsymbol{x}_0$ in the reference configuration. In

but it would be faster to interpolate only at $\boldsymbol{x}_{12}$, then use quadratic log-Euclidean interpolations along the edge to evaluate $\mathcal{M}(e_{12}(t))$, see Rochery [17].

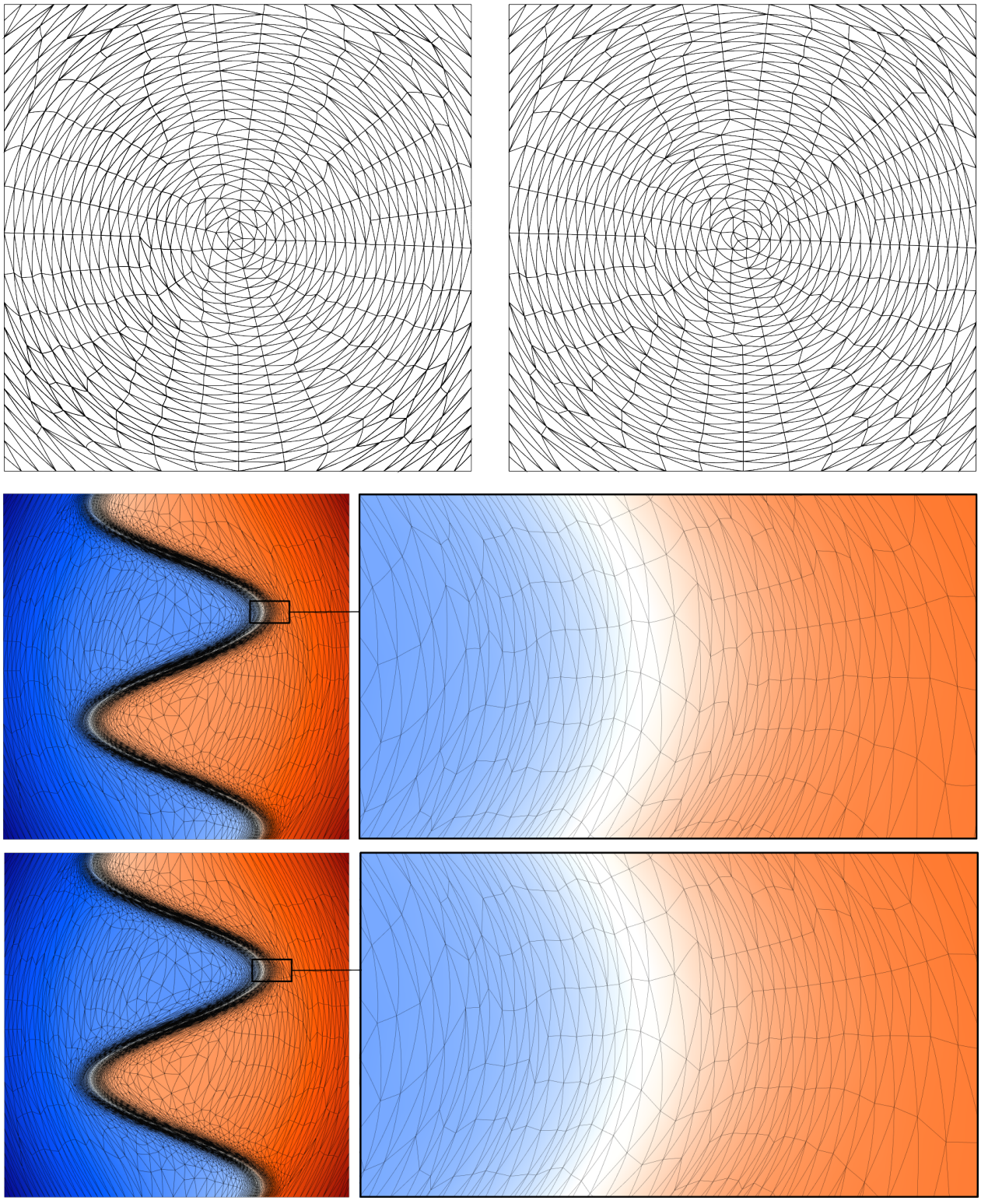


Figure 7.11 Curved meshes for $f = x^2 + y^2$ and $f = x^3 + x^2y + \tanh[10\,(2x - \sin 5y)]$ to minimize the metric distortion (top left and middle) and the edge length (top right and bottom). The vertices are sampled along the eigenvectors of the isocontour metric $\mathcal{M}_{\text{iso}}$ and form angles close to 60° with respect to $\mathcal{M}_{\text{iso}}$ (metric-aligned approach).

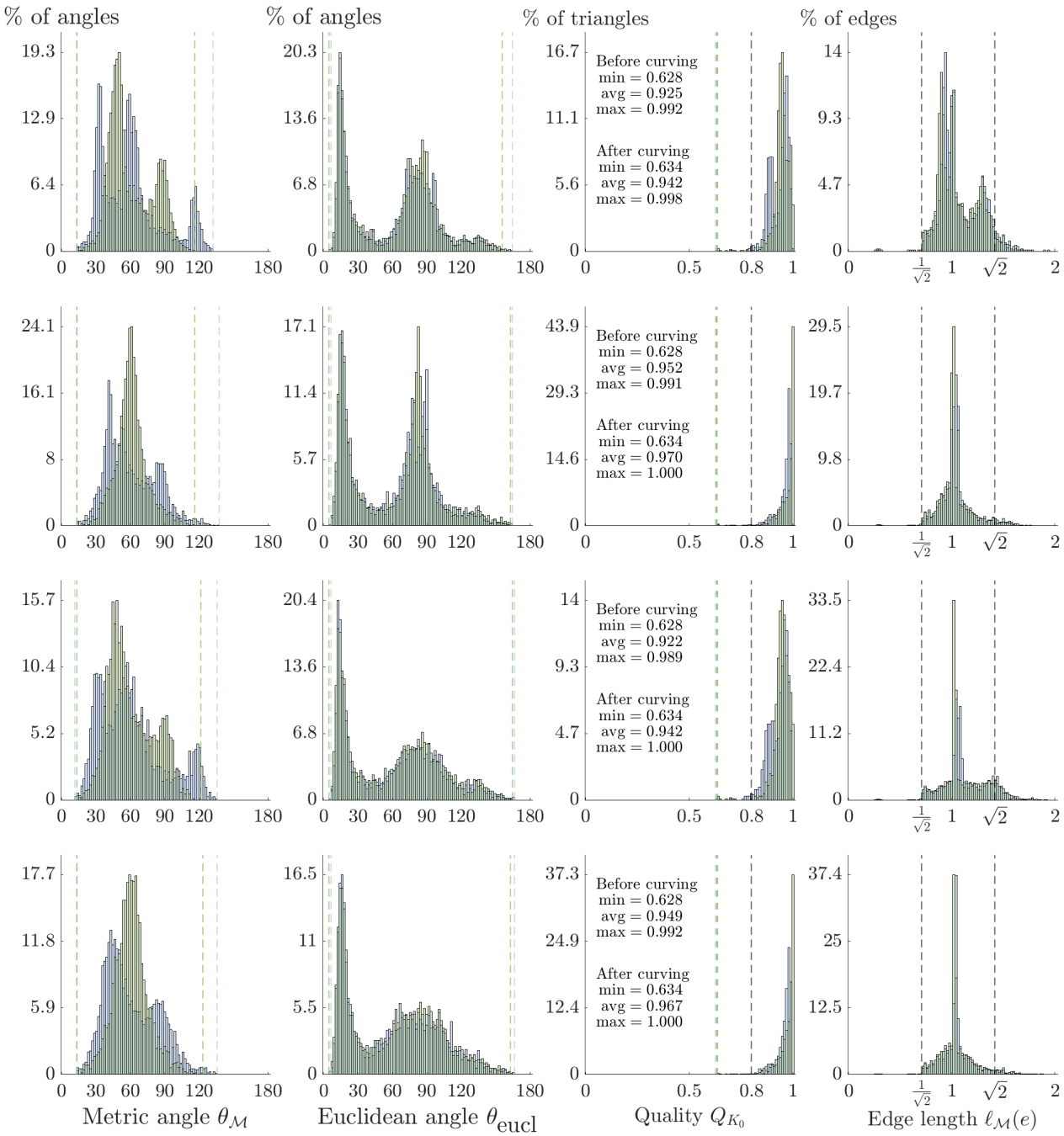


Figure 7.12 Quality indicators for $f = x^2 + y^2$ before (in blue) and after (in green) curving to minimize the distortion. Top two rows: vertices sampled along the eigenvectors. Bottom two rows: along the geodesics. Rows 1 and 3: metric-orthogonal approach. Rows 2 and 4: metric-aligned approach.

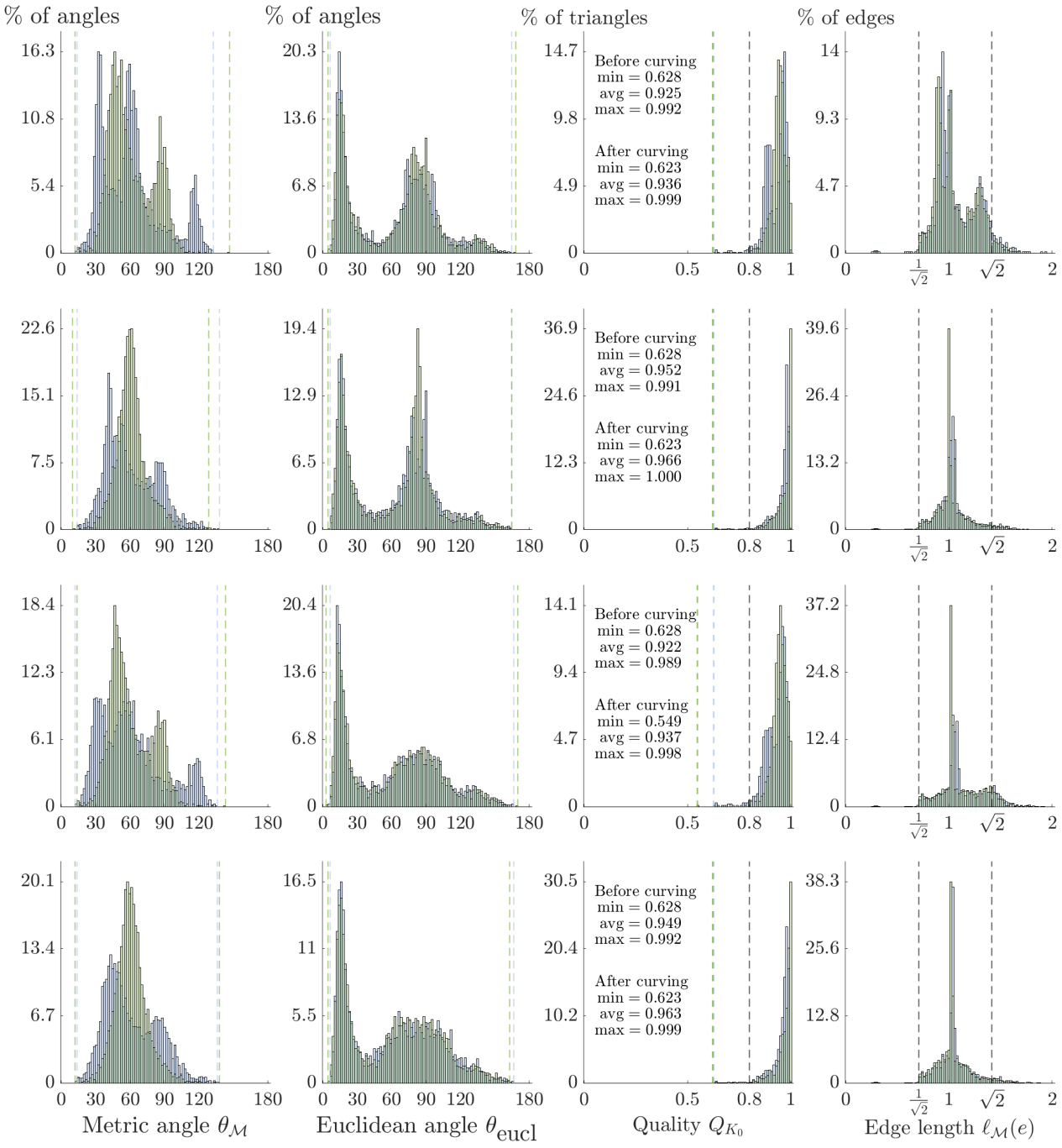


Figure 7.13 Quality indicators for $f = x^2 + y^2$ before (in blue) and after (in green) curving to minimize the edge length. Top two rows: vertices sampled along the eigenvectors. Bottom two rows: along the geodesics. Rows 1 and 3: metric-orthogonal approach. Rows 2 and 4: metric-aligned approach.

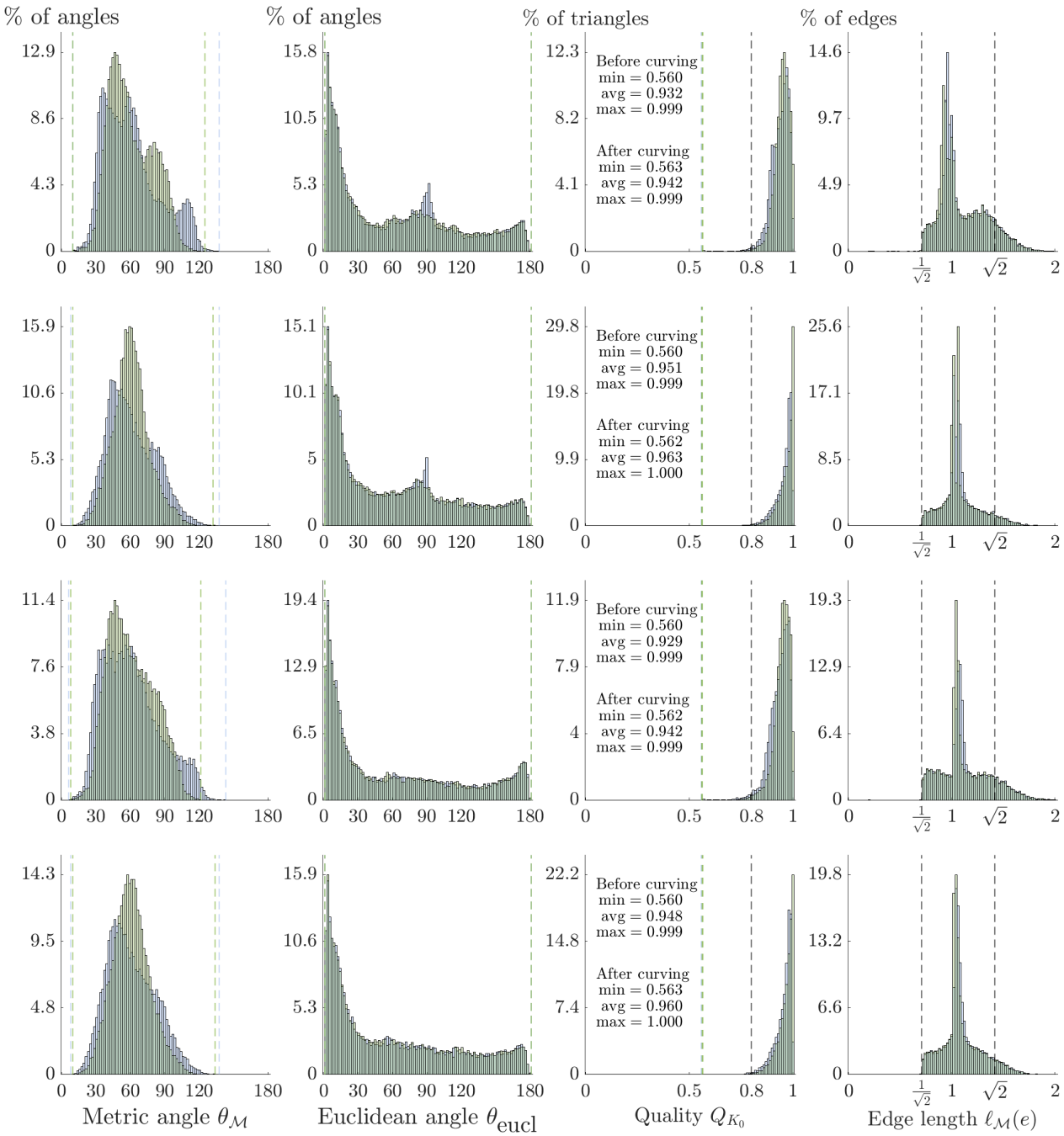


Figure 7.14 Quality indicators for $f = x^3 + x^2y + \tanh[10\,(2x - \sin 5y)]$ before (in blue) and after (in green) curving to minimize the distortion. Top two rows: vertices sampled along the eigenvectors. Bottom two rows: along the geodesics. Rows 1 and 3: metric-orthogonal approach. Rows 2 and 4: metric-aligned approach.

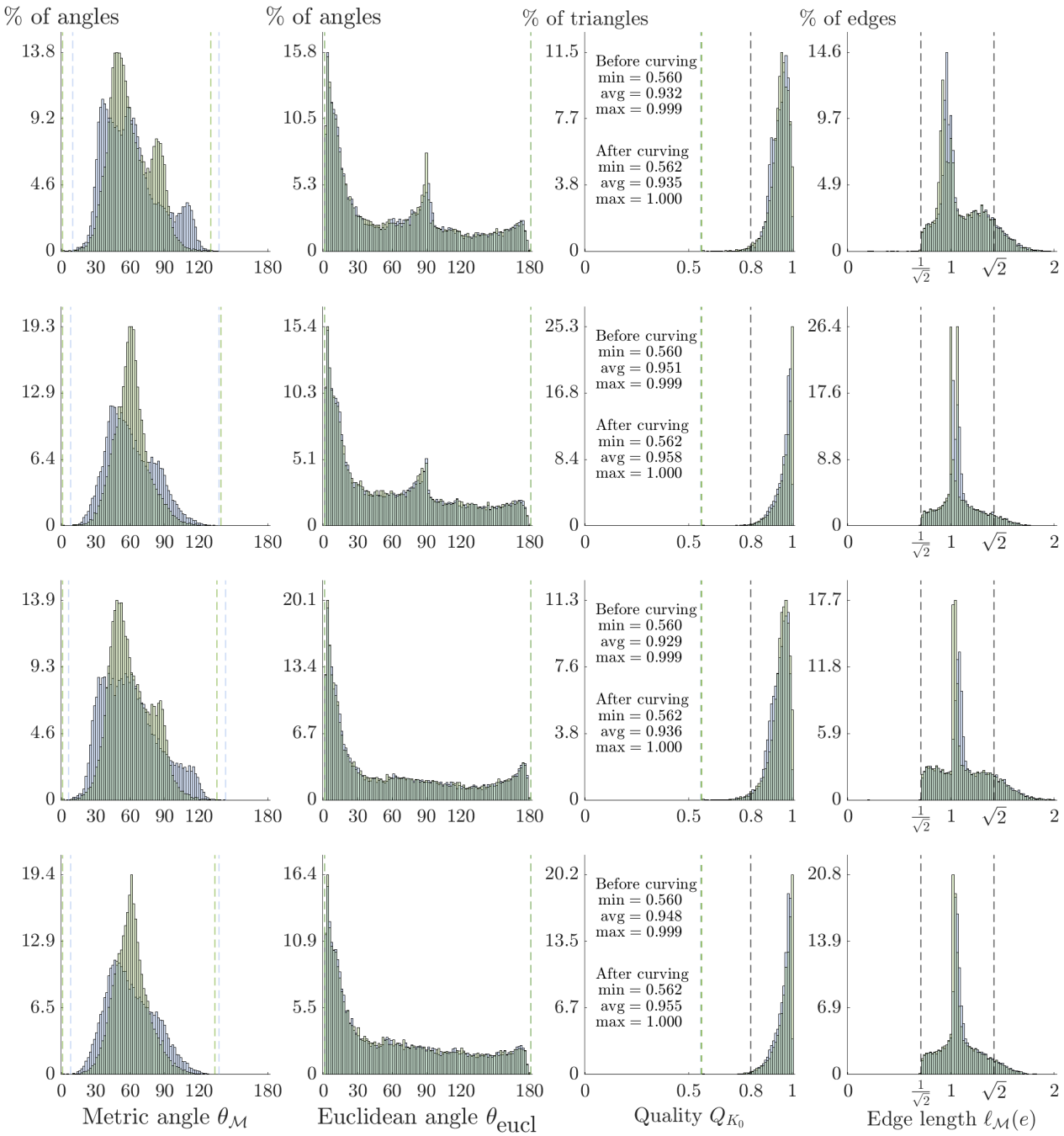


Figure 7.15 Quality indicators for $f = x^3 + x^2y + \tanh[10\,(2x - \sin 5y)]$ before (in blue) and after (in green) curving to minimize the edge length. Top two rows: vertices sampled along the eigenvectors. Bottom two rows: along the geodesics. Rows 1 and 3: metric-orthogonal approach. Rows 2 and 4: metric-aligned approach.

particular, the maximum linear approximation of the attraction or separation between two initially close particles is related to the largest eigenvalue $\lambda_{\max}$ of $\boldsymbol{C}$, yielding the following definition of the Lyapunov exponent:

$$\sigma_{\mathrm{FTLE}}(\boldsymbol{x}_0) \triangleq \frac{1}{2|T|} \ln\left(\lambda_{\max}(\boldsymbol{x}_0)\right). \tag{7.21}$$

Instead of adapting with respect to the FTLE, the works above showed that the derivatives of the flow map can be used to drive the adaptation while accurately identifying the LCS: the Hessian of the flow map is used in [5] to define the optimal metric for linear interpolation from Alauzet and Loseille [8, 9], while [6, 37] use only the gradient to derive local L^2 and H^1 error estimates controlling the mesh refinement and edge swaps. The advantage of these approaches is that the exact derivatives of the flow map are obtained by taking the gradient of $\boldsymbol{x}'(t) = u(\boldsymbol{x}, t)$ w.r.t. $\boldsymbol{X}$, whereas the derivatives of the FTLE field would typically require recovery methods. Thus, adapting on the flow map allows to assess the creation of quasi-unit curvilinear triangulations on more complex but still analytical fields.

From the trajectory ODE, the gradient of the final component x^i satisfies the ODE:

$$\frac{d}{dt}\left(\frac{\partial x^i}{\partial X^j}\right) \triangleq \frac{d}{dt} G^i_j = \left.\frac{\partial u^i}{\partial x^m}\right|_{(\boldsymbol{x},t)} \left.\frac{\partial x^m}{\partial X^j}\right|_t = \left.\frac{\partial u^i}{\partial x^m}\right|_{(\boldsymbol{x},t)} G^m_j(t), \tag{7.22}$$

and its Hessian matrix satisfies:

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial^2 x^i}{\partial X^j \partial X^k}\right) \triangleq \frac{d}{dt} H^i_{jk} &= \left.\frac{\partial^2 u^i}{\partial x^m \partial x^n}\right|_{(\boldsymbol{x},t)} \left.\frac{\partial x^m}{\partial X^j}\right|_t \left.\frac{\partial x^n}{\partial X^k}\right|_t + \left.\frac{\partial u^i}{\partial x^m}\right|_{(\boldsymbol{x},t)} \left.\frac{\partial^2 x^m}{\partial X^j \partial X^k}\right|_t \\ &= \left.\frac{\partial^2 u^i}{\partial x^m \partial x^n}\right|_{(\boldsymbol{x},t)} G^m_j(t) G^n_k(t) + \left.\frac{\partial u^i}{\partial x^m}\right|_{(\boldsymbol{x},t)} H^m_{jk}(t).\end{aligned} \tag{7.23}$$

Here, these $2 + 4 + 6 = 12$ ODE (in 2D), together with the initial conditions $\boldsymbol{x}(0) = \boldsymbol{x}_0, G^i_j(0) = \delta^i_j$ and $H^i_{jk}(0) = 0$, are integrated from $t_0 = 0.5$ to $t_{\mathrm{end}} = 1.5$ with a standard RK4 scheme. This provides a precise approximation of the analytical gradient and Hessian of the flow maps, driving the adaptation. We consider, as in [5,6,37], Solomon and Gollub's velocity field describing periodic Rayleigh-Bénard convection cells. This velocity $\boldsymbol{u}(\boldsymbol{x}, t) = \nabla \times \psi \boldsymbol{e}_z$ derives from the stream function:

$$\psi(\boldsymbol{x}, t) = \sin(\pi(x - g(t))) \sin(\pi y), \qquad \text{with} \qquad g(t) = 0.3 \sin(4t) + 0.1 \sin(2t). \tag{7.24}$$

A curved mesh adapted to the flow map is shown in Fig. 7.16 and Fig. 7.17 summarizes its quality indicators. This mesh minimizes the Riemannian edge length with respect to the

isocontour metric $\mathcal{M}_{\text{iso}}$, with a metric-aligned vertex sampling along the metric eigenvectors. The mesh is accurately curved where necessary and exhibits high ratios of anisotropy along the ridges of the FTLE field. The metric angle distribution is centered around 60° w.r.t. $\mathcal{M}_{\text{iso}}$, yielding in this case almost 30% of right angles in the Euclidean space, and 98% of the triangles have a distortion-based $\mathcal{P}^2$ quality in $[0.8, 1]$. Since the vertices are sampled along the eigenvectors, there is no short edge after swapping the linear mesh, but a few (0.48%) remain after curving. Long edges remain (6%) in the final mesh as they are not yet explicitly treated.

7.3.2 Lid-driven cavity flow

Finally, we consider the academic incompressible flow inside a lid-driven square cavity. This illlustrates quasi-unit triangulations adapted to a CFD application. As in Chapter 5, the nondimensional steady incompressible Navier-Stokes equations are solved with our in-house finite element solver:

$$\begin{aligned} \nabla \cdot \boldsymbol{u} &= 0, \\ (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \frac{1}{Re}\nabla^2 \boldsymbol{u} \quad &= \mathbf{0}. \end{aligned} \tag{7.25}$$

A no-slip boundary condition is imposed on all but the top boundary, for which we set $\boldsymbol{u} = (1, 0)$, and the lower left corner has a reference pressure $p = 0$. For this example, we use SUPG-PSPG stabilized $\Pi^1\Pi^1$ finite elements to discretize the $(\boldsymbol{u}, p)$ pair. The initial mesh is a coarse uniform discretization of the square, on which a low Reynolds ($Re = 100$) solution is first computed. The Reynolds number is increased progressively while adapting the straight mesh every few steps, so that the background mesh is an adapted anisotropic mesh with a converged solution at $Re = 1000$. We use once again the isocontour metric $\mathcal{M}^{\text{iso}}$ to control the linear interpolation error w.r.t. the 2-norm of the velocity $\|\boldsymbol{u}\|$ and with a target edge error $\epsilon = 10^{-3}$. The vertices are sampled with the metric-aligned approach along the metric eigenvectors, to create unit triangles w.r.t. $K_{\triangle}$. This yields a $\mathcal{P}^2$ mesh with 12,655 vertices and 6,250 triangles, Fig. 7.18. The mesh follows the features of $\|\boldsymbol{u}\|$, with appreciable anisotropy on the lid and where the fluid hits the right boundary, and curves to follow the main recirculation cell. The mesh quality indicators are shown in Fig. 7.19: the metric inner angles are centered around 60° as expected and the vast majority of edges are unit. Similarly, there are a few outlier low-quality triangles, but about 96% of the elements have sufficient quality, showing that the triangulation is quasi-unit for this metric.

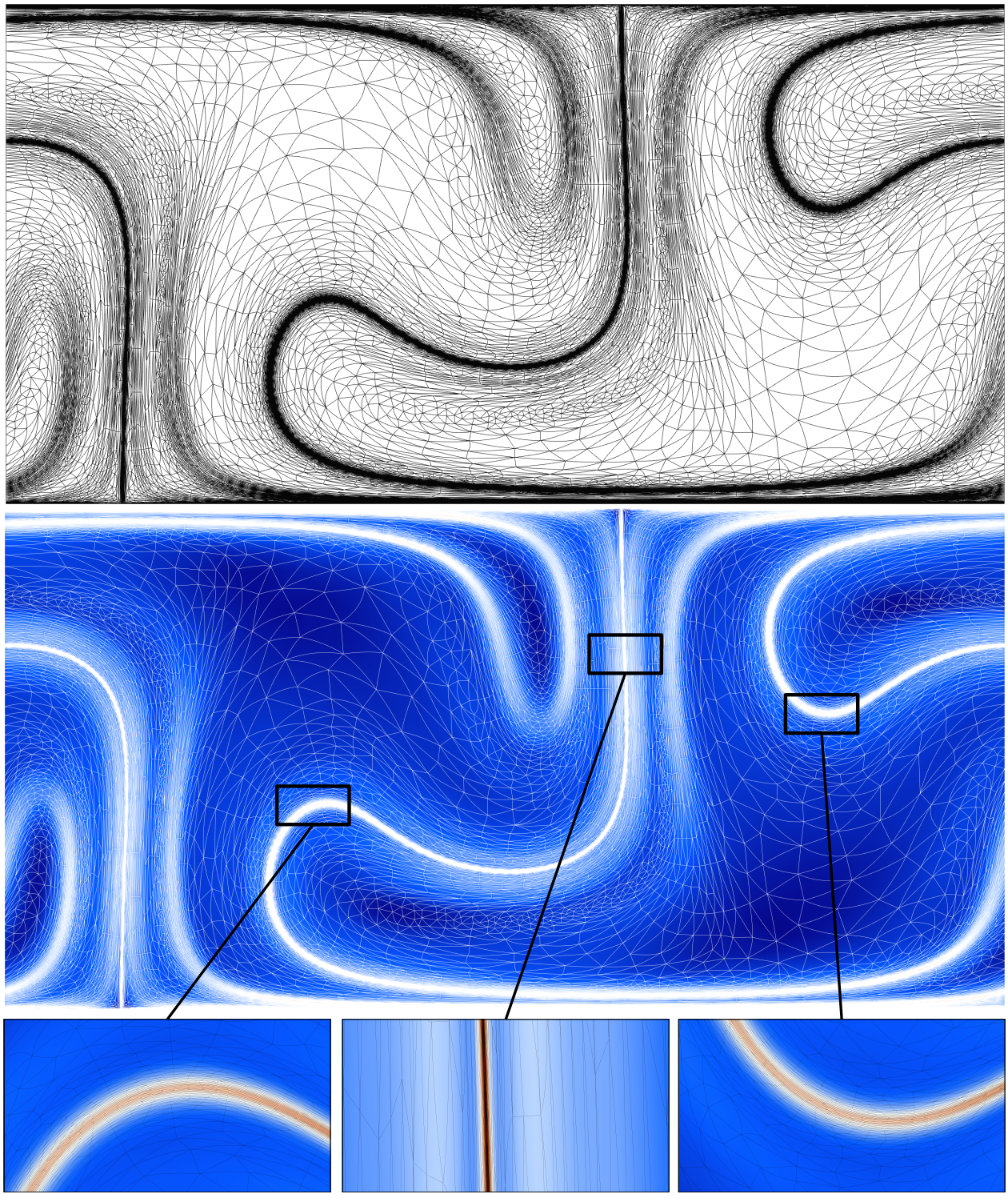


Figure 7.16 FTLE field and $\mathcal{P}^2$ mesh (83,043 vertices and 41,258 triangles) adapted to Solomon and Gollub's flow map and minimizing the Riemannian edge length w.r.t. the isocontour metric.

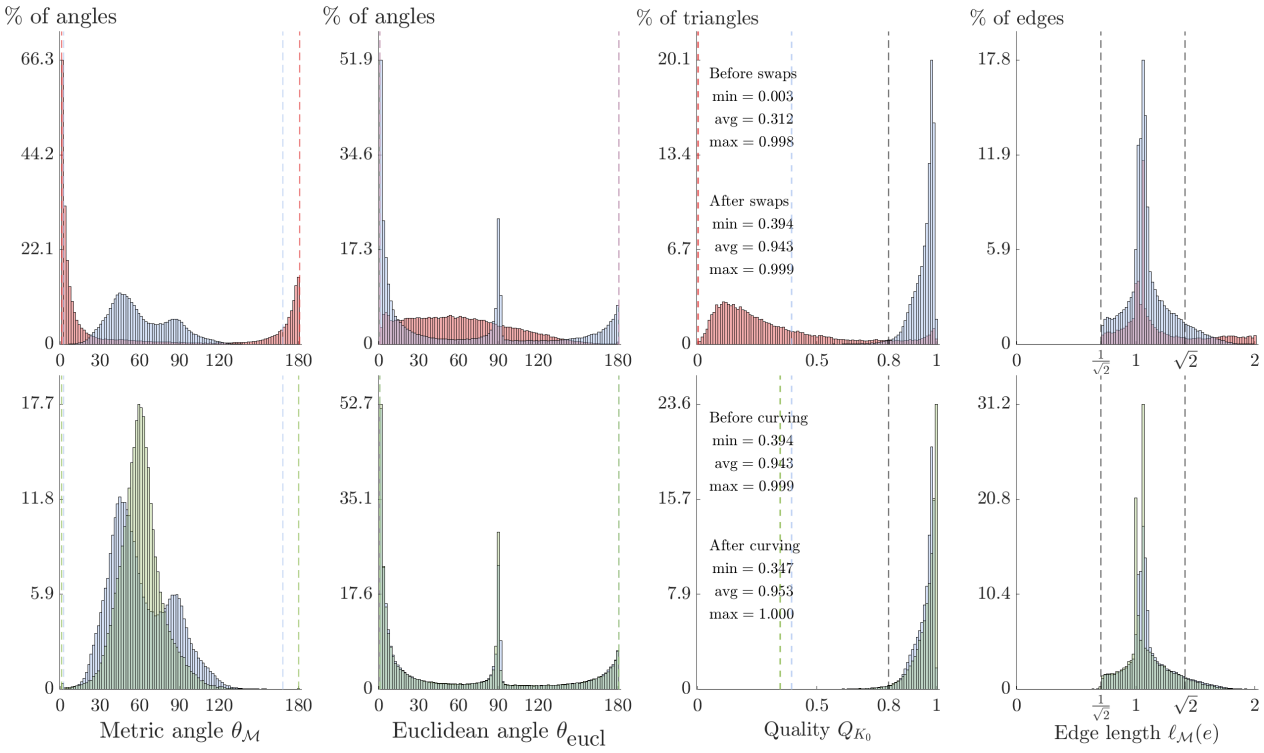


Figure 7.17 Quality indicators for the curved mesh adapted to Solomon and Gollub's flow map. Top: before (red) and after (blue) swapping the edges of the linear mesh. Bottom: before (blue) and after (green) curving the edges to minimize the length. The blue data are the same on both rows (after swap = before curving).

7.4 Conclusion

We detailed a complete methodology to generate 2D quasi-unit $\mathcal{P}^2$ triangulations with respect to an input metric field $\mathcal{M}$. Following the definitions introduced in Chapter 4, quasi-unit $\mathcal{P}^2$ triangles have controlled edge lengths, edge parameterizations and inner angles w.r.t. $\mathcal{M}$. In addition to sampling the vertices at unit distance from one another, this is enforced by either minimizing the distortion with respect to the chosen ideal simplex, or by creating geodesic edges meeting at the prescribed angles.

Mesh generation was approached by curving a valid and metric-conforming linear mesh. This ensures the final mesh is valid if edge curvature is not needed. Starting from a metric-conforming discretization of the boundary, the linear mesh is obtained by triangulating vertices sampled at controlled angles along either the geodesics or the integral curves of the metric eigenvectors. Isometric triangles have geodesic edges, however, following the eigenvectors is cheaper and yields integral curves close to the geodesics for all the tested cases. This initial mesh is improved by swapping the straight edges for the distortion-based quality.

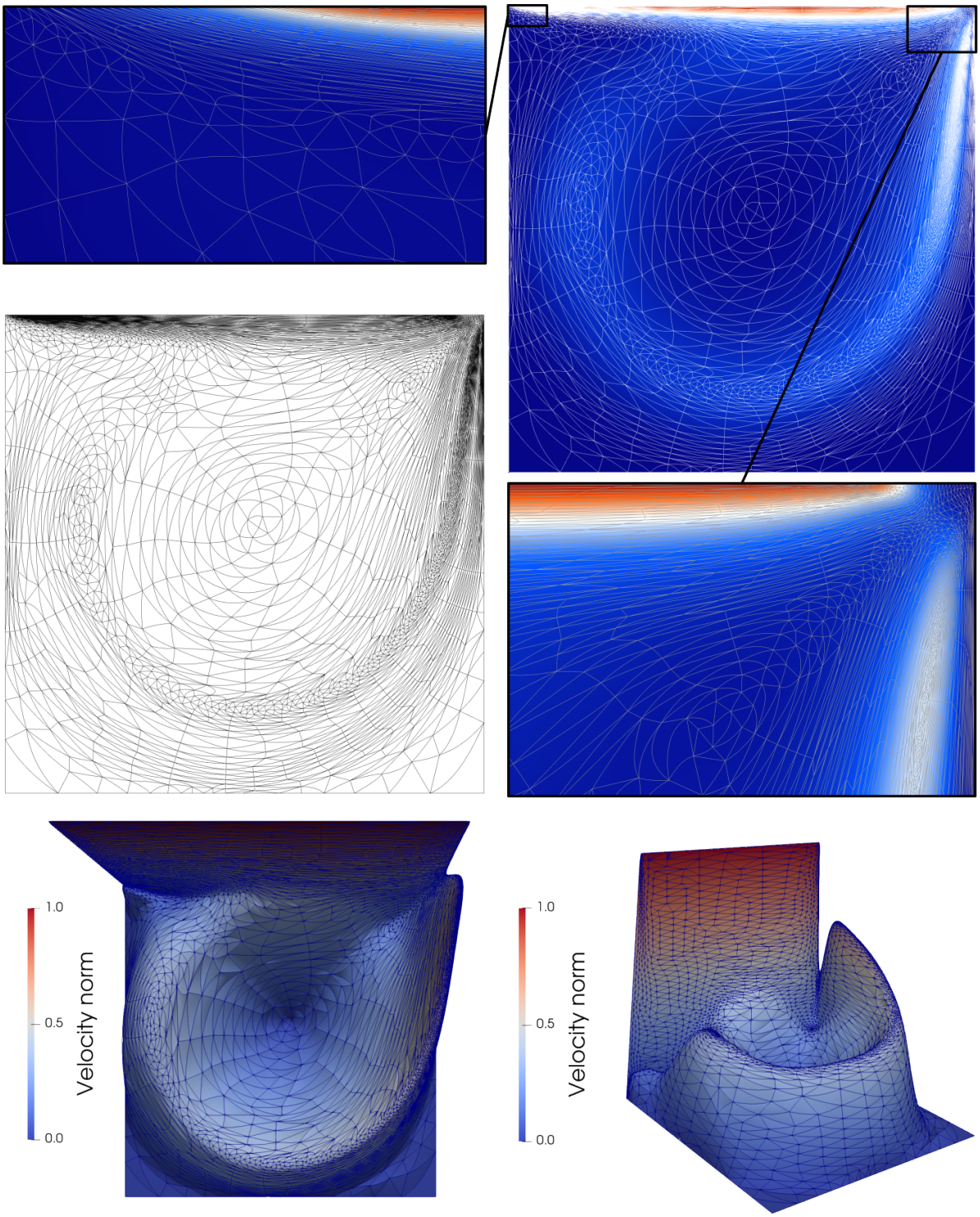


Figure 7.18 $\mathcal{P}^2$ mesh (12,655 vertices and 6,250 triangles) adapted to the norm of the velocity of the flow inside a lid-driven cavity.

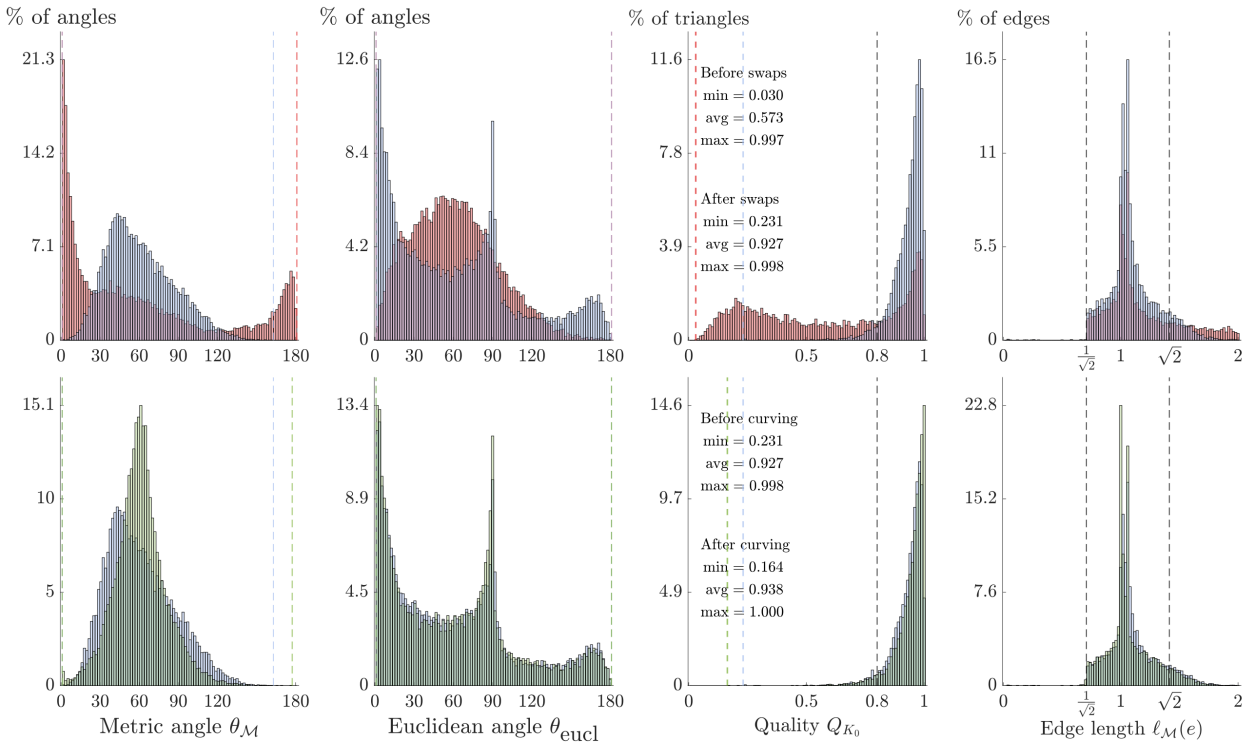


Figure 7.19 Quality indicators for the curved mesh adapted for the lid-driven cavity. Top: before (red) and after (blue) swapping the edges of the linear mesh. Bottom: before (blue) and after (green) curving the edges to minimize the length. The blue data are the same on both rows (after swap = before curving).

In a second step, straight edges are curved to minimize either the distortion w.r.t. K_0 or the Riemannian edge length. Length minimization is a cheap and local problem on each $\mathcal{P}^2$ edge, whereas minimizing the distortion is an global and expensive optimization problem.

Curved $\mathcal{P}^2$ meshes for both criteria exhibit very similar quantitative quality indicators. We can infer that definitions *QU1p* and *QU2p* describe quasi-unit curvilinear triangles in a comparable way in practice, despite *QU1p* being a stronger indicator. *QU2p*, that is, quasi-geodesic parabolic triangles of controlled length and angles, is the cheapest of the two and can be expected to generate quasi-unit $\mathcal{P}^2$ meshes competitively in the future.

The meshing procedure was illustrated on simple and more complex analytical fields, as well as on a classical CFD benchmark. For all cases, the curved triangulations exhibit good quality indicators, showing that the vast majority of elements are close to unit.

This chapter addressed the adequacy between metric and mesh, but not between metric and error estimation. Consequently, optimal triangulation for the interpolation error on quadratic meshes are not yet obtained.

Several issues remain in the current approach. First, a formal characterization of the link between the considered trajectories, eigenvectors and geodesics, is lacking. Eigenvectors are not defined for Riemannian metrics, so it is possible that more general notions are at play. The second and more practical issue is that long edges and slivers are not handled in the current pipeline. The former can negatively impact the L^2 interpolation error, while the sliverness is directly involved in H^1 seminorm estimates [38]. In future developments, edge splits or reconnections of cavities built around long edges can be considered to eliminate edges with length outside of the prescribed bracket, while implementing a cavity operator [24,42,43,107] would allow to filter slivers during the vertex insertion step.

Lastly, metric interpolation, and specifically the point localization step, is the bottleneck of the proposed methodology. This is intrinsic to meshing with a nonuniform size prescription, as also discussed in Chapter 8 for isotropic size fields. Significant speedups can be expected from optimizing the point localization in the background mesh, by e.g. supporting multiple points queries and/or hinting at the location in the background mesh, to avoid constantly rechecking for overlapping boxes in the RTree.

CHAPTER 8 AUTOMATIC FEATURE-PRESERVING SIZE FIELD FOR 3D MESH GENERATION

This chapter presents early work of this thesis regarding the generation and storage of an isotropic size field induced by the CAD model. It is disconnected from the previous chapters, and is a reproduction of the paper:

Bawin, A., Henrotte, F., & Remacle, J.-F. (2021). "Automatic feature-preserving size field for three-dimensional mesh generation.", International Journal for Numerical Methods in Engineering, 122(18), 4825-4847.

8.1 Abstract

This paper presents a methodology aiming at easing considerably the generation of high-quality meshes for complex 3D domains. To this end, a mesh size field $h(\mathbf{x})$ is computed, taking surface curvatures and geometric features into account. The size field is tuned by five intuitive parameters and yields quality meshes for arbitrary geometries. Mesh size is initialized on a surface triangulation of the domain based on discrete curvatures and medial axis transform computations. It is then propagated into the volume while ensuring the size gradient ∇h is controlled so as to obtain a smoothly graded mesh. As the size field is stored in an independent octree data structure, it can be computed separately, then plugged into any mesh generator able to respect a prescribed size field. The procedure is automatic, in the sense that minimal interaction with the user is required. Applications of our methodology on CAD models taken from the very large ABC dataset are presented. In particular, all presented meshes were obtained with the same generic set of parameters, demonstrating the universality of the technique.

8.2 Introduction

Geometric models used in industry have considerably grown in complexity over the last decades, and it is now common to mesh models with tens of thousands of model faces. Ideally, a designer would create the CAD model, press the *generate mesh* button, and obtain in less than a minute a computational mesh valid *as is* for a finite element simulation. Practitioners in the field know however that things do not work out that easily in reality. Mesh generation for complex geometries is in practice a time-consuming task often involving intermediary

meshes, progressively enhanced to fulfill specified mesh size and quality requirements.

The purpose of mesh generation is to build meshes with elements of controlled size and quality. In the context of mesh adaptation, the meshing algorithm is constrained by a mesh size field defined on the domain to be meshed, and whose value at a point is the expected element size in the vicinity of that point. The mesh size is usually derived from an error estimation procedure performed on the solution of a prior finite element or finite volume analysis, by requesting a smaller size at places where the discretization error is deemed large.

Yet, when solving a problem for the first time, an initial mesh has to be generated without information from a prior computation, and the size field to generate that initial mesh has to be constructed from scratch on basis of the geometrical data of the model only. Given a CAD model, there exist a number of theoretical prerequisites on the size field to ensure a computable mesh, and the purpose of this paper is to describe an automated algorithm to compute a mesh size field fulfilling those prerequisites *a priori*. The proposed approach is "user-driven", in the sense that users should be able to generate a workable computational mesh in one click on basis of a limited number of intuitive meshing parameters, understandable by any finite element practitioner with no extensive background in meshing.

An isotropic mesh size field $h(\mathbf{x})$ is thus a scalar function indicating the expected element size at any point $\mathbf{x}$ in a domain to be meshed. A first design choice concerns the mathematical representation of h. As our goal is to build a "first mesh", no background mesh is yet available against which h could be interpolated. A classical solution (e.g. in Gmsh [112]) is to define element sizes directly on the geometrical entities of the model. Mesh size can be prescribed at the vertices of the CAD model, for instance, and smoothly interpolated on model edges. They are then subsequently interpolated on surface mesh vertices. Size fields interpolated this way may however be biased by geometric features of the surface mesh, such as gaps, fins or channels, which are assigned locally a small size that is not expected to spread out at distance in the bulk of the volume. This approach is therefore not 100% reliable and defining the size on auxiliary objects allows for a better control and prevents the aforementioned phenomenon.

Two kinds of representation for mesh size fields are encountered in the literature: simplicial background meshes [113–116], and Cartesian grids, initially in the form of uniform grids [117], and later on in the form of non-uniform or tree-based grids, such as octrees [118–120]. A graphical representation of such data-structures in the two-dimensional case can be found in the first figure of Persson [121]. As the refinement of uniform Cartesian grid is constrained by the smallest feature in the CAD model, their memory cost quickly becomes prohibitive in practice, and they were rapidly abandoned for the sake of simplicial background meshes and

octrees.

Both these representations have their pros and cons. Simplicial background meshes are typically triangulations provided by the CAD modeller, where mesh size is stored on vertices and interpolated on queried nodes. They offer an accurate representation of the boundaries, and the mesh size query procedure in the three-dimensional case is reduced to a search in the two-dimensional parametric space [115, 122]. However, size fields represented this way are rather sensitive to the location of the vertices in the background mesh [114], and the access time to the mesh size at a point in the background mesh might be linear in the number of nodes in the worst case. To improve on this, Chen et al. [115] proposed a *walk-through* algorithm based on the backward search from Shan et al. [123], although it requires a well-guessed element to quickly locate the point in the mesh, assuming an interconnection between the background mesh and the meshing algorithm. Octrees, on the other hand, are orientation sensitive Cartesian structures with applications in both direct meshing [124,125] and size field design. They lack the geometrical flexibility of simplicial meshes and significative refinement may be necessary to accurately resolve surface-based information, e.g. curvatures. However, octree-based size fields offer adaptive capabilities to represent quickly and easily complex size distributions across the structure. Moreover, octrees offer fast access to query points in $\mathcal{O}(\log_8 n)$ time, where n denotes the number of octants, i.e., the number of leaves in the octree.

In this work, we choose to store the mesh size field in an octree for adaptivity and efficiency concerns. The octree implementation is provided by p4est [126], of which the serial version is used. In our implementation, a uniform mesh size is assigned to each octant in the octree, which is the most natural option with p4est.

As mentioned earlier, there are theoretical prerequisites to ensure the computability of a mesh on a given CAD model. Those prerequisites can be associated with the following five intuitive mesh parameters.

Bulk size. A bulk or default mesh size h_b. When creating the size field, the octree is refined uniformly until every octant size is smaller or equal to h_b. This amounts to say that all sizes are initially set to the bulk value h_b.

Curvature. When using piecewise linear elements, the main term of geometrical error produced by a mesh is related to the curvature of surfaces. The local mesh size $h(\mathbf{x})$ should hence be related to the maximal curvature $\kappa_n(\mathbf{x})$ of the surfaces. This is done with the node density parameter n_d that specifies the number of subdivisions of the perimeter of the local osculating circle.

Small features. A CAD model may also contain narrow or thin regions, or *features*, e.g. longerons that are the load-bearing components of aerospace structures. As such regions may have moderate or no curvature at all, they are likely to be overlooked by an algorithm that solely links mesh size with curvature. Features are typically included using a distance function [127] or the medial axis transform [119, 120]. The latter is used here to estimate the thickness of narrow regions, see Section 2.2. On this basis a third parameter n_g is defined in our algorithm that specifies the minimal number of elements across the thickness of a narrow region. We call *feature mesh size* h_f that thickness divided by n_g. The feature mesh size h_f and the curvature mesh size h_c are the indicators used by our algorithm to recursively refine the octree containing the mesh size field, see Section 2.3.

Boundedness. A minimum mesh size $h_{\min}$ has to be defined as a fourth parameter in our algorithm to forbid unacceptably small mesh sizes, whenever curvatures are very high for instance (e.g. at corners or at the tip of a cone).

Smoothness. Accurate finite element and finite volume simulations usually require that mesh sizes vary not too abruptly across the domain of computation. Yet, curvatures and feature sizes may exhibit sharp variations in practice, resulting in unacceptably large mesh size gradations. A fifth parameter $\alpha > 1$ is thus defined that bounds (from above) the length ratio between two adjacent edges in the mesh. In Section 2.4, we show that this condition is equivalent to limit the mesh size gradient to $\|\nabla h\| < \alpha - 1$.

Our approach thus defines five parameters that are easy to understand by finite element practitioners, and can be given a reasonable and rather universal default value:

- the bulk size, or default mesh size on newly created octants, h_b. The default value is $h_b = L/20$, where L is the largest dimension of the axis-aligned bounding box of the CAD model;
- the minimal size allowed in the final mesh, $h_{\min}$. The default value is $h_{\min} = L/1000$;
- the number of elements n_d used to accurately discretize a complete circle. The default value is $n_d = 20$;
- the number of element layers in thin gaps n_g. The default value is $n_g = 4$;
- the gradation, or length ratio of two adjacent edges in the final mesh, α. The default value is $\alpha = 1.1$.

In addition to these geometry-based criteria, a user-defined mesh size function $h_u(\mathbf{x})$ can be provided to specify exterior constraints on the size field. If so, the smallest instruction

between $h_u(\mathbf{x})$ and the geometry-based mesh size is considered throughout the construction of the size field.

This size field computation has been implemented in Gmsh (version 5 and higher), which is also the tool used to generate all meshes presented in this paper. It is planned that the algorithm presented in this paper be soon integrated as a standard procedure in the meshing pipeline of Gmsh.

8.3 Description of the algorithm: a worked-out example

To illustrate the steps of the construction of the mesh size field $h(\mathbf{x})$, the CAD model of an engine block is considered as a application example (Fig. 8.1). This geometry contains curved surfaces and narrow features that are typical of industrial CAD models. The input data for our algorithm is a surface mesh of the CAD model, from which curvature and feature mesh size are computed. The size field is generated in an independent structure in the five following steps (Fig. 8.1): (i) compute the curvature mesh size h_c from the approximate curvature on the surface mesh of the model; (ii) compute the feature mesh size h_f from the medial axis of the geometry; (iii) initialize the octree as the bounding box of the model and refine it uniformly until the size of all octants is at most the bulk size h_b; (iv) recursively refine the octree based on both the curvature and the feature mesh size, and assign the appropriate uniform mesh size in all newly created octants; (v) smooth out the size field so as to limit its gradient to $\alpha - 1$. During step (v), the structure of the octree is not modified. Only its stored sizes $h(\mathbf{x})$ are limited to satisfy $|\nabla h| < \alpha - 1$, see Section 2.4.

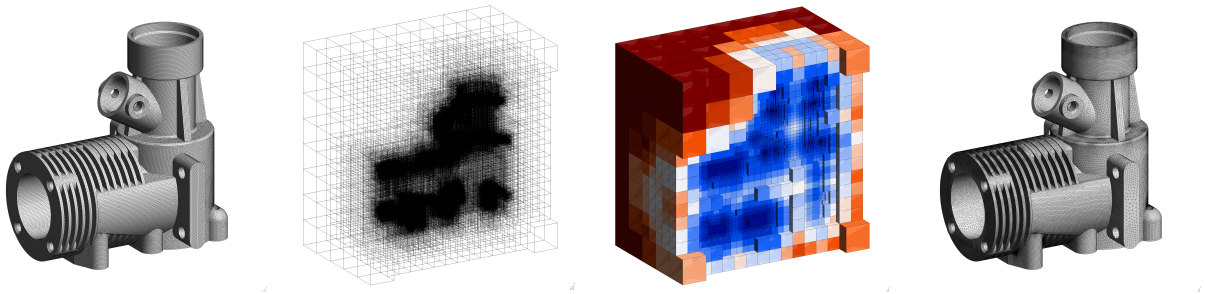


Figure 8.1 Overview of the algorithm for mesh size field computation, from left to right: $(i - ii)$ surface mesh of the engine block, from which discrete curvature and feature sizes are computed; $(iii - iv)$ the octree is refined based on curvature and feature mesh sizes, (v) mesh size gradient is limited, yielding smoother mesh size field; generation of the final mesh.

8.3.1 Approximation of surface curvatures

When a CAD model is represented with a mesh of piecewise linear elements, the main term of geometrical error is due to the curvature of surfaces. The mesh size should thus be reduced in areas of high curvature. We introduce to this end the *curvature mesh size* $h_c(\mathbf{x})$. Although definition slightly varies in the literature [115, 116, 128, 129], they all rely on the subdivision of the perimeter of local osculating circles. The osculating circle at a point of a planar curve is the circle that best approximates the curve in the vicinity of the point, i.e., having same tangent and same curvature. On a smooth surface, there is an osculating circle in every direction, and the most critical curvature mesh size is related to the minimal radius of these circles, or reciprocally to the maximum normal curvature $\kappa_{n,\max}(\mathbf{x})$. Curvature mesh size is thus defined as follows:

$$h_c(\mathbf{x}) = \frac{2\pi r(\mathbf{x})}{n_d} = \frac{2\pi}{\kappa_{n,\max}(\mathbf{x})\, n_d}, \tag{8.1}$$

with n_d a user-defined node density.

If a CAD model is available in the background, surface parametrizations are at-hand and the maximal curvature $\kappa_{n,\max}(\mathbf{x})$ can be obtained through the solid modeller's API. Curvature queries on the CAD model can be costly however, and are usually designed for single point query. In this work, the input data considered for the size field pipeline is a triangulation of the CAD model, on which normal curvatures are approximated. We assume that the triangulation is colored, i.e., triangles on each side of a feature edge, such as a ridge or a corner, are identified with a different color, so that surface discontinuities are not taken into account when computing the discrete curvature.

To approximate surface curvature, the tensor averaging methodology described by Rusinkiewicz [130] was considered. It is here briefly recalled with their notations. Let $(\mathbf{u}, \mathbf{v})$ denote an orthonormal basis in the tangent plane at a point $\mathbf{x}$ of a smooth surface, and $\boldsymbol{s} = (s_1, s_2)$ be an arbitrary direction in that plane. The normal curvature at $\mathbf{x}$ in the direction $\boldsymbol{s}$ is given by

$$\kappa_n(\mathbf{x}) = \mathbf{II}(\boldsymbol{s}, \boldsymbol{s}) = \begin{pmatrix} s_1 & s_2 \end{pmatrix} [\mathbf{II}] \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} s_1 & s_2 \end{pmatrix} \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \tag{8.2}$$

where $\mathbf{II}$ denotes the second fundamental form defined hereafter, and $[\mathbf{II}]$ denotes its matrix representation in the chosen basis. The eigenvalues $\kappa_1(\mathbf{x})$ and $\kappa_2(\mathbf{x})$ of the symmetric matrix $[\mathbf{II}]$, known as the *principal curvatures*, are the maximum and minimum values of normal curvature at $\mathbf{x}$. Since $\kappa_{n,\max}(\mathbf{x}) = \max(|\kappa_1(\mathbf{x})|, |\kappa_2(\mathbf{x})|)$ is required to define the curvature mesh size (8.1), our goal is to build an approximation of $[\mathbf{II}]$ at each vertex of the surface mesh.

The idea proposed by Rusinkiewicz [130] is to first compute $[\mathbf{II}]$ on the triangles, and average them over adjacent triangles to obtain the needed per-vertex information. We start by computing per-vertex normal vectors $\mathbf{n}_i$ by averaging the normals of triangles adjacent to each vertex using arithmetic mean. On each triangle, an arbitrary orthonormal coordinate system $(\mathbf{u}_f, \mathbf{v}_f)$ is then defined. The components of the quadratic form $\mathbf{II}$ in that basis read

$$[\mathbf{II}] = \begin{pmatrix} \mathbf{II}(\boldsymbol{u}_f, \boldsymbol{u}_f) & \mathbf{II}(\boldsymbol{u}_f, \boldsymbol{v}_f) \\ \mathbf{II}(\boldsymbol{u}_f, \boldsymbol{v}_f) & \mathbf{II}(\boldsymbol{v}_f, \boldsymbol{v}_f) \end{pmatrix}, \tag{8.3}$$

and can be evaluated as $\mathbf{II}(\boldsymbol{u}, \boldsymbol{v}) = L(\boldsymbol{u}) \cdot \boldsymbol{v}$ where the *shape operator* $L(\boldsymbol{s}) = \nabla_{\boldsymbol{s}}\mathbf{n}$ is the directional derivative of the normal vector $\mathbf{n}$ along a direction $\boldsymbol{s}$ in the tangent plane. Along the particular direction in the plane $(\boldsymbol{u}_f, \boldsymbol{v}_f)$ given by the triangle edge $\boldsymbol{e}_0 = (\boldsymbol{e}_0 \cdot \boldsymbol{u}_f)\boldsymbol{u}_f + (\boldsymbol{e}_0 \cdot \boldsymbol{v}_f)\boldsymbol{v}_f$ connecting vertices $\mathbf{x}_1$ and $\mathbf{x}_2$, one has the following finite difference approximation:

$$\mathbf{II}(\boldsymbol{e}_0, \boldsymbol{u}_f) = L(\boldsymbol{e}_0) \cdot \boldsymbol{u}_f = \nabla_{\boldsymbol{e_0}}\mathbf{n} \cdot \boldsymbol{u}_f = (\mathbf{n}_2 - \mathbf{n}_1) \cdot \boldsymbol{u}_f, \tag{8.4}$$

where the directional derivative $\nabla_{\boldsymbol{e_0}}\mathbf{n}$ in Rusinkiewicz [130] is approximated by the difference of normal vectors along $\boldsymbol{e}_0$. On the other hand, since the second fundamental form is a bilinear form, we have

$$\mathbf{II}(\boldsymbol{e}_0, \boldsymbol{u}_f) = \mathbf{II}\Big((\boldsymbol{e}_0 \cdot \boldsymbol{u}_f)\boldsymbol{u}_f + (\boldsymbol{e}_0 \cdot \boldsymbol{v}_f)\boldsymbol{v}_f, \boldsymbol{u}_f\Big) = (\boldsymbol{e}_0 \cdot \boldsymbol{u}_f)\,\mathbf{II}(\boldsymbol{u}_f, \boldsymbol{u}_f) + (\boldsymbol{e}_0 \cdot \boldsymbol{v}_f)\,\mathbf{II}(\boldsymbol{v}_f, \boldsymbol{u}_f). \tag{8.5}$$

Combining (8.4) and (8.5) and proceeding the same way for $\boldsymbol{v}_f$ yields the following equality for the components of $[\mathbf{II}]$:

$$[\mathbf{II}] \begin{pmatrix} \boldsymbol{e}_0 \cdot \boldsymbol{u}_f \\ \boldsymbol{e}_0 \cdot \boldsymbol{v}_f \end{pmatrix} = \begin{pmatrix} (\mathbf{n}_2 - \mathbf{n}_1) \cdot \boldsymbol{u}_f \\ (\mathbf{n}_2 - \mathbf{n}_1) \cdot \boldsymbol{v}_f \end{pmatrix}. \tag{8.6}$$

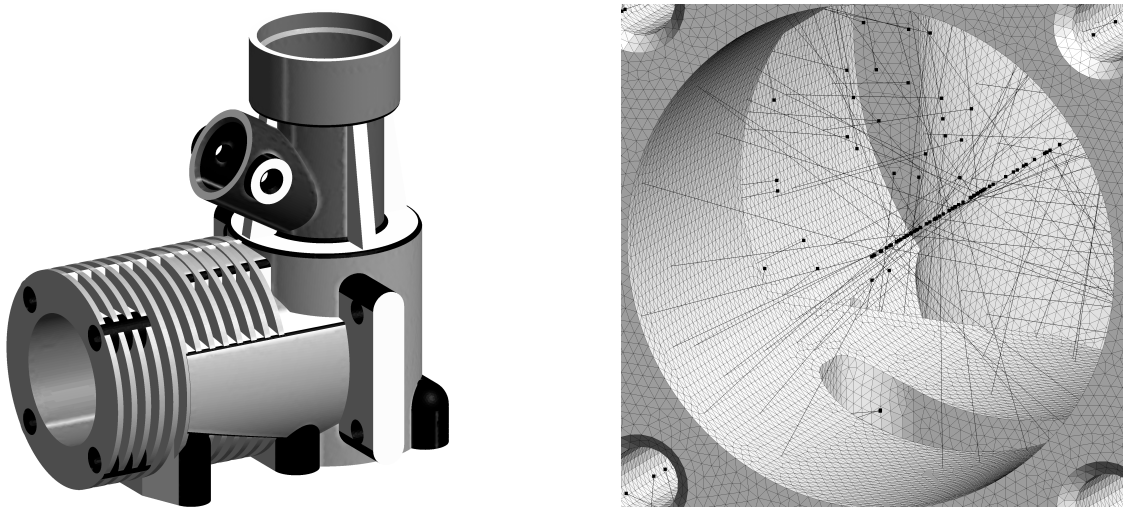
Repeating the same procedure for the two remaining edges $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, one ends up with a system of 6 equations for 3 unknowns, which can be solved using a least square method to obtain the matrix representation of $\mathbf{II}$ for the triangle. This per-triangle representation of $\mathbf{II}$ is expressed in the local basis $(\boldsymbol{u}_f, \boldsymbol{v}_f)$. To combine the contributions of triangles adjacent to a vertex p, the orthonormal basis $(\boldsymbol{u}_p, \boldsymbol{v}_p)$ is defined in the plane perpendicular to the normal vector at p. Denoting $(\boldsymbol{u}'_f, \boldsymbol{v}'_f)$ as the basis $(\boldsymbol{u}_f, \boldsymbol{v}_f)$ rotated to be made coplanar with $(\boldsymbol{u}_p, \boldsymbol{v}_p)$, the components of the triangle contribution to the per-vertex representation $[\mathbf{II}]_p$ can then be obtained as

$$e_p = \boldsymbol{u}_p^T\,[\mathbf{II}]\,\boldsymbol{u}_p = \begin{pmatrix} \boldsymbol{u}_p \cdot \boldsymbol{u}'_f \\ \boldsymbol{u}_p \cdot \boldsymbol{v}'_f \end{pmatrix}^T [\mathbf{II}] \begin{pmatrix} \boldsymbol{u}_p \cdot \boldsymbol{u}'_f \\ \boldsymbol{u}_p \cdot \boldsymbol{v}'_f \end{pmatrix}, \qquad f_p = \boldsymbol{u}_p^T\,[\mathbf{II}]\,\boldsymbol{v}_p, \qquad g_p = \boldsymbol{v}_p^T\,[\mathbf{II}]\,\boldsymbol{v}_p. \tag{8.7}$$

As for the normal vectors, the contributions of triangles adjacent to the vertex p are then averaged to obtain $[\mathbf{II}]_p$. Once again, arithmetic mean is used to compute the average tensor, that is, all triangles have an equal contribution regardless of their area.

Once the matrix representation of the second fundamental form $[\mathbf{II}]_p$ has been computed at a vertex, its eigenvalues are extracted and the curvature mesh size h_c at the vertex is set using (8.1).

The computed maximal curvature for the engine block model is shown below (Fig. 8.2a): curvature is computed on each colored face of the triangulation, discounting the feature edges.



(a) (b)

Figure 8.2 Left: Approximated maximum normal discrete curvature $\kappa_n = \max(|\kappa_1|, |\kappa_2|)$ of the surface triangulation, shown in grayscale: curved areas are in gray-black while regions with mild curvature are in light gray. Right: A subset of the Voronoï vertices lying the farthest from the Delaunay vertices, called *poles* (black dots), approximate the medial axis as the mesh density increases.

8.3.2 Feature size

Whenever two surfaces with moderate or no curvature are close to each other but the distance between them is smaller than the mesh size, the mesh generator will place only one element in the gap between those surfaces. In many engineering applications like solid mechanics or fluid mechanics, having only one element in a gap means that both sides are connected by one single mesh edge. If Dirichlet boundary conditions are applied, such as a non-slip boundary condition, the gap is then essentially closed, leading to an unwanted change of the domain topology. Note that *a posteriori* error estimation will not detect large errors in those

closed gaps where the solution is essentially constant.

Hence, special care should be given to such narrow geometrical regions, where curvature information alone is not enough to determine a suitable mesh size field. To this end, we define the *feature size* $f(\mathbf{x})$ as a measure of the local gap thickness (Fig. 8.3a). If ∂V is the boundary of a volume V, $f(\mathbf{x})$, $\mathbf{x} \in \partial V$, is defined as twice the distance between $\mathbf{x}$ and the *medial axis* of volume V. If a surface bounds two volumes, then the minimum feature size is chosen. The *feature mesh size* at the considered vertices, now, is the feature size divided by the desired number of element layers in narrow regions n_g, i.e., $h_f(\mathbf{x}) = f(\mathbf{x})/n_g$. The feature mesh size h_f is thus in a similar relationship to the feature size f than the curvature mesh size h_c was to the maximum principal curvature κ.

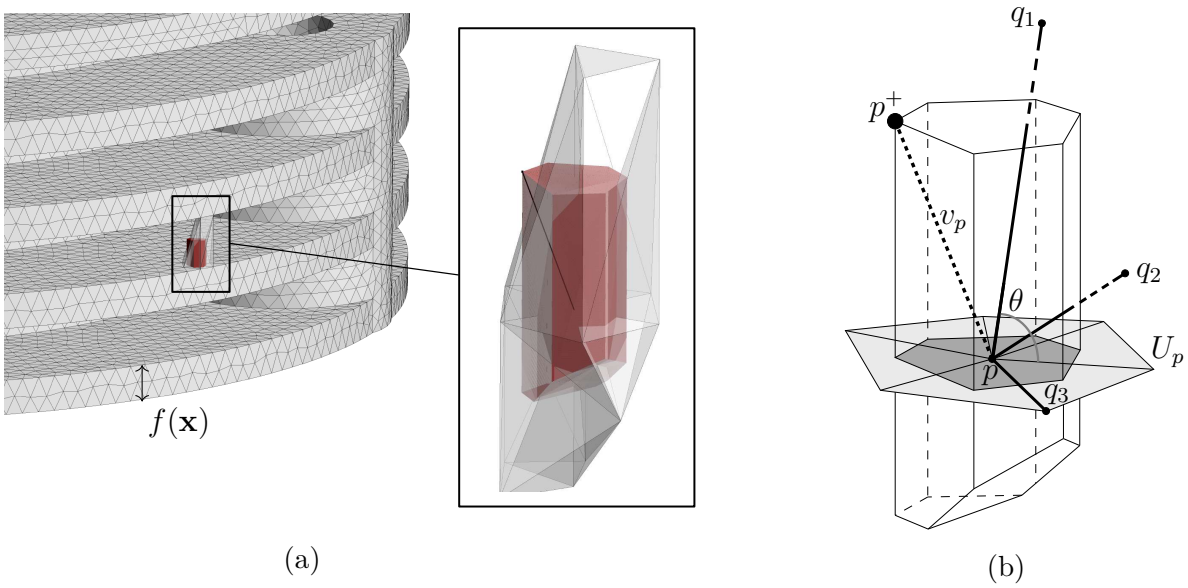
The evaluation of the feature size $f(\mathbf{x})$ requires computing an approximation of the medial axis of all volumes in the computational domain. The medial axis of a volume V, also referred to as its *skeleton*, is defined as the set of points having more than one closest point on its boundary ∂V. Equivalently, the medial axis is the set of the centers of all spheres tangent to ∂V in two or more points, and the feature size $f(\mathbf{x})$ is twice the radius of the sphere tangent at $\mathbf{x}$.

We rely on the algorithm MEDIAL introduced by Dey and Zhao [131] to compute a discrete approximation of the medial axis. The input data for MEDIAL is a Delaunay tetrahedrization of the vertices of the surface mesh only, i.e., a set of tetrahedra, often called *empty mesh*, filling the computational domain and whose nodes all lie on the surfaces. This algorithm is based on the Voronoï diagram of these vertices and has suitable convergence properties, in the sense that the output set of facets converges to the medial axis as the surface mesh density increases. In this work, we implemented MEDIAL as is. For sake of completeness, we here briefly remind the rationale behind the algorithm, using their notations.

In 2D, the vertices of the Voronoï cells dual to the empty mesh give straigthaway an approximation of the medial axis. The same is however not always true in 3D, because sliver tetrahedra of the empty mesh can persist close to the boundary as the surface mesh is refined. However, by pruning as explained below the Voronoï vertices dual of these sliver tetrahedra, a subset of Voronoï vertices called *poles* can be defined that do approximate the medial axis (Fig. 8.2b). Given the Voronoï cell dual of a surface vertex p (Fig. 8.3a), the corresponding pole p^+ is defined as the Voronoï vertex that is the most distant from p. Each surface vertex p is thus associated with a pole p^+ and a *pole vector* $v_p = p^+ - p$, the latter approximating the normal to ∂V at p (Fig. 8.3b).

The plane passing through p with normal v_p intersects edges of the Voronoï cell, and the Delaunay facets dual to these edges are pictorially called the *umbrella* U_p *of* p by Dey and

Zhao (Fig. 8.3b, in light grey).



(a) (b)

Figure 8.3 Left: View of the input triangular surface mesh and of the Delaunay tetrahedra adjacent to a mesh vertex p. Zoom in on the same tetrahedra (transparent) and the associated dual Voronoï cell (in red). Right: Voronoï cell dual to the Delaunay vertex p in the triangular surface mesh. The pole vector v_p (dashed) connects the mesh vertex p to the farthest Voronoï vertex (tetrahedron circumcenter) or *pole* p^+. The Delaunay edges pq_1, pq_2 and pq_3 connect p to neighbouring Delaunay vertices q_1, q_2 and q_3, respectively. Only pq_1 and pq_2 satisfy one of the filtering conditions and are considered to compute the feature size. In grey, the umbrella U_p of p is the Delaunay facets (triangles) dual to the Voronoï edges cut by the plane through p with normal v_p (not shown).

Triangles of the umbrella are used to select some of the Delaunay edges adjacent to p whose dual facets will eventually form the discrete medial axis. The idea is to select Delaunay edges pq that (a) make a sufficiently large angle with the triangles of the umbrella, or (b) are significantly longer than the circumradius of these triangles. Condition (a) measures how normal the edge pq is to the umbrella U_p and is referred to as the *angle condition*. The edge should make an angle larger than θ with each triangle of the umbrella, or conversely, an angle smaller than $\pi/2 - \theta$ with their normal vector. In practice, we evaluate [131]:

$$\max_{i \in U_p} \angle(\mathbf{pq}, \hat{\mathbf{n}}_i) < \frac{\pi}{2} - \theta, \tag{8.8}$$

where $\mathbf{pq}$ is a vector parallel to the edge pq, i denotes a triangle in U_p and the threshold angle is set to $\theta = \pi/8$. Condition (b) ensures that edges of the surface mesh are removed. It does so by selecting long edges for which the angle condition has failed, and is referred to as

the *ratio condition*. In practice, only edges at least $\rho = 8$ times longer than the circumradius R of the triangles in U_p are considered, and the condition reads [131]:

$$\max_{i \in U_p} \frac{\|pq\|}{R_i} > \rho. \tag{8.9}$$

The numerical values for parameters θ and ρ are those suggested by Dey and Zhao. They yield a good approximation of the medial axis for a large variety of geometries. If a Delaunay edge pq satisfies either of these two conditions, its dual Voronoï facet is added to a set F, forming the approximate medial axis. Consider the three mesh vertices q_i connected to p through a Delaunay edge $pq_i = q_i - p$ on Figure 8.3b. Edge pq_1 makes a large angle θ with the triangles of the umbrella, and edge pq_2 is several times longer than the largest circumradius: both are added to the set E dual to F. Edge pq_3 is a short surface edge mesh lying flat to the umbrella, and is thus removed from the list of candidate edges.

In our size field computation, we need not compute the dual facet to edges in E: for each Delaunay edge pq satisfying the angle or ratio conditions, the local feature size is directly given by the edge length $\|pq\|$. Mesh size at both vertices p and q is thus defined as the edge length divided by the desired number of element layers in features:

$$h_f(\mathbf{x}_p) = h_f(\mathbf{x}_q) = \frac{\|pq\|}{n_g}. \tag{8.10}$$

The mesh size on the octants containing these vertices is then lower bounded if necessary:

$$h = \max\left(h_{\min}, \min\left(h_f, h_c, h_u, h_b\right)\right). \tag{8.11}$$

The medial axis also contains the centers of spheres with radius vanishing to zero in angles and corners of the volume (Fig. 8.4a). Dual edges in these corners are smaller than the feature size one wishes to identify, and should thus be disregarded to avoid spurious small mesh size in these areas. In practice, an edge pq making an angle larger than θ with either one of the normal vectors $\hat{\mathbf{n}}_p$ and $\hat{\mathbf{n}}_q$ at its ends is also filtered out (Fig. 8.4b).

The quality of the approximation of the medial axis is directly related to the node density of the surface mesh. As pointed out by Dey and Zhao, the surface mesh should be an ε-sample, i.e., vertices should have neighbours within a distance $\varepsilon f(\mathbf{x})$, where $f(\mathbf{x})$ is the feature size and ε is small. Of course, as the aim is here precisely to compute the feature size, it is not known beforehand whether or not the input surface mesh is a ε-sample. A first solution is to measure beforehand the most critical feature size $f_{crit} = \min_{\mathbf{x}} f(\mathbf{x})$ of the CAD model, then generate a uniform mesh with constant mesh size $h_{crit} = \varepsilon f_{crit}$, typically with $\varepsilon \leq 0.25$ as

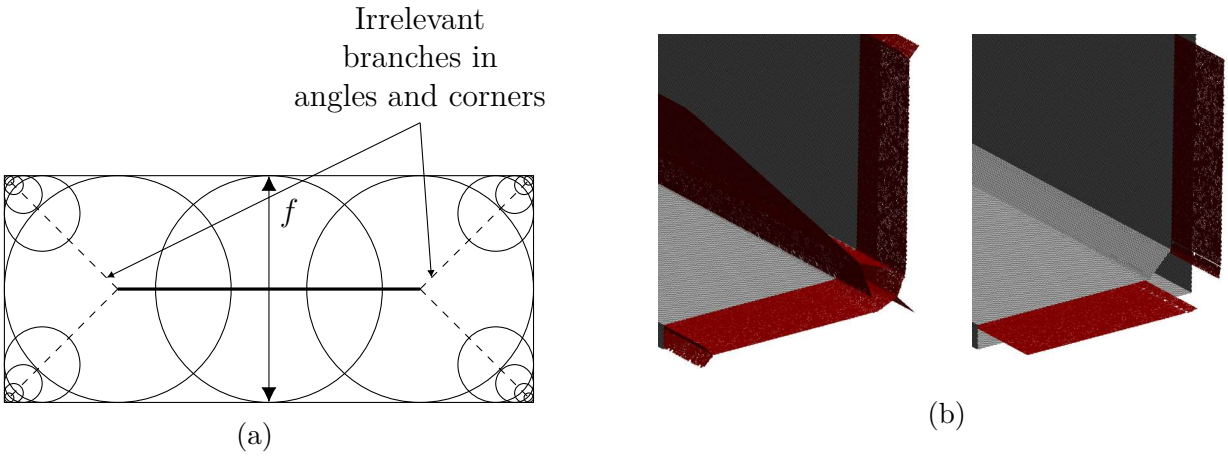
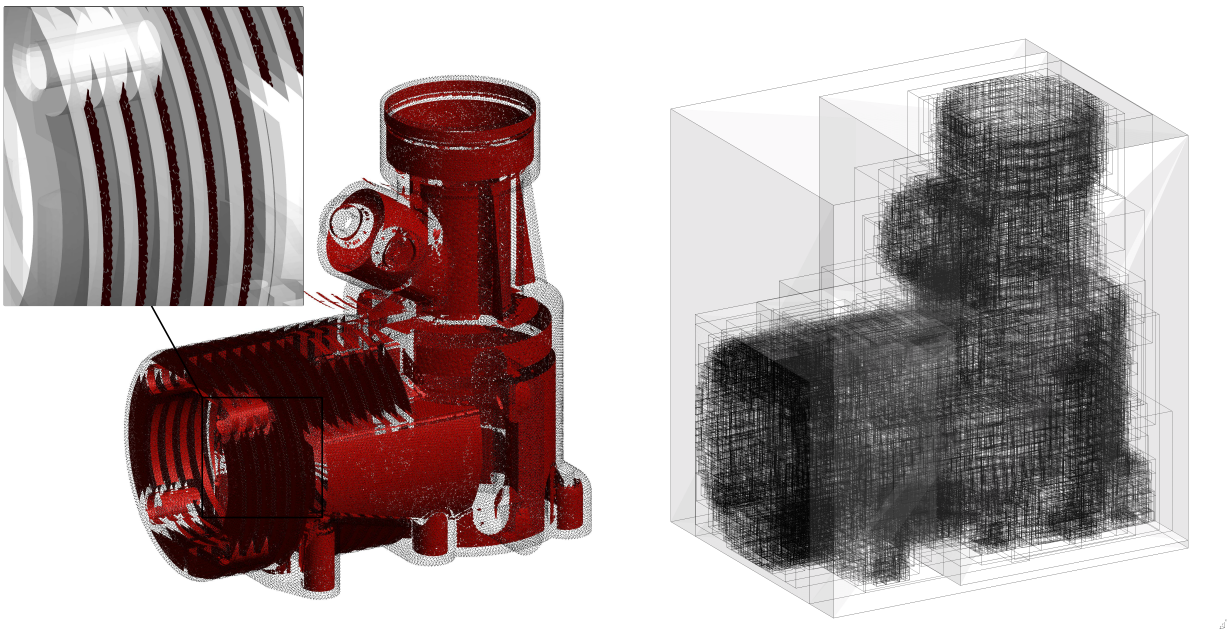


Figure 8.4 Left: The medial axis of a rectangle consists of a main branch (thick) and four secondary branches (dashed) connecting the main branch to the corners. The radii of the spheres whose center lies on the main branch are representative of the feature size f (here, the thickness of the rectangle), while the radii of the spheres on the secondary branches shrink to zero as the branch approaches the surface, and are thus not representative of f. To remove those secondary branches (Voronoï facets), the dual Delaunay edges are applied a second filtering process ensuring that the angle between the edge and the normal vectors at its extremities does not exceed θ. Right: the medial axis (in red) of an angle geometry before filtering the branches in the corners (left) and the medial axis after filtering (right).

suggested in [131]. The characteristic size of the input surface mesh is then constrained by the smallest feature in the geometry, which may result in an expensive size field computation. While we could certainly adjust the mesh size of the input mesh to be h_{crit} only in the small features, this would amount to manually specify the mesh size field, which we want to avoid. To circumvent this, one can compute an initial size field based on a reasonably fine uniform mesh, then perform the mesh generation from this field. This intermediary mesh will include an initial refinement in the small features, and will be a better candidate for the final size field computation.

The resulting medial axis for our block example after both filtering operations (i.e., conditions (8.8) and (8.9), as well as removing the spurious branches in corners) is shown on Figure 8.5a. Taking into account the local feature size allows for a refined mesh in small features of the geometry, especially in areas with zero curvature, which would be overlooked otherwise (Fig. 8.6, see also Fig. 8.10 for the volume mesh).



(a) (b)

Figure 8.5 Left: Approximated medial axis of the CAD model. The Voronoï facets dual to the filtered Delaunay edges are drawn in red. In the close-up view, only the facets of the medial axis lying outside of the volume are shown. Right: the RTree structure built from the bounding boxes of the triangles of the surface mesh.

8.3.3 Octree initialization and refinement

The mesh size field $h(\mathbf{x})$ over the domain to be meshed is now built as an octree structure. The initial octant is defined as the axis-aligned bounding box of the surface mesh, stretched in all three dimensions by a factor 1.5. In order to ensure a suitable gradation in the mesh, the guiding principle for the refinement of the octree is that the dimension of each octant should eventually be representative of the local mesh size. The octree is then first subdivided recursively and uniformly until the size of each octant is at most the bulk size h_b, and the local mesh size assigned to the octants created during this initial step are set to h_b. The octree is then further refined on basis of the curvature mesh size h_c and the feature mesh size h_f. This information, which is available on the surfaces, has to be transferred to the octree, which is a three-dimensional structure. One needs for that to detect efficiently the intersections between octants and the surface mesh. To this end, the three-dimensional bounding box of each triangle of the surface mesh is added to an R-Tree [63], a data structure used for spatial access methods (Fig. 8.5b) which acts here as the intermediary between the geometry and the octree. More specifically, the R-Tree provides for each octant of index i a list $\mathcal{T}_i$ of

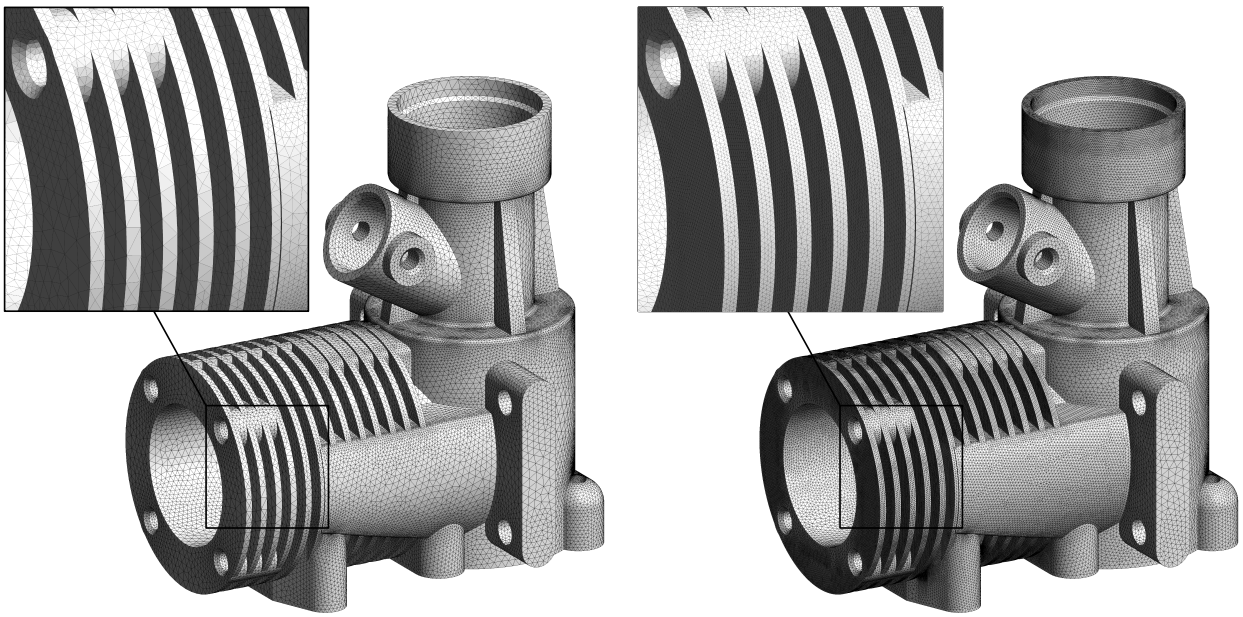


Figure 8.6 Surface mesh generated from the computed mesh size field: from curvature only (left) and considering both curvature and feature sizes (right). On the right, four layers of elements are generated in the fins around the largest cylinder.

triangles whose bounding box intersects the octant. The octant with index i is then divided until it becomes smaller than the minimal mesh size (h_c or h_f) at the vertices of all triangles in $\mathcal{T}_i$. Whenever a user-defined mesh size function $h_u = u(\mathbf{x})$ is provided, this additional constraint is taken into account at this level. The octant size being bounded from below by the user-defined minimal mesh size $h_{\min}$, octants are thus subdivided until the condition

$$h_{octant} \leq \max\left(h_{\min}, \min\left(h_c, h_f, h_u, h_b\right)\right) \tag{8.12}$$

is met everywhere in the octree.

Once this refinement is completed, the octree is balanced to ensure a maximum 2:1 level ratio between two octants across adjacent faces (Fig. 8.7), that is, the levels of two octants on each side of a face should not differ by more than one unity. This balancing is necessary to obtain suitable stencils for finite difference computations during the gradient limiting step, see Section 2.4. It is performed by P4EST. Newly created octants having an intersection with the surface mesh are assigned the mesh size determined by (8.12), otherwise the size is set to the bulk size h_b (Fig. 8.9a). At this stage, the size field features large variations, and a limitation step is required to end up with a mesh size field suitable for high-quality mesh

generation.

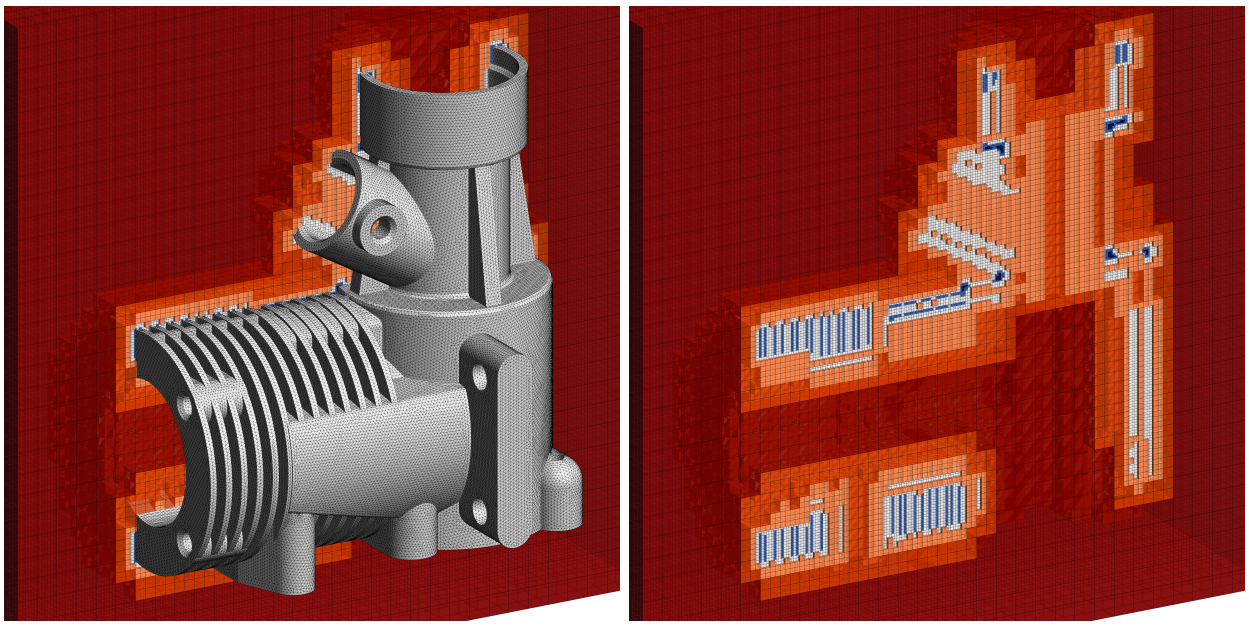


Figure 8.7 Octree after refinement: the color of each octant signifies its refinement level, from coarse (red) to fine (blue).

8.3.4 Size limitation

Large mesh size gradients may be the cause of low quality finite element solutions. In solid mechanics e.g., they may be the cause of excessive values of strains when a very small element is adjacent to a large one. One of the goals of our approach is to ensure that two adjacent edges in the mesh have their length ratio below a user-defined gradation $\alpha > 1$, that is, the length ratio along a direction follows a geometric progression. We briefly show here that such a geometric progression is achieved by limiting the size gradient along a given direction by $\alpha - 1$:

$$\left|\frac{\partial h}{\partial x}\right| \leq \alpha - 1, \tag{8.13}$$

that is, a constant gradation yields a geometric progression of edges length and not a linear progression as one could expect at first glance. To see this, let us consider a one dimensional mesh in the direction x along which the size gradient is maximal. Let us denote by $\mathbf{x}_i, i = 0, \ldots n$ the vertices of the mesh and $h_i = h(\mathbf{x_i})$ the length of the edge starting at $\mathbf{x}_i$. As the edge lengths follow a geometric progression of ratio α, we have $h_i = \alpha^i h_0$. The x-coordinate

of vertex $\mathbf{x}_n$ is given by:

$$x_n = \sum_{i=0}^{n-1} h_i = h_0 \left(1 + \alpha + \alpha^2 + \cdots + \alpha^{n-1}\right) = h_0 \frac{\alpha^n - 1}{\alpha - 1} \tag{8.14}$$

from where follows:

$$\alpha^n h_0 = h_0 + x_n(\alpha - 1). \tag{8.15}$$

Using $h_n = \alpha^n h_0$, one ends up with:

$$h_n = h(x_n) = h_0 + x_n(\alpha - 1), \tag{8.16}$$

showing that the size field $h(x)$ is a linear function of space, despite the fact that the edge lengths are in a geometric progression. The derivative $\partial h/\partial x$ is hence constant, with magnitude $\alpha - 1$. Locally, since $\alpha = h_{i+1}/h_i$, we also have the following finite difference approximation for the size gradient:

$$\left|\frac{\partial h}{\partial x}\right| = \frac{h_{i+1} - h_i}{\frac{1}{2}(h_{i+1} + h_i)} = \frac{2(\alpha - 1)}{\alpha + 1}, \tag{8.17}$$

which is close to (8.18) for typical values of $\alpha \in [1, 1.4]$. Limiting the size derivative along any direction thus amounts to limit the $2-$norm of the size gradient, that is

$$\|\nabla h(\mathbf{x})\|_2 \leq \alpha - 1. \tag{8.18}$$

A similar computation by Chen et al. [115] leads to a third condition $\|\nabla h(\mathbf{x})\|_2 \leq \ln \alpha$, also close to (8.18) for the considered range of gradation, as we have $\ln(1 + (\alpha - 1)) \simeq \alpha - 1$.

As pointed out by Persson [121], the one-dimensional analysis discussed above is not sufficient to ensure that the expected mesh size gradation condition is fulfilled all over the mesh. Indeed, when storing the size field $h(\mathbf{x})$ in a background mesh, the expected gradation is ensured only along the edges of the elements of the background mesh, and larger size variations might be encountered in the interior of the elements. To also constrain size gradients inside the elements of the background mesh, it is then neccessary to iterate over the edges of the background mesh to modify appropriately the size stored at the nodes [115, 116]. This results in a size field $h(\mathbf{x})$ that eventually satisfies (8.18), but in a background mesh dependent fashion [121]. This drawback is avoided with our methodology because the mesh size function is rather stored in a balanced octree. Condition (8.18) is satisfied everywhere in the volume, and thus also in the interior of the elements, as a result of the subdivision of the octants until convergence.

Gradient computation by finite difference. Condition (8.18) requires to limit the size derivative along all directions, which is a costly operation. In practice, we limit the size variation along the directions of the axes, i.e. the directions of the octree, and condition (8.13) is enforced using a finite difference scheme. This amounts to impose

$$\|\nabla h(\mathbf{x})\|_\infty \leq \alpha - 1, \tag{8.19}$$

a looser requirement resulting in directions along which the gradation may only be controlled up to a factor $\sqrt{3}$ in the worst case. As the results show however, limiting the $\infty-$norm yields high quality meshes for all the considered CAD models, with an average discrete size gradation matching the required gradation, see Section 3. We iterate over the octants until

$$\left|\frac{\partial h}{\partial x_k}\right| \leq \alpha - 1, \tag{8.20}$$

is verified everywhere, with $k = 1, \ldots, 3$. For each octant, we take advantage of the 2:1 balancing provided by p4est to compute the gradient with a cell-centered finite difference scheme. Since the octree is balanced, only three stencils must be considered to approximate the gradient in all cases. With the p4est terminology, one side of a face between two octants is said to be either full (F) or hanging (H), depending on whether the octant on that side is a leaf or is itself subdivided. The stencils to be considered (Fig. 8.8) are then FFF, FFH, HFH, since the all-hanging case (HHH) can be regarded as multiple all-full (FFF) stencils. To evaluate (8.20) at the center of the middle octant along, say, the x-direction, we use the Taylor expansion:

$$\left.\frac{\partial h}{\partial x}\right|_i = \frac{1}{2}\left(\frac{\bar{h}_{i+1} - h_i}{\Delta x_{i+1} + \Delta x_i} + \frac{h_i - \bar{h}_{i-1}}{\Delta x_i + \Delta x_{i-1}}\right), \tag{8.21}$$

where $i-1$, i and $i+1$ represent the three positions in the considered stencil, $\bar{h}_i$ denotes the arithmetic average of the mesh size of the four octants adjacent to the hanging face i, and Δx_i is the half-length of the octants in the position i. For the all-full stencil FFF, one has $\bar{h}_i = h_i$ and the Δx_i are all equal to the half length of the octants, so that the approximation (8.21) reduces then to the usual second-order centered scheme.

Size limitation. The computed gradient is then used to limit the mesh size field. For any two adjacent octants with gradient larger than $\alpha - 1$, the larger is reduced so as to satisfy condition (8.18). More precisely, if h_1 and h_2 denote the mesh size in the octants with $h_2 \geq h_1$, h_2 is modified as

$$h_2 = \min(h_2,\, h_1 + \Delta x(\alpha - 1)). \tag{8.22}$$

This expression is an approximation of the solution to the steady-state equation proposed

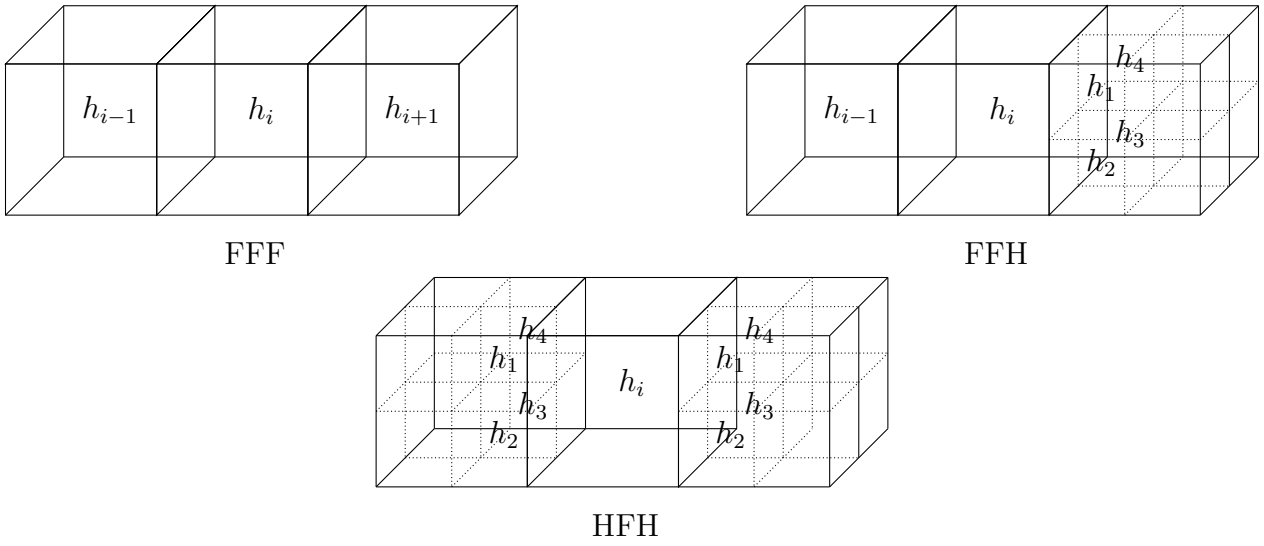


Figure 8.8 Finite difference stencils to compute ∇h in the octree.

by Persson [121] in his continuous formulation of the gradient limiting problem, ensuring mesh size limitation is propagated in the direction of increasing values. Smaller mesh sizes are left unchanged by the limitation process, preserving the sharp features of the geometric model. The limitation is performed iteratively in the three directions, until condition (8.18) is satisfied everywhere in the octree (Fig.8.9b).

8.3.5 Size query in the octree

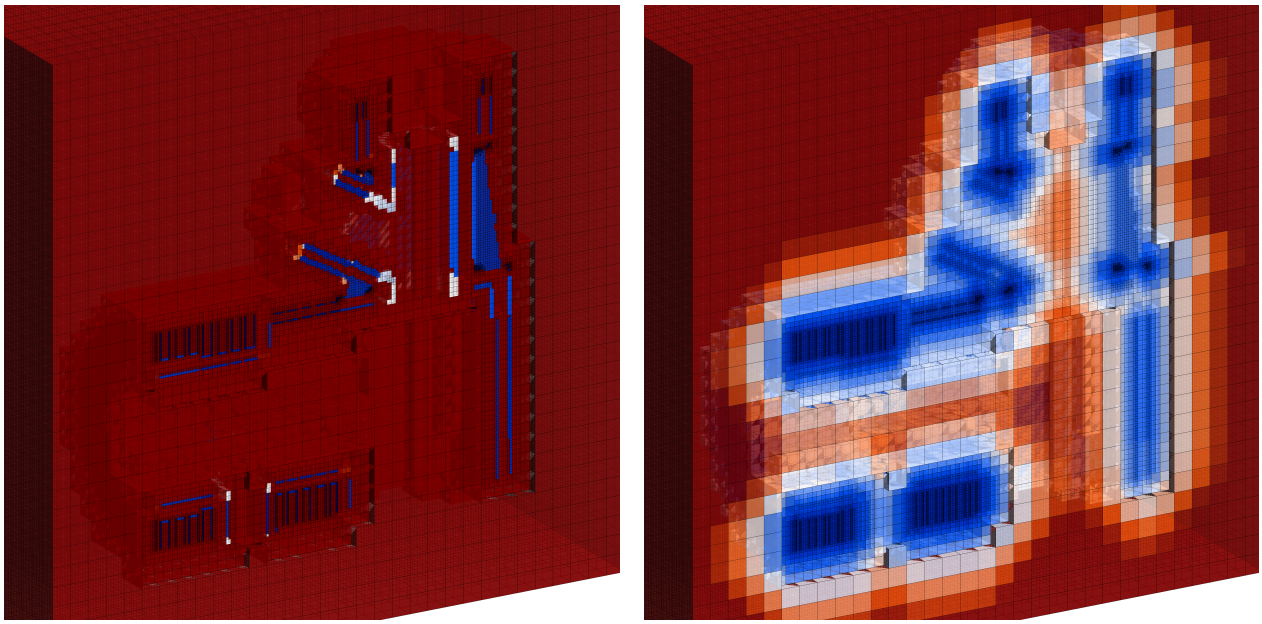
Size queries to evaluate the size at different locations on parametrized curves, surfaces and volumes during the meshing process are performed by Gmsh. The implementation of the query routines in the octree is provided by p4est. Since both the size $h(\mathbf{x})$ and its gradient are known in each octant, the mesh size at the query point $\mathbf{x}$ is evaluated using a first-order Taylor expansion:

$$h(\mathbf{x}) = h_i + \nabla h \cdot (\mathbf{x} - \mathbf{x_c}), \tag{8.23}$$

where h_i and $\mathbf{x}_c$ denote the size and the coordinates at the center of the octant, respectively.

8.4 Results

The proposed algorithm was applied to a variety of CAD models accessed from GrabCAD (https://grabcad.com/), the ABC Dataset model library [132] and Gmsh's benchmarks suite. For all geometries, the same set of parameters was used with values given in Section 1, that is, $n_d = 20$ elements in curved areas, $n_g = 4$ elements in thin layers and a mesh gradation



(a) (b)

Figure 8.9 Limiting the mesh size stored in the octants: (a) initial mesh size field computed from curvature and feature size, assigned in octants intersecting the surface mesh (b) size field after limitation.

of $\alpha = 1.1$. Note that those default parameters were chosen with what we consider to be common sense, and not after a series of trial and error: for example, discretizing a circle with 20 line segments is what is used by default to graphically represent a circle in Gmsh. The number of layers $n_g = 4$ was set with bending structures computations in mind, where it is known that having less than 4 layers leads to overestimating the stiffness of the structures. The gradation of 1.1 originates from regular discussions with CFD practitioners: this value can be set higher for computational mechanics applications for example, yet, it should not exceed 1.7-1.8, unless the generated mesh serves for visualization purposes. As shown in this section, those intuitive parameters yield quality meshes for a vast majority of the considered models. In each case, we generate a triangulation of the CAD model with a prescribed uniform mesh size. The surface mesh is the input of the algorithm to compute a mesh size field on all volumes. Surface curvatures and the approximate medial axis are then estimated, and the mesh size field $h(\mathbf{x})$ is computed as described in the previous sections. The octree is stored on disk in the native p4est format on creation, and is then loaded as a background field in Gmsh.

This sections aims at demonstrating the efficiency and accuracy of our approach. As far as efficiency is concerned, the CPU time dedicated to queries in the octree is distinguished from

the time devoted to the construction of the octree. We then compare the sum of those two times to the meshing time. In particular, we show that the proposed size field construction does not change the order of magnitude of the meshing time, i.e., seconds remain seconds and do not become minutes. We then analyze how accurately the prescribed parameter are reached in the mesh generated from the size field. To this end, a discrete counterpart of the gradation is compared to the parameter α, and the number of elements in small gaps is checked to be close to n_g. An efficiency index is then introduced for each of the generated mesh: this index measures how close the resulting mesh is to a unit mesh and thus expresses the adequacy between the size field and the mesh.

All computations are run on a laptop with Intel Core i7 8750h CPU (2.2 GHz) and 16Gb memory, and execution times for two selected test cases can be found in the table hereafter.

8.4.1 Surface and volume meshing

Engine block. We first consider the engine block test case depicted in Fig. 8.2a. The geometry, while presenting a relatively low (≈ 500) number of faces, features entities of variable radii of curvature as well as thin areas, and is an interesting application example for our algorithm. The CAD model being an assembly of thin components, such as narrow fins surrounding the main cylinder, the feature mesh size is the dominant meshing criterion. Surface meshes were presented in Section 2, and show the impact of the feature mesh size (Fig. 8.6) on the final mesh density. On the left part of Figure 8.6, the size field is constructed based only on curvature: one or two elements are generated in the narrow fins, whose mesh size is defined by the radius of curvature of the inner cylinder. On the right part of Figure 8.6, the feature mesh size h_c is also considered in the construction of the size field, and $n_g = 4$ element layers are then as expected generated in the fins. Figure 8.10 shows the tetrahedral mesh, where prescribed mesh size specifications are respected inside the volume. Since the spurious branches of the medial axis are removed in the rounded corners, mesh size in these locations is constrained by curvature. This results in a smaller size defined by the node density n_d (zooms 2 and 3). Mesh size in the thin cylinders and the fins is computed to fit 4 layers of elements (zooms 1, 4, 5, 6 and 7).

Sport bike engine. The second test case is a four-cylinder engine of a Honda™ CBR600F4i sport bike[1]. The model is composed of thin areas such as pipes, gears and plates, and of regions of high curvature. The enclosing volume being not conformal (i.e., not "air-tight"), a volume mesh could not be generated without first repairing the CAD. Only surface meshes

[1] `https://grabcad.com/library/honda-cbr600-f4i-engine-1`

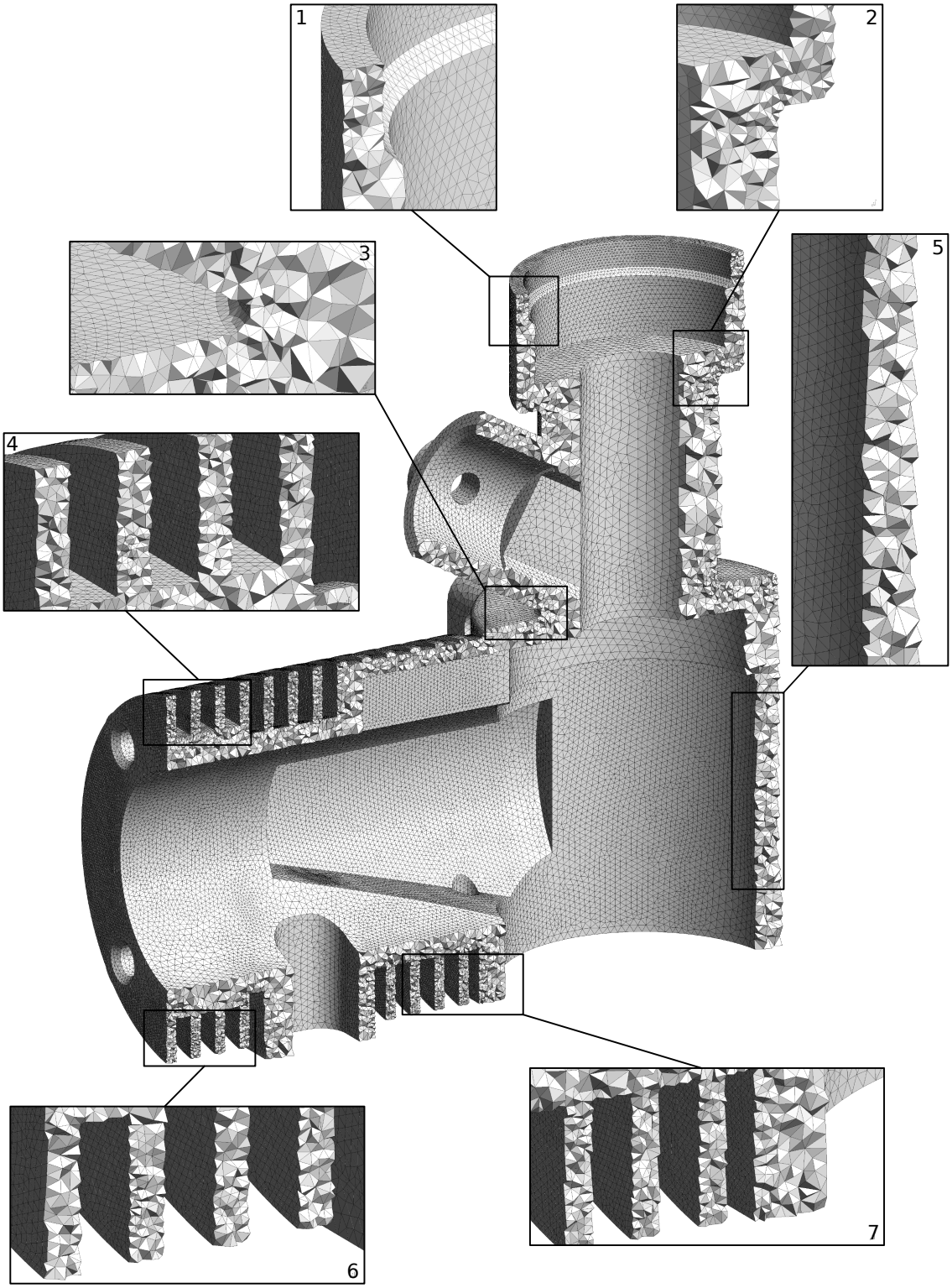


Figure 8.10 Final volume mesh of our worked-out example: the size field is computed with $n_d = 20$ nodes on the local osculating circles and $n_g = 4$ layers of elements in all geometric features. The mesh contains 583,776 nodes and 2,716,170 tetrahedra.

are thus presented for this example. The resulting surface meshes accurately capture areas of higher curvature (Fig. 8.11, left) as well as small features (Fig. 8.11, right and Fig. 8.12), demonstrating the robustness of the algorithm on large CAD models.

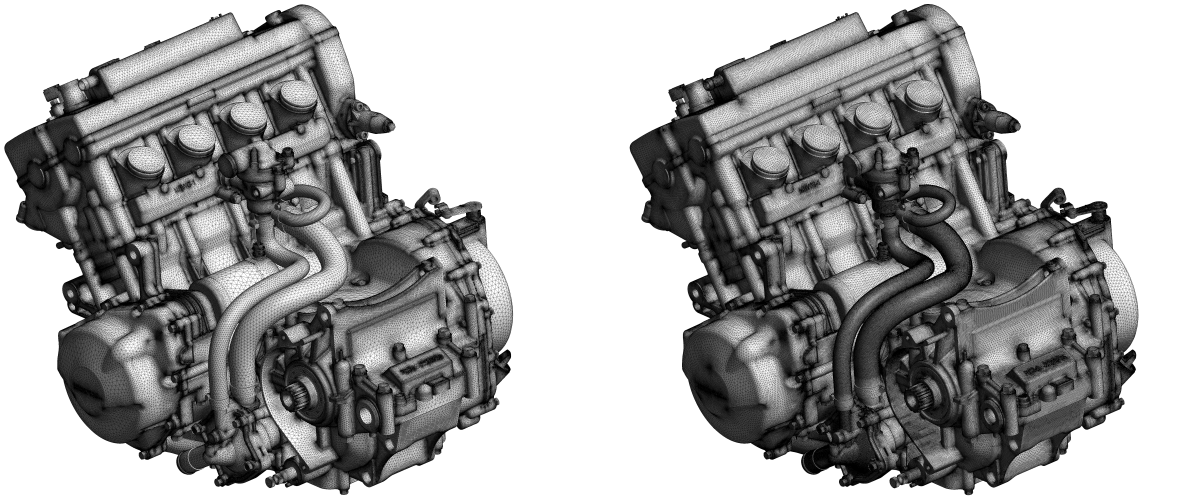


Figure 8.11 The surface mesh based only on curvature contains 1,521,410 nodes and 3,042,060 triangles (left); the mesh based on both curvature and feature size contains 2,302,630 nodes and 4,605,010 triangles (right).

ETA 6497-1 watch movement. This geometry[2] is a complete clockwork composed of flat plates and gears of different sizes. Feature size is the constraining factor on most flat pieces, while curvature determines the mesh size at e.g. gear teeth. The surface mesh and the volume mesh for this test case are respectively shown on Fig. 8.13 and 8.14.

Space shuttle. This example was selected to illustrate specifically the influence of the node density n_d and of the gradation parameter α on the mesh. To show their influence on the whole model, including thin areas, feature size was not considered in this case, and only curvature was taken into account. As expected, larger values of n_d result in more elements on the local osculating circle to a surface, and hence in a finer mesh (Fig. 8.15). Similarly, a gradation parameter α close to 1 limits the size gradient, also resulting in a globally finer and more homogeneous mesh (Fig. 8.16).

[2] `https://grabcad.com/library/eta-6497-1-complete-watch-movement`

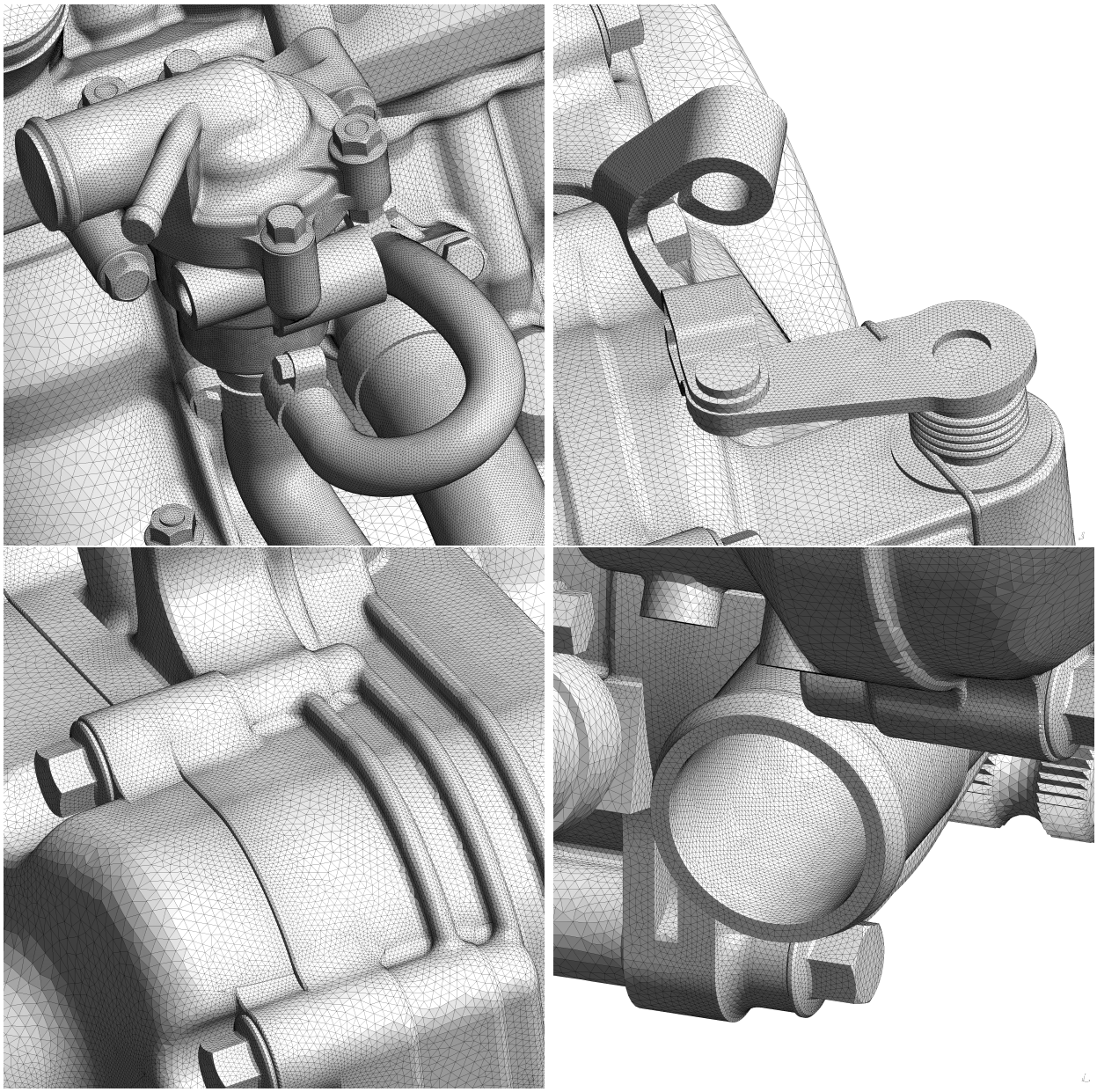


Figure 8.12 Zooms on selected parts of the Honda engine.

Various geometries. Finally, our algorithm was also applied with the same parameters to a series of CAD model obtained from GrabCAD, the ABC Dataset library and the Gmsh benchmarks library (Fig. 8.19). In all cases, accurate size fields were computed within a few seconds to a few minutes, yielding smooth meshes suitable for numerical simulations. Note that feature size computation was not enabled for all models (e.g. for the lava lamp).

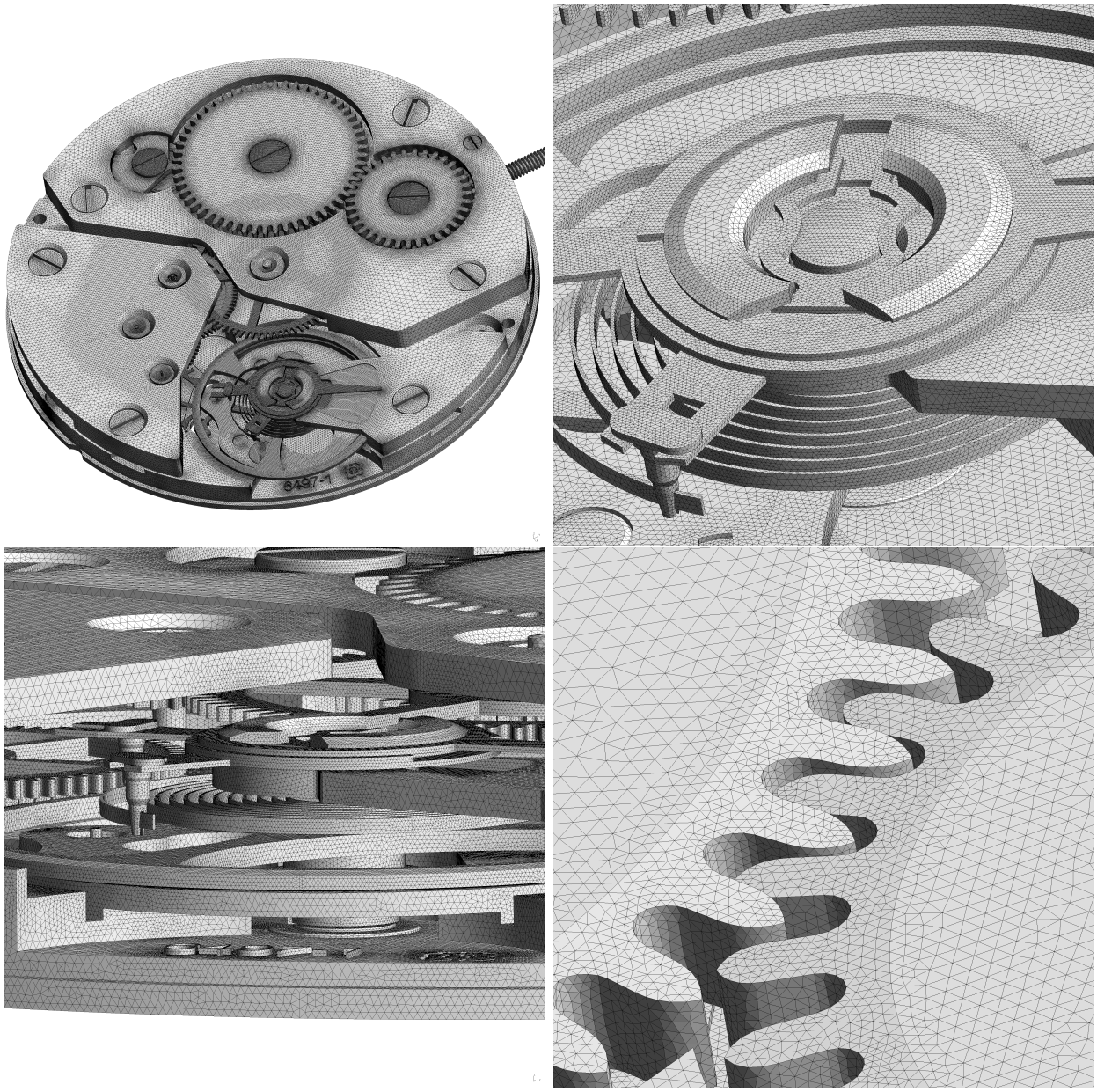


Figure 8.13 Zooms on selected parts of the surface mesh of the watch movement.

8.4.2 Adequacy between the mesh and the size field

With the generated mesh at-hand, we define two quantitative estimates to assess its adequacy with the size field: an *efficiency index* measures how far the mesh is from a unit mesh in the isotropic metric field described by the size field, and a discrete counterpart to the gradation verifies that edge length progression is limited in the mesh. Finally, a standard isotropic quality indicator is presented.

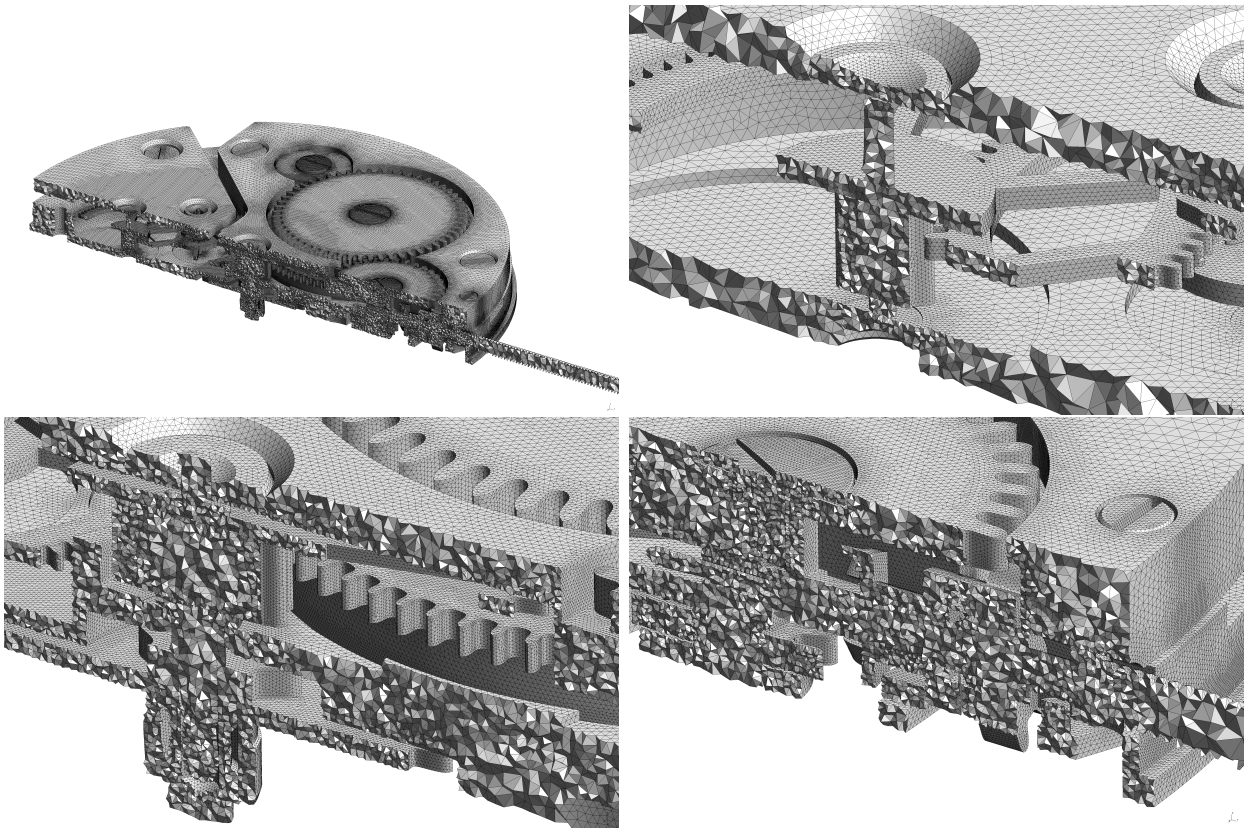
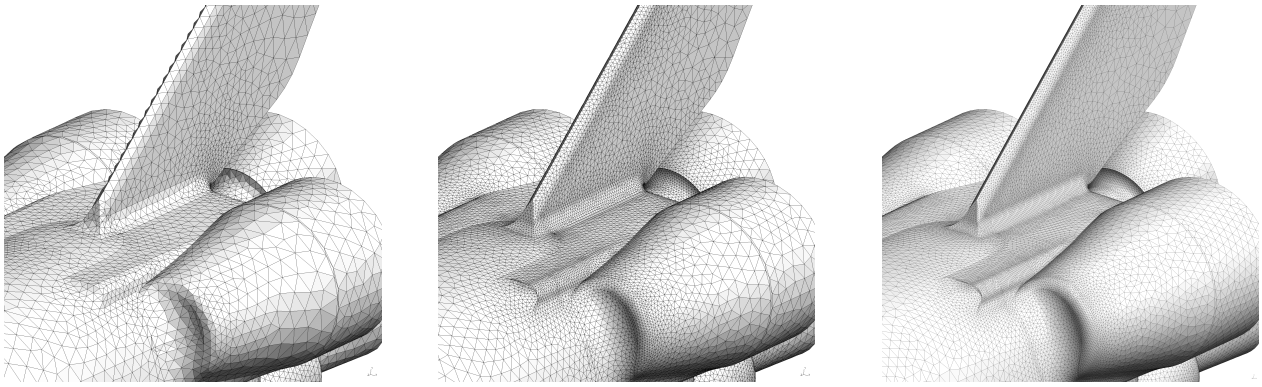


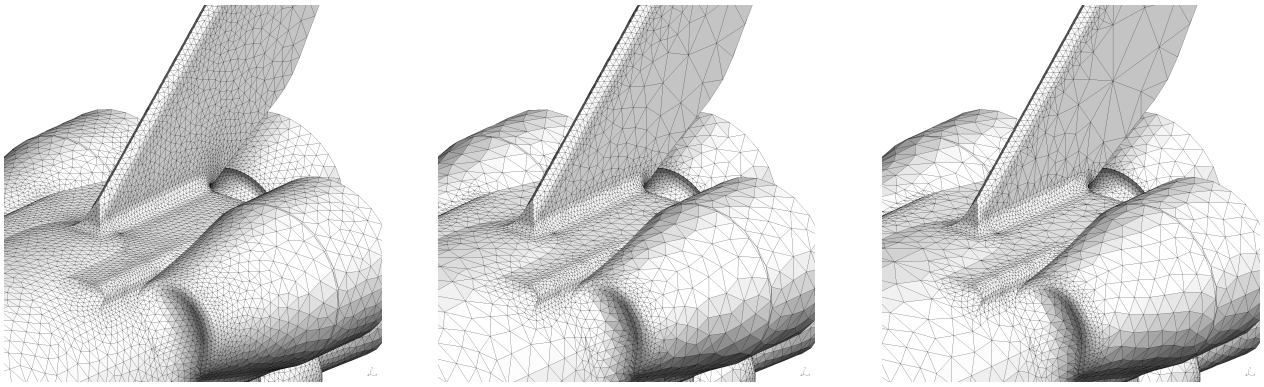
Figure 8.14 Zooms on selected parts of the volume mesh of the watch movement.



(a) $n_d = 10$ (b) $n_d = 30$ (c) $n_d = 50$

Figure 8.15 Influence of the node density n_d on the final mesh for a gradation of $\alpha = 1.1$.

Efficiency index. The resulting mesh should be as close as possible as a unit mesh in the metric field associated to the isotropic sizing function [64], i.e. the edges lengths should be as close as 1 when measured in the given metric. Since we are dealing with the discrete counterpart of this continuous mesh framework, we consider that the size specification is



(a) $\alpha = 1.1$ (b) $\alpha = 1.4$ (c) $\alpha = 1.8$

Figure 8.16 Influence of the gradation α on the final mesh for a node density of $n_d = 20$.

respected when the mesh is quasi-unit, that is, when the edges lie in $[\frac{1}{\sqrt{2}}, \sqrt{2}]$. To evaluate the adequacy between the generated mesh and the prescribed size field, we define an *efficiency index*. Let l_i be the length of an edge i computed in the local metric and n_e the number of edges in the mesh. The efficiency index τ of a mesh is defined as the exponential to one of all the edges length in the mesh, that is:

$$\tau = \exp\left(\frac{1}{n_e}\sum_{i=1}^{n_e}\bar{l}_i\right), \tag{8.24}$$

with $\bar{l}_i = l_i - 1$ if $l_i < 1$ and $\bar{l}_i = 1/l_i - 1$ if $l_i \geq 1$. The efficiency index thus lies in $]0, 1]$, the upper bound being reached for a unit mesh. We run our algorithm on close to two hundred CAD models and computed the efficiency index (Fig. 8.17), whose median is slightly above 0.8. This shows that the size field given to the meshing tool is realistic, as it can generate almost unit edges for the given metric field.

Discrete gradation. The resulting mesh should feature a geometric progression in adjacent edges lengths with a ratio up to α. Instead of monitoring the maximum edge ratio at every vertices of the final mesh, we use a looser indicator defined on the edges. The *discrete gradation* $\alpha_{d,i}$ measures the progression between the average edges lengths at both vertices of each edge e_i. Let v_1 and v_2 denote the vertices of e_i and $l_{avg}(v)$ the average edge length at vertex v. We define the discrete gradation at edge e_i as follows:

$$\alpha_{d,i} = \frac{\max[\, l_{avg}(v_1), l_{avg}(v_2)\,]}{\min[\, l_{avg}(v_1), l_{avg}(v_2)\,]} \tag{8.25}$$

In the same way that the efficiency index is a global indicator for a given mesh, we define

an average discrete gradation as the average of $\alpha_{d,i}$ over all edges of the mesh. The average discrete gradation should be close to the user-defined gradation α. In our results over the same sample of meshes, the discrete gradation is close to the required gradation, although it lies slightly above for a variety of volume meshes as a consequence of our choice to limit $\|\nabla h\|_\infty$ instead of $\|\nabla h\|_2$ (Fig. 8.18).

Quality measure. In addition to the measures of adequacy, element-wise mesh quality is also evaluated using the ratio γ of the radius r of the sphere inscribed in a tetrahedron to the radius R of its circumscribing sphere:

$$\gamma = \frac{3r}{R}. \tag{8.26}$$

This ratio varies from 1 for a regular tetrahedron, and approaches 0 for sliver tetrahedra. The quality distribution for 50 meshes is shown on Fig. 8.17: these meshes exhibit high-quality elements with $\gamma = 0.8$ on average. The current volume meshing algorithm [106] of Gmsh performs mesh optimization if the quality of an element is below 0.4, hence the very reduced number of elements below this threshold.

8.4.3 Execution time

The execution times for the engine block and the watch movement are given in Table 8.1. The mesh size field time is split between the main steps of the algorithm: (i) insert the bounding boxes of the triangles of the surface mesh in the RTree, (ii) compute the approximated curvature with a least square method, (iii) compute the medial axis of the geometry, (iv) initialize the root octant and refine the octree and (v) limit the size gradient to $\alpha - 1$. Computing the medial axis requires the Delaunay tetrahedrization of the points from the surface mesh: this step is detailed and included in the time associated to the medial axis computation. The step "Others" includes all the secondary steps, such as splitting the mesh into faces before computing curvature.

To limit the size gradient, we iterate over the octants until condition (8.12) is satisfied in all three directions. This step is dependent on the depth of the octree, hence on the minimum size h_{min}, the node density n_d and the curvature of the surface mesh given as input. A highly curved region of the model due to a poor resolution of the surface mesh will result in high density in the octree, increasing the size limitation time. This can generally be circumvented by first generating a slightly refined surface mesh, which can be computed quickly and will rule out these extreme curvature magnitudes. Computation time for the medial axis, on the other hand, is linear with the number of nodes of the input mesh: this is indeed observed in

Table 8.1.

Table 8.1 Execution times

	Block		Watch movement	
	Time (s)	%	Time (s)	%
Insert surface mesh in RTree	0.47	3.0 %	2.30	1.2 %
Compute curvature	0.36	2.3 %	1.13	0.6 %
Compute medial axis	7.19	45.7 %	64.06	34.1 %
incl. Delaunay tetrahedrization of surf. mesh	0.61	3.9 %	2.33	1.2 %
Create and refine octree	0.96	6.1 %	20.24	10.8 %
Limit size gradient	5.85	37.1 %	93.89	50.0 %
Others	0.69	4.3 %	6.07	3.2 %
Total	15.75	100.0 %	187.69	100.0 %
Mesh 1D entities	2.48	2.5 %	38.51	4.5 %
Mesh 2D entities	25.68	25.6 %	355.58	41.0 %
Mesh 3D entities	72.12	71.9 %	469.94	54.4
Total	100.3	100.0 %	864.03	100.0 %
including Size queries	65.13	64.9 %	483.53	56.0 %
# of curves in the CAD model	1584		12,991	
# of faces in the CAD model	533		4,760	
# of nodes in input surface mesh	145,024		654,348	
# of elements in input surface mesh	290,116		1,308,870	
# of octants	670,566		9,052,065	
# of triangles in final surface mesh	573,808		3,629,000	
# of tetrahedron in final volume mesh	2,716,170		12,273,300	

8.5 Conclusion

A methodology to automatically generate an accurate size field, storing the sizing information in an octree, was presented. This tool eliminates the tedious operation that is assigning by hand the mesh size on all geometric entities of a CAD model, thus saving considerable time. This algorithm was a missing piece of the standard meshing pipeline of Gmsh, and was designed to become the default tool for size specification going onward. Five user parameters are required to generate a mesh suitable for numerical simulations, two of which being the minimum and the maximum, or *bulk*, size, whose value is assigned based solely on the characteristic dimension of the CAD model. This leaves the user with only three parameters in order to tune the mesh density. Special care was given to the small features of the geometry: through an approximate medial axis computation, we ensure multiple layers of elements are always generated in narrow regions. This is particularly useful in simulations where Dirichlet boundary conditions are applied. To illustrate our size field computation,

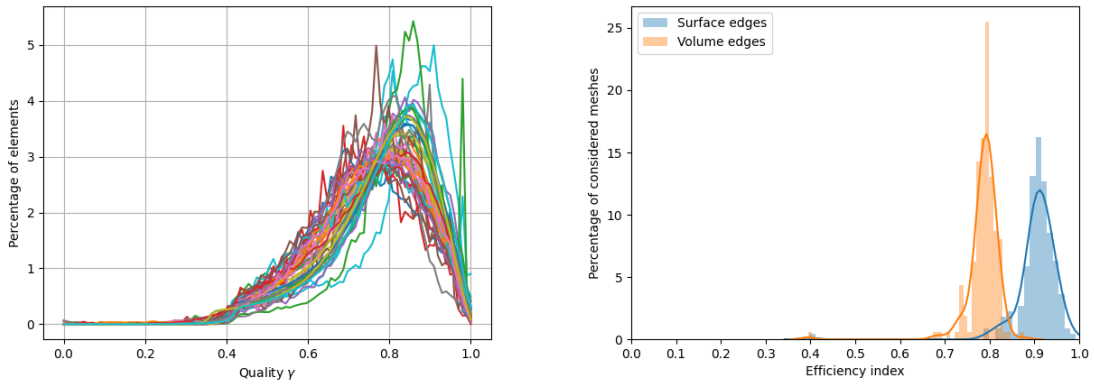


Figure 8.17 Left: elements quality for 50 meshes. Right: efficiency index for a sample of 190 meshes.

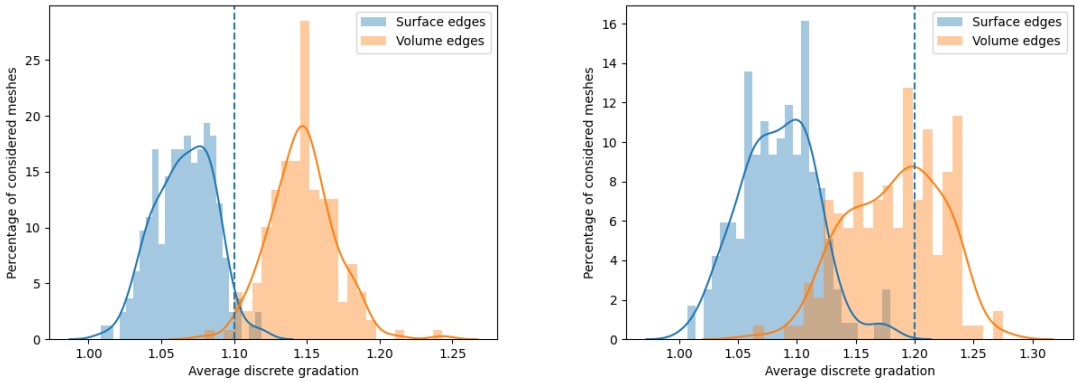


Figure 8.18 Left: discrete gradation for a sample of 190 meshes for a prescribed $\alpha = 1.1$. Right: $\alpha = 1.2$

we applied our algorithm on a large variety of CAD models: for all the test cases, high-quality surface and volume meshes were obtained in a robust and automatic fashion, and are adapted to the features of the geometric model. The adequacy between the size fields and the meshing algorithm was demonstrated: the generated meshes are unit or quasi-unit according to the prescribed isotropic metric field, and the desired mesh gradation is met. In thin areas of the CAD model, the required number of layers of elements is present in the final mesh. More importantly, generated meshes are high-quality meshes ready to be used for numerical simulations. Emphasis was placed on selecting the same set of parameters for all the geometries: parameters that *make sense* to the user do indeed yield a suitable mesh for a given geometry, making this tool intuitive and user-friendly. Our choice was to use the p4est library to implement the background tree structure: although it adds an external dependence, the octree routines are efficient and a parallel implementation is available for

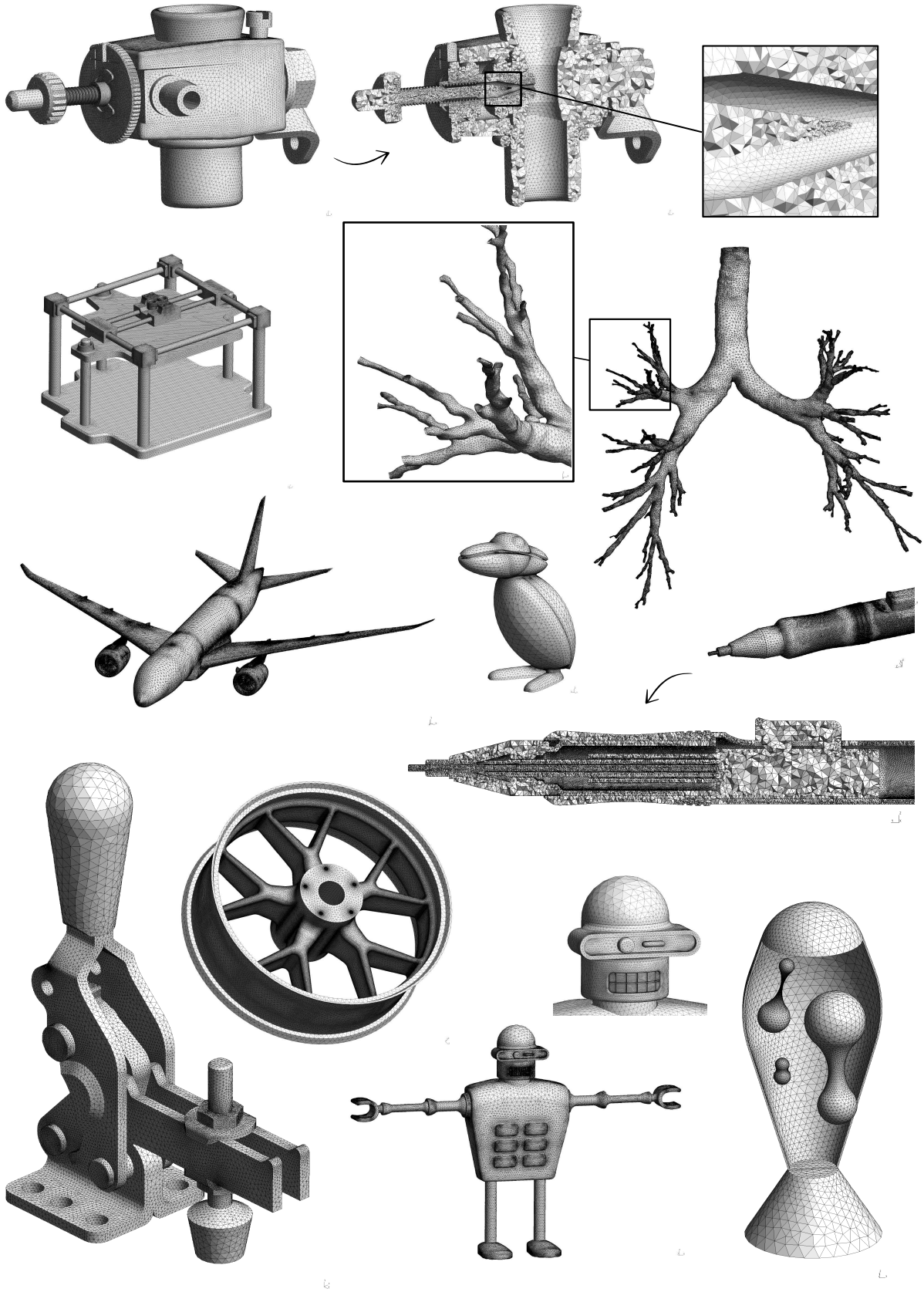


Figure 8.19 Application of the size field on various geometries from GrabCAD, the ABC Dataset and the Gmsh benchmarks.

further improvements. Future work will focus on extending to anisotropic size prescription, as well as coupling this geometry-based size field with numerical solutions post-processing results, to include *a posteriori* error estimation in the size field design. In this sense, this tool will assist users performing mesh adaptation, storing the error-based metric field while providing size prescription based on the CAD throughout the adaptation process.

CHAPTER 9 CONCLUSION AND PERSPECTIVES

This thesis was motivated by a simple interpolation problem, namely: given a unit square and a smooth function f, how can we compute a $\mathcal{P}^2$ triangulation with a given number of vertices that minimizes the interpolation error $\|f - \Pi^k f\|_{L^2}$? To tackle this problem, we addressed the generalization of key ingredients of the continuous mesh framework to high-order simplices. This thesis built upon recent work on curvilinear mesh adaptation [17, 23] to derive broader interpolation error estimates, and introduced new notions in the form of a definition of unitness. The main results are here recalled:

Error estimate on high-order triangles Working in the reference space, we introduced in Chapter 4 an interpolation error estimate that extends the one for linear interpolation presented in [17]. Instead of bounding the derivatives of the field of interest in the physical space, the derivatives of its composition with the nonlinear reference-to-physical transformation are bounded in the reference space instead. This translates into additional terms involving both the metric derivatives and the lower-order derivatives of the field, which is typical of high-order elements.

Unit isometric simplices Unit elements denote elements which are indistinguishable with respect to their metric. This concept is central to the continuous mesh framework, as it describes which simplices are considered ideal by the metric. Up to a rigid rotation, linear simplices are completely determined by their edges, thus the classical requirement that all edges should be unit is sufficient to generate unit and quasi-unit meshes. Curved elements can exhibit arbitrary edge parameterization, so a more restrictive definition is required. We introduced in Chapter 4 a definition of unit simplices based on Riemannian isometries. These transformations are isometries in the usual sense, but also preserve the Riemannian metric. Being isometric is a strict requirement that ensures that lengths are preserved globally, but also locally. Key properties are consequently preserved: inner angles, area, edge length, energy, and geodesicity of the edge parameterizations. From this local property, the usual definition for linear elements is recovered, as well as the definition in terms of the Jacobian matrix proposed in [17]. Quasi-unitness is defined in the form of quasi-isometric elements, which have controlled inner angles and edge parameterization in addition to edge length. This constructive definition was applied in Chapter 7 to generate quasi-unit $\mathcal{P}^2$ triangulations.

Metrics for error control with $\mathcal{P}^2$ triangles For linear meshes, an analytical optimal metric minimizing the interpolation error was proposed almost twenty years ago by Loseille and Alauzet [1,8,9], and has since been extended to handle high-order interpolation. Continuous error estimates on e.g. $\mathcal{P}^2$ meshes are very recent [17], and deriving the optimal metric on these meshes is still an open problem. The contribution of Chapter 6 is twofold. First, we analyzed an ad hoc Riemannian metric based on the metric induced by the graph of the field of interest. Interpolation error estimates along quadratic curves were used to limit the principal sizes to a threshold. This metric is suboptimal for linear meshes, but predicts close to second-order convergence on the examined test cases for linear interpolants on quadratic triangles. In particular, the pessimistic results from [47, 48] are avoided.

Second, to work towards obtaining the optimal metric, we applied the rationale of the log-simplex method [11, 12] to the error estimate extended in Chapter 4 to arbitrary order interpolation on $\mathcal{P}^2$ triangles. This yields a nonlinear system of PDEs for the metric components, as metric derivatives are involved in the error estimate. As a first attempt to provide a solution, we proposed a naive but straightforward extension of the log-simplex iterative scheme to include metric derivatives.

Generation of quasi-isometric triangulations In Chapter 7, the frontal meshing algorithm inherited from Zhang's work was modified to control the inner angles of the triangles and create quasi-isometric elements following the definitions introduced in Chapter 4. $\mathcal{P}^2$ meshes are generated by curving a high-quality linear mesh, optimized in a first time for the metric-based distortion. The ability to generate quasi-unit curved triangulations was illustrated on analytical fields and on the academic lid-driven cavity problem. Quality indicators show that the meshes respect the prescribed criteria, in terms of edge length, inner angles, and distortion-based quality.

Automatic isotropic size field for 3D models Lastly, a practical tool, aimed at the generation of an initial mesh rather than adaptation, was developed and implemented in Gmsh. Given a 3D CAD model, it computes a smooth isotropic size field constrained by the curvature of the surface mesh and its distance to the medial axis. The size field is stored in an octree for fast access. The objective is to ease the meshing process by guaranteeing a good initial mesh with a reduced number (5) of user parameters. Aside from $h_{\min}$ and $h_{\max}$, the user is expected to provide the required node density in curved regions, the number of layers in thin gaps to avoid locking phenomena, and the desired bound on the size variation. The robustness and efficiency of this tool were illustrated on a wide variety of CAD models.

9.1 Future work

Despite metric-based curvilinear mesh adaptation being a recent research topic, significant work was achieved in the last years to push it towards becoming a competitive alternative to state-of-the-art anisotropic meshing. We believe a significant ingredient was described in this thesis, by proposing a broader definition of unitness. The following items discuss bigger or smaller challenges which should be addressed to provide optimal curvilinear meshes for the interpolation error in the short or medium term, starting with the derivation of the optimal Riemannian metric:

Solving the optimal metric problem The principal theoretical point concerns solving the problem discussed in Chapter 6 to provide the optimal metric for the interpolation error on $\mathcal{P}^2$ meshes. In this regard, our conclusions are the same as in [17], as it is not clear if this problem can be solved analytically. More specific error estimates, not relying on a quadratic form bounding the high-order derivatives, could also be derived, leading to a different problem altogether from the one discussed in Chapter 6.

Characterization of the involved manifolds An in-depth understanding of the manifolds at play would solve some remaining interrogations. For instance, the eigenvalues and eigenvectors of the metric are central to metric-based frameworks, however, as bilinear forms, these quantities are not meaningful for metric tensors in general. This is not a problem in practice, as the mesh is generated in the Euclidean space with a fixed basis, but it could help characterize the difference between the trajectories obtained by following their integral curves and the geodesics. Similarly, we wondered at the end of Chapter 5 in which conditions the optimal metric for the interpolation error could be induced by the graph of a real-valued function. If this function exists, embedding techniques as in Caplan et al. [30] would create uniform triangulations on the graph, consisting of elements endowed with equal error, yielding projected meshes with control over the interpolation error on the original field.

Long edges and slivers These two aspects are not treated in our current pipeline and can negatively impact the error in the final mesh, even if it is adapted to the optimal metric. Slivers could be handled by a cavity operator [43]. This requires reworking a sizable portion of the meshing procedure. Long edges could be split by inserting additional vertices after the sampling. In a one-dimensional approximation, the frontal algorithm creates long edges when two meeting vertices are at distance $\ell \in [\ell_{\max} + \epsilon, 1 + \ell_{\min} - \epsilon]$ from each other, forbidding the insertion between them of a new vertex at unit distance from either. Forcing the insertion of

a vertex at half distance would yield two edges with length in $[(\ell_{\max}+\epsilon)/2, (1+\ell_{\min}-\epsilon)/2]$, which is acceptable when targeting equilateral triangles.

3D mesh generation The definition of unit isometric simplices covers tetrahedra, however the meshing procedure was presented for triangles only. The limitation comes from our finite element solver, which time did not allow us to extend to 3D. Although going to 3D naturally comes with its lot of intricacies, some steps are expected to be straightforward to extend: vertex sampling at prescribed angles, point cloud tetrahedralization, edge curving to minimize length or distortion. Already problematic aspects, such as long edges and slivers, are expected to come in more flavours in three dimensions.

Complete integration in Gmsh As a more practical aspect, the metric interpolation routines still have to be integrated into Gmsh to offer a curvilinear meshing pipeline independent of our solver. This is not expected to be a problem, as Gmsh handles tensor-valued data on high-order meshes and already ships with the `Eigen` library, the only dependency used in our code to interpolate the metric and its derivatives.

Various optimizations Finally, various compromises were made, always as a trade-off between cost and accuracy: integration scheme along the chosen set of directions, exclusion zone for a vertex in the RTree for collision detection, precision of the length computation between neighbours, number of quadrature points in the various length, area, and distortion integrals, etc. These are required as soon as the metric varies, to keep the mesh generation tractable. Quantifying which choices, if any, are critical would help speed up the process. Lurking behind all these computations are the size queries, that is, metric interpolation and point localization in the background mesh. Location hints and multiple points queries come to mind to speed up these queries.

REFERENCES

[1] 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." Ph.D. dissertation, Université Pierre et Marie Curie-Paris VI, 2008.

[2] W. G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M.-G. Vallet, "Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent cfd. part i: General principles," *International Journal for Numerical Methods in Fluids*, vol. 32, no. 6, pp. 725–744, 2000.

[3] L. Formaggia, S. Micheletti, and S. Perotto, "Anisotropic mesh adaptation in computational fluid dynamics: application to the advection–diffusion–reaction and the stokes problems," *Applied Numerical Mathematics*, vol. 51, no. 4, pp. 511–533, 2004.

[4] A. Loseille, A. Dervieux, and F. Alauzet, "Fully anisotropic goal-oriented mesh adaptation for 3d steady euler equations," *Journal of computational physics*, vol. 229, no. 8, pp. 2866–2897, 2010.

[5] P. Miron, J. Vétel, A. Garon, M. Delfour, and M. El Hassan, "Anisotropic mesh adaptation on lagrangian coherent structures," *Journal of Computational Physics*, vol. 231, no. 19, pp. 6419–6437, 2012.

[6] A. Fortin, T. Briffard, and A. Garon, "A more efficient anisotropic mesh adaptation for the computation of lagrangian coherent structures," *Journal of Computational Physics*, vol. 285, pp. 100–110, 2015.

[7] F. Alauzet and A. Loseille, "A decade of progress on anisotropic mesh adaptation for computational fluid dynamics," *Computer-Aided Design*, vol. 72, pp. 13–39, 2016.

[8] A. Loseille and F. Alauzet, "Continuous mesh framework part i: well-posed continuous interpolation error," *SIAM Journal on Numerical Analysis*, vol. 49, no. 1, pp. 38–60, 2011.

[9] ——, "Continuous mesh framework part ii: validations and applications," *SIAM Journal on Numerical Analysis*, vol. 49, no. 1, pp. 61–86, 2011.

[10] E. C. Mbinky, "Adaptation de maillages pour des schémas numériques d'ordre très élevé," Ph.D. dissertation, Université Pierre et Marie Curie-Paris VI, 2013.

[11] O. Coulaud and A. Loseille, "Very high order anisotropic metric-based mesh adaptation in 3d," *Procedia engineering*, vol. 163, pp. 353–365, 2016.

[12] O. Coulaud, A. Loseille, and P. Schrooyen, "Anisotropic mesh adaptation for high-order finite elements spaces with the log-simplex method. application to discontinuous galerkin methods," *Journal of Computational Physics*, vol. 501, p. 112774, 2024.

[13] P.-E. Bernard, J.-F. Remacle, and V. Legat, "Boundary discretization for high-order discontinuous galerkin computations of tidal flows around shallow water islands," *International Journal for Numerical Methods in Fluids*, vol. 59, no. 5, pp. 535–557, 2009.

[14] J. P. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. J. Mavriplis, "Cfd vision 2030 study: a path to revolutionary computational aerosciences," Tech. Rep., 2014.

[15] A. Johnen, J.-F. Remacle, and C. Geuzaine, "Geometrical validity of curvilinear finite elements," *Journal of Computational Physics*, vol. 233, pp. 359–372, 2013.

[16] T. Toulorge, C. Geuzaine, J.-F. Remacle, and J. Lambrechts, "Robust untangling of curvilinear meshes," *Journal of Computational Physics*, vol. 254, pp. 8–26, 2013.

[17] L. Rochery, "High-order metric-based anisotropic mesh adaptation." Ph.D. dissertation, Université Paris Saclay, 2023.

[18] M. Fortunato and P.-O. Persson, "High-order unstructured curved mesh generation using the winslow equations," *Journal of Computational Physics*, vol. 307, pp. 1–14, 2016.

[19] D. Moxey, D. Ekelschot, Ü. Keskin, S. J. Sherwin, and J. Peiró, "High-order curvilinear meshing using a thermo-elastic analogy," *Computer-Aided Design*, vol. 72, pp. 130–139, 2016.

[20] E. Ruiz-Gironés, X. Roca, and J. Sarrate, "High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3d cad representation," *Computer-Aided Design*, vol. 72, pp. 52–64, 2016.

[21] R. Zhang, A. Johnen, and J.-F. Remacle, "Curvilinear mesh adaptation," in *International meshing roundtable*. Springer, 2018, pp. 57–69.

[22] R. Zhang, A. Johnen, J.-F. Remacle, F. Henrotte, and A. Bawin, "The generation of unit p2 meshes: error estimation and mesh adaptation," *International Meshing Roundtable (virtual)*, pp. 1–13, 2021.

[23] R. Zhang, "Metric-based curvilinear mesh generation and adaptation," Ph.D. dissertation, UCLouvain, 2022.

[24] L. Rochery and A. Loseille, "P2 cavity operator and riemannian curved edge length optimization: a path to high-order mesh adaptation," in *AIAA Scitech 2021 Forum*, 2021, p. 1781.

[25] G. Aparicio-Estrems, A. Gargallo-Peiró, and X. Roca, "Defining a stretching and alignment aware quality measure for linear and curved 2d meshes," *27th International Meshing Roundtable 27*, pp. 37–55, 2019.

[26] ——, "High-order metric interpolation for curved r-adaption by distortion minimization," in *Proceedings of the 2022 SIAM international meshing roundtable*. Zenodo, 2022, pp. 1–12.

[27] G. Aparicio-Estrems, "Metric-aware optimization of high-order meshes for curved adaptivity," Ph.D. dissertation, Universitat Politècnica de Catalunya, 2023.

[28] D. Sanjaya, "Towards automated, metric-conforming, mesh optimization for high-order, finite-element methods," Ph.D. dissertation, 2019.

[29] D. P. Sanjaya, K. Fidkowski, and S. M. Murman, "Comparison of algorithms for high-order, metric-based mesh optimization," in *AIAA SciTech 2020 Forum*, 2020, p. 1141.

[30] P. C. Caplan, R. Haimes, and X. Roca, "Isometric embedding of curvilinear meshes defined on riemannian metric spaces," *27th International Meshing Roundtable 27*, pp. 23–36, 2019.

[31] M. Yano and D. L. Darmofal, "An optimization-based framework for anisotropic simplex mesh adaptation," *Journal of Computational Physics*, vol. 231, no. 22, pp. 7626–7649, 2012.

[32] L. Rochery and A. Loseille, "Fast high-order mesh correction for metric-based cavity remeshing and a posteriori curving of p2 tetrahedral meshes," *Computer-Aided Design*, vol. 163, p. 103575, 2023.

[33] L. Rochery, M. C. Galbraith, D. L. Darmofal, and S. Allmaras, "A generalized continuous mesh framework for explicit mesh curving," in *AIAA SCITECH 2024 Forum*, 2024, p. 0787.

[34] B. Lévy and N. Bonneel, "Variational anisotropic surface meshing with voronoi parallel linear enumeration," in *Proceedings of the 21st international meshing roundtable*. Springer, 2013, pp. 349–366.

[35] F. Dassi, P. Farrell, and H. Si, "An anisoptropic surface remeshing strategy combining higher dimensional embedding with radial basis functions," *Procedia engineering*, vol. 163, pp. 72–83, 2016.

[36] R. Bois, M. Fortin, and A. Fortin, "A fully optimal anisotropic mesh adaptation method based on a hierarchical error estimator," *Computer Methods in Applied Mechanics and Engineering*, vol. 209, pp. 12–27, 2012.

[37] A. Fortin, T. Briffard, L. Plasman, and S. Léger, "An anisotropic mesh adaptation method based on gradient recovery and optimal shape elements," 2024.

[38] J.-M. Mirebeau, "Approximation adaptative et anisotrope par éléments finis théorie et algorithmes." Ph.D. dissertation, Citeseer, 2010.

[39] H. A. Carson, "Provably convergent anisotropic output-based adaptation for continuous finite element discretizations," Ph.D. dissertation, Massachusetts Institute of Technology, 2020.

[40] A. Huang, "An adaptive variational multiscale method with discontinuous subscales for aerodynamic flows," Ph.D. dissertation, Massachusetts Institute of Technology, 2020.

[41] A. Bawin, A. Garon, and J.-F. Remacle, "Optimally convergent isoparametric $\mathcal{P}^2$ mesh generation," in *International Meshing Roundtable*. Springer, 2023, pp. 373–395.

[42] A. Loseille, "Metric-orthogonal anisotropic mesh generation," *Procedia Engineering*, vol. 82, pp. 403–415, 2014.

[43] A. Loseille, D. L. Marcum, and F. Alauzet, "Alignment and orthogonality in anisotropic metric-based mesh adaptation," in *53rd AIAA Aerospace Sciences Meeting*, 2015, p. 0915.

[44] R. Anderson, J. Andrej, A. Barker, J. Bramwell, J.-S. Camier, J. Cerveny, V. Dobrev, Y. Dudouit, A. Fisher, T. Kolev *et al.*, "Mfem: A modular finite element methods library," *Computers & Mathematics with Applications*, vol. 81, pp. 42–74, 2021.

[45] C. Burstedde, L. C. Wilcox, and O. Ghattas, "p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees," *SIAM Journal on Scientific Computing*, vol. 33, no. 3, pp. 1103–1133, 2011.

[46] J. M. Lee, *Introduction to Smooth manifolds*. Springer, 2012.

[47] L. Botti, "Influence of reference-to-physical frame mappings on approximation properties of discontinuous piecewise polynomial spaces," *Journal of Scientific Computing*, vol. 52, pp. 675–703, 2012.

[48] D. Moxey, S. P. Sastry, and R. M. Kirby, "Interpolation error bounds for curvilinear finite elements and their implications on adaptive mesh refinement," *Journal of Scientific Computing*, vol. 78, no. 2, pp. 1045–1062, 2019.

[49] P. F. M. Tesini, "An h-multigrid approach for high-order discountinuous galerkin methods," 2008.

[50] F. Bassi, L. Botti, A. Colombo, D. A. Di Pietro, and P. Tesini, "On the flexibility of agglomeration based physical space discontinuous galerkin discretizations," *Journal of Computational Physics*, vol. 231, no. 1, pp. 45–65, 2012.

[51] P. Solin, K. Segeth, and I. Dolezel, *Higher-order finite element methods*. Chapman and Hall/CRC, 2003.

[52] P. G. Ciarlet and P.-A. Raviart, "Interpolation theory over curved elements, with applications to finite element methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 1, no. 2, pp. 217–249, 1972.

[53] L. Freitag, P. Knupp, T. Munson, and S. Shontz, "A comparison of optimization software for mesh shape-quality improvement problems." Argonne National Lab., IL (US), Tech. Rep., 2002.

[54] S. Agarwal and K. Mierle, "Ceres solver: Tutorial & reference," *Google Inc*, vol. 2, no. 72, p. 8, 2012.

[55] F. Alauzet, A. Loseille, A. Dervieux, and P. Frey, "Multi-dimensional continuous metric for mesh adaptation," in *Proceedings of the 15th international meshing roundtable*. Springer, 2006, pp. 191–214.

[56] J. M. Lee, *Introduction to Riemannian manifolds*. Springer, 2018, vol. 2.

[57] S. R. (https://math.stackexchange.com/users/10549/selene routley), "What is a manifold?" Mathematics Stack Exchange, uRL:https://math.stackexchange.com/q/1211764 (version: 2019-10-12). [Online]. Available: https://math.stackexchange.com/q/1211764

[58] J. Foster, J. D. Nightingale, and J. Foster, *A short course in General Relativity*. Springer, 1995.

[59] P. C. Caplan, R. Haimes, D. L. Darmofal, and M. C. Galbraith, "Anisotropic geometry-conforming d-simplicial meshing via isometric embeddings," *Procedia engineering*, vol. 203, pp. 141–153, 2017.

[60] C. Epstein and M. Gage, "The curve shortening flow," in *Wave Motion: Theory, Modelling, and Computation: Proceedings of a Conference in Honor of the 60th Birthday of Peter D. Lax*. Springer, 1987, pp. 15–59.

[61] P. J. Frey and P.-L. George, "Mesh generation," *(No Title)*, 2008.

[62] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-euclidean metrics for fast and simple calculus on diffusion tensors," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 56, no. 2, pp. 411–421, 2006.

[63] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The r*-tree: an efficient and robust access method for points and rectangles," ser. Acm Sigmod Record, vol. 19, no. 2. Acm, 1990, pp. 322–331.

[64] P. J. Frey and P.-L. George, *Maillages: applications aux éléments finis*. Hermès Science Publications, 1999.

[65] F. Alauzet, "Size gradation control of anisotropic meshes," *Finite Elements in Analysis and Design*, vol. 46, no. 1-2, pp. 181–202, 2010.

[66] F. Hecht, "Bamg: bidimensional anisotropic mesh generator," *User Guide. INRIA, Rocquencourt*, vol. 17, 1998.

[67] C. Dobrzynski, *MMG3D: User guide*, Inria, 2012, available at http://www.mmgtools.org/.

[68] C. Dobrzynski and P. Frey, "Anisotropic delaunay mesh adaptation for unsteady simulations," in *Proceedings of the 17th international Meshing Roundtable*. Springer, 2008, pp. 177–194.

[69] P. Sebastiani, "On the derivatives of matrix powers," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 3, pp. 640–648, 1996.

[70] W. Huang and R. D. Russell, *Adaptive moving mesh methods*. Springer Science & Business Media, 2010, vol. 174.

[71] E. F. D'Azevedo and R. B. Simpson, "On optimal interpolation triangle incidences," *SIAM Journal on scientific and statistical computing*, vol. 10, no. 6, pp. 1063–1075, 1989.

[72] M. Berzins, "A solution-based triangular and tetrahedral mesh quality indicator," *SIAM Journal on Scientific Computing*, vol. 19, no. 6, pp. 2051–2060, 1998.

[73] T. Apel, *Anisotropic finite elements: local estimates and applications*. Teubner Stuttgart, 1999.

[74] L. Formaggia and S. Perotto, "New anisotropic a priori error estimates," *Numerische Mathematik*, vol. 89, no. 4, pp. 641–667, 2001.

[75] L. Chen, P. Sun, and J. Xu, "Optimal anisotropic meshes for minimizing interpolation errors in l^p-norm," *Mathematics of Computation*, vol. 76, no. 257, pp. 179–204, 2007.

[76] L. Chen, "On minimizing the linear interpolation error of convex quadratic functions and the optimal simplex," *East J. Approx*, vol. 14, no. 3, pp. 271–284, 2008.

[77] W. Cao, "An interpolation error estimate on anisotropic meshes in \mathcalr^n and optimal metrics for mesh refinement," *SIAM journal on numerical analysis*, vol. 45, no. 6, pp. 2368–2391, 2007.

[78] ——, "Anisotropic measures of third order derivatives and the quadratic interpolation error on triangular elements," *SIAM Journal on Scientific Computing*, vol. 29, no. 2, pp. 756–781, 2007.

[79] ——, "An interpolation error estimate in $\mathbb{R}^2$ based on the anisotropic measures of higher order derivatives," *Mathematics of computation*, vol. 77, no. 261, pp. 265–286, 2008.

[80] J.-M. Mirebeau, "Optimally adapted meshes for finite elements of arbitrary order and w 1, p norms," *Numerische Mathematik*, vol. 120, no. 2, pp. 271–305, 2012.

[81] F. Hecht and R. Kuate, "An approximation of anisotropic metrics from higher order interpolation error for triangular mesh adaptation," *Journal of computational and applied mathematics*, vol. 258, pp. 99–115, 2014.

[82] V. Dolejší, "Anisotropic hp-adaptive method based on interpolation error estimates in the lq-norm," *Applied Numerical Mathematics*, vol. 82, pp. 80–114, 2014.

[83] O. C. Zienkiewicz and J. Z. Zhu, "A simple error estimator and adaptive procedure for practical engineerng analysis," *International journal for numerical methods in engineering*, vol. 24, no. 2, pp. 337–357, 1987.

[84] ——, "The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique," *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, pp. 1331–1364, 1992.

[85] Z. Zhang and A. Naga, "A new finite element gradient recovery method: superconvergence property," *SIAM Journal on Scientific Computing*, vol. 26, no. 4, pp. 1192–1213, 2005.

[86] D. A. Venditti and D. L. Darmofal, "Grid adaptation for functional outputs: application to two-dimensional inviscid flows," *Journal of Computational Physics*, vol. 176, no. 1, pp. 40–69, 2002.

[87] ——, "Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows," *Journal of Computational Physics*, vol. 187, no. 1, pp. 22–46, 2003.

[88] F. Alauzet and L. Frazza, "Feature-based and goal-oriented anisotropic mesh adaptation for rans applications in aeronautics and aerospace," *Journal of Computational Physics*, vol. 439, p. 110340, 2021.

[89] A. Ern and J.-L. Guermond, *Theory and practice of finite elements*. Springer, 2004, vol. 159.

[90] R. Kuate, "Adaptation de maillage anisotrope: étude, construction d'estimateurs d'erreur et raffinement hexaédrique." Ph.D. dissertation, Université Pierre et Marie Curie-Paris VI, 2008.

[91] G. Gamrath, D. Anderson, K. Bestuzheva, W.-K. Chen, L. Eifler, M. Gasse, P. Gemander, A. Gleixner, L. Gottwald, K. Halbig, G. Hendel, C. Hojny, T. Koch, P. L. Bodic, S. J. Maher, F. Matter, M. Miltenberger, E. Mühmer, B. Müller, M. Pfetsch, F. Schlösser, F. Serrano, Y. Shinano, C. Tawfik, S. Vigerske, F. Wegscheider, D. Weninger, and J. Witzig, "The scip optimization suite 7.0," ZIB, Takustr. 7, 14195 Berlin, Tech. Rep. 20-10, 2020.

[92] D. F. Kurtulus, "On the unsteady behavior of the flow around naca 0012 airfoil with steady external conditions at re= 1000," *International Journal of Micro Air Vehicles*, vol. 7, no. 3, pp. 301–326, 2015.

[93] T. Kouser, Y. Xiong, D. Yang, and S. Peng, "Direct numerical simulations on the three-dimensional wake transition of flows over naca0012 airfoil at re= 1000," *International Journal of Micro Air Vehicles*, vol. 13, 2021.

[94] D. R. Q. Pacheco, *Stable and stabilised finite element methods for incompressible flows of generalised Newtonian fluids*. Verlag der Technischen Universität Graz, 2021.

[95] N. A. Pierce and M. B. Giles, "Adjoint recovery of superconvergent functionals from pde approximations," *SIAM review*, vol. 42, no. 2, pp. 247–264, 2000.

[96] K. J. Fidkowski and D. L. Darmofal, "Review of output-based error estimation and mesh adaptation in computational fluid dynamics," *AIAA journal*, vol. 49, no. 4, pp. 673–694, 2011.

[97] P.-A. Beaufort, "Parameterizations for mesh generation: discrete atlases, crossfields and 3d frames," Ph.D. dissertation, UCL-Université Catholique de Louvain, 2019.

[98] T. Shifrin, "Differential geometry: a first course in curves and surfaces," *University of Georgia*, p. 24, 2015.

[99] D. Drucker and S. A. Williams, "When does water find the shortest path downhill? the geometry of steepest descent curves," *The American mathematical monthly*, vol. 110, no. 10, pp. 869–885, 2003.

[100] D. N. (https://math.stackexchange.com/users/390907/dimitrios nt), "Prove the curvature of a level set equals divergence of the normalized gradient," Mathematics Stack Exchange, uRL:https://math.stackexchange.com/q/3856789 (version: 2020-10-08). [Online]. Available: https://math.stackexchange.com/q/3856789

[101] D. Marcum and F. Alauzet, "Aligned metric-based anisotropic solution adaptive mesh generation," *Procedia Engineering*, vol. 82, pp. 428–444, 2014.

[102] ——, "3d metric-aligned and orthogonal solution adaptive mesh generation," *Procedia engineering*, vol. 203, pp. 78–90, 2017.

[103] T. C. Baudouin, J.-F. Remacle, E. Marchandise, F. Henrotte, and C. Geuzaine, "A frontal approach to hex-dominant mesh generation," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 1, pp. 1–30, 2014.

[104] Z. Xiao, C. Ollivier-Gooch, and J. D. Z. Vazquez, "Anisotropic tetrahedral mesh adaptation with improved metric alignment and orthogonality," *Computer-Aided Design*, vol. 143, p. 103136, 2022.

[105] J. Liu and S. Sun, "Small polyhedron reconnection: a new way to eliminate poorly-shaped tetrahedra," in *Proceedings of the 15th international meshing roundtable*. Springer, 2006, pp. 241–257.

[106] C. Marot, K. Verhetsel, and J.-F. Remacle, "Reviving the search for optimal tetrahedralizations," in *Proceedings of the 28th International Meshing Roundtable, IMR 2019, Buffalo, New York, USA, October 14-17, 2019*, J. Peirò and R. Viertel, Eds., 2019, pp. 326–336.

[107] A. Loseille and R. Lohner, "Cavity-based operators for mesh adaptation," in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013, p. 152.

[108] P. Knupp, "Introducing the target-matrix paradigm for mesh optimization via node-movement," *Engineering with Computers*, vol. 28, no. 4, pp. 419–429, 2012.

[109] V. Dobrev, P. Knupp, T. Kolev, K. Mittal, and V. Tomov, "The target-matrix optimization paradigm for high-order meshes," *SIAM Journal on Scientific Computing*, vol. 41, no. 1, pp. B50–B68, 2019.

[110] G. Haller, "Finding finite-time invariant manifolds in two-dimensional velocity fields," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 10, no. 1, pp. 99–108, 2000.

[111] ——, "Distinguished material surfaces and coherent structures in three-dimensional fluid flows," *Physica D: Nonlinear Phenomena*, vol. 149, no. 4, pp. 248–277, 2001.

[112] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities," *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[113] A. Cunha, S. Canann, and S. Saigal, "Automatic boundary sizing for 2d and 3d meshes," *ASME Applied Mechanics Division-Publications-AMD*, vol. 220, pp. 65–72, 1997.

[114] S. J. Owen and S. Saigal, "Neighborhood-based element sizing control for finite element surface meshing," ser. 6th International Meshing Roundtable Proceedings, 1997, pp. 143–154.

[115] J. Chen, Z. Xiao, Y. Zheng, J. Zheng, C. Li, and K. Liang, "Automatic sizing functions for unstructured surface mesh generation," *International Journal for Numerical Methods in Engineering*, vol. 109, no. 4, pp. 577–608, 2017.

[116] J. Chen, Z. Liu, Y. Zheng, P. Zheng, J. Zheng, Z. Xiao, and C. Yu, "Automatic sizing functions for 3d unstructured mesh generation," *Procedia engineering*, vol. 203, pp. 245–257, 2017.

[117] S. Pirzadeh, "Structured background grids for generation of unstructured grids by advancing-front method," *AIAA journal*, vol. 31, no. 2, pp. 257–265, 1993.

[118] J. Zhu, T. D. Blacker, and R. Smith, "Background overlay grid size functions," in *Proceedings of the 11th International Meshing Roundtable, IMR 2002, Ithaca, New York, USA, September 15-18, 2002*, N. Chrisochoides, Ed., 2002, pp. 65–73. [Online]. Available: http://imr.sandia.gov/papers/abstracts/Zh242.html

[119] K.-F. Tchon, M. Khachan, F. Guibault, and R. Camarero, "Three-dimensional anisotropic geometric metrics based on local domain curvature and thickness," *Computer-Aided Design*, vol. 37, no. 2, pp. 173–187, 2005.

[120] W. R. Quadros, V. Vyas, M. Brewer, S. J. Owen, and K. Shimada, "A computational framework for automating generation of sizing function in assembly meshing via disconnected skeletons," *Engineering with Computers*, vol. 26, no. 3, pp. 231–247, 2010.

[121] P.-O. Persson, "Mesh size functions for implicit geometries and pde-based gradient limiting," *Engineering with Computers*, vol. 22, no. 2, pp. 95–109, 2006.

[122] J. Chen, B. Cao, Y. Zheng, L. Xie, C. Li, and Z. Xiao, "Automatic surface repairing, defeaturing and meshing algorithms based on an extended b-rep," *Advances in Engineering Software*, vol. 86, pp. 55–69, 2015.

[123] J. Shan, Y. Li, Y. Guo, and Z. Guan, "A robust backward search method based on walk-through for point location on a 3d surface mesh," *International journal for numerical methods in engineering*, vol. 73, no. 8, pp. 1061–1076, 2008.

[124] Y. Zhang, W. Wang, X. Liang, Y. Bazilevs, M.-C. Hsu, T. Kvamsdal, R. Brekken, and J. Isaksen, "High-fidelity tetrahedral mesh generation from medical imaging data for fluid-structure interaction analysis of cerebral aneurysms," *Computer Modeling in Engineering and Sciences (CMES)*, vol. 42, no. 2, p. 131, 2009.

[125] X. Liang and Y. Zhang, "An octree-based dual contouring method for triangular and tetrahedral mesh generation with guaranteed angle range," *Engineering with Computers*, vol. 30, no. 2, pp. 211–222, 2014.

[126] C. Burstedde, L. C. Wilcox, and O. Ghattas, "`p4est`: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees," *SIAM Journal on Scientific Computing*, vol. 33, no. 3, pp. 1103–1133, 2011.

[127] Y. Zhang, C. Bajaj, and B.-S. Sohn, "3d finite element meshing from imaging data," *Computer methods in applied mechanics and engineering*, vol. 194, no. 48-49, pp. 5083–5106, 2005.

[128] M. Turner, D. Moxey, and J. Peiró, "Automatic mesh sizing specification of complex three dimensional domains using an octree structure," in *Research note, 24th International Meshing Roundtable, Austin, Texas, USA, October 11-14, 2015*, F. Ledous and K. Lewis, Eds.

[129] F. Deister, U. Tremel, O. Hassan, and N. P. Weatherill, "Fully automatic and fast mesh size specification for unstructured mesh generation," *Engineering with computers*, vol. 20, no. 3, pp. 237–248, 2004.

[130] S. Rusinkiewicz, "Estimating curvatures and their derivatives on triangle meshes," ser. Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. IEEE, 2004, pp. 486–493.

[131] T. K. Dey and W. Zhao, "Approximating the medial axis from the voronoi diagram with a convergence guarantee," *Algorithmica*, vol. 38, no. 1, pp. 179–200, 2004.

[132] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and D. Panozzo, "ABC: A big CAD model dataset for geometric deep learning," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 9601–9611. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Koch_ABC_A_Big_CAD_Model_Dataset_for_Geometric_Deep_Learning_CVPR_2019_paper.html

[133] M.-G. Vallet, C.-M. Manole, J. Dompierre, S. Dufour, and F. Guibault, "Numerical comparison of some hessian recovery techniques," *International Journal for Numerical Methods in Engineering*, vol. 72, no. 8, pp. 987–1007, 2007.

[134] M. Picasso, F. Alauzet, H. Borouchaki, and P.-L. George, "A numerical study of some hessian recovery techniques on isotropic and anisotropic meshes," *SIAM Journal on Scientific Computing*, vol. 33, no. 3, pp. 1058–1076, 2011.

[135] H. Guo, Z. Zhang, and R. Zhao, "Hessian recovery for finite element methods," *Mathematics of Computation*, vol. 86, no. 306, pp. 1671–1692, 2017.

[136] A. Naga and Z. Zhang, "The polynomial-preserving recovery for higher order finite element methods in 2d and 3d," *Discrete and continuous dynamical systems series B*, vol. 5, no. 3, p. 769, 2005.

[137] ——, "A posteriori error estimates based on the polynomial preserving recovery," *SIAM Journal on Numerical Analysis*, vol. 42, no. 4, pp. 1780–1800, 2004.

[138] T. Briffard, "Contributions à l'adaptation de maillage anisotrope sur base hiérarchique," Ph.D. dissertation, Université Laval Quebec, Canada, 2017.

[139] H. Guo, Z. Zhang, R. Zhao, and Q. Zou, "Polynomial preserving recovery on boundary," *Journal of Computational and Applied Mathematics*, vol. 307, pp. 119–133, 2016.

[140] C. Huang and Z. Zhang, "Polynomial preserving recovery for quadratic elements on anisotropic meshes," *Numerical Methods for Partial Differential Equations*, vol. 28, no. 3, pp. 966–983, 2012.

[141] J. Xu and Z. Zhang, "Analysis of recovery type a posteriori error estimators for mildly structured grids," *Mathematics of Computation*, vol. 73, no. 247, pp. 1139–1152, 2004.

[142] Z. Zhang, "Polynomial preserving recovery for meshes from delaunay triangulation or with high aspect ratio," *Numerical Methods for Partial Differential Equations: An International Journal*, vol. 24, no. 3, pp. 960–971, 2008.

APPENDIX A PROOFS AND DEVELOPMENTS

A.1 Appendix of Chapter 2

A.1.1 Gradients of the L^2 norm of the interpolation error

Let E be the quadratic interpolation error defined on $\mathcal{P}^2$ triangles by:

$$E = \|e_f\|_{L^2(\Omega)} = \sqrt{\int_\Omega e_f^2 \, dx} \simeq \sqrt{\sum_{K\in\mathcal{T}} \sum_{i=1}^{n_q} e_f^2(\boldsymbol{\xi}_i)\, \mathcal{J}_K(\boldsymbol{\xi}_i)\, w_i}. \tag{1}$$

The dependence of E in the Lagrange control vertices $\boldsymbol{x}_j$ is explicited through:

$$\begin{aligned} e_f(\boldsymbol{\xi}_i;\boldsymbol{x}_j) &= f\left(F_K(\boldsymbol{\xi}_i)\right) - \Pi_K^2 f(\boldsymbol{\xi}_i) = f\left(\sum_j \phi_j^{\text{geo}}(\boldsymbol{\xi}_i)\boldsymbol{x}_j\right) - \sum_j \phi_j(\boldsymbol{\xi}_i) f(\boldsymbol{x}_j), \\ \mathcal{J}_K(\boldsymbol{\xi}_i;\boldsymbol{x}_j) &= x_\xi(\boldsymbol{\xi}_i) y_\eta(\boldsymbol{\xi}_i) - y_\xi(\boldsymbol{\xi}_i) x_\eta(\boldsymbol{\xi}_i) \quad \text{with e.g.} \quad x_\xi(\boldsymbol{\xi}) = \sum_j x_j \frac{\partial \phi_j^{\text{geo}}}{\partial \xi}(\boldsymbol{\xi}). \end{aligned} \tag{2}$$

A vertex $\boldsymbol{x}_j$ is shared by a shell $\omega(\boldsymbol{x}_j)$ of triangles (two for edge vertices) and only affects the elementwise error on this shell, so that the gradient of E with respect to the Lagrange control vertices writes:

$$\nabla_{\boldsymbol{x}_j} E = \frac{1}{2E} \nabla_{\boldsymbol{x}_j} \int_{\omega(\boldsymbol{x}_j)} e_f^2 \, dx \simeq \frac{1}{2E} \sum_{K\in\omega(\boldsymbol{x}_j)} \sum_{i=1}^{n_q} \nabla_{\boldsymbol{x}_j} \left(e_f^2(\boldsymbol{\xi}_i) \mathcal{J}_K(\boldsymbol{\xi}_i) \right) w_i. \tag{3}$$

The gradient of the term in parentheses is $(\nabla_{\boldsymbol{x}_j} e_f^2)\mathcal{J}_K + e_f^2(\nabla_{\boldsymbol{x}_j}\mathcal{J}_K)$, with the involved gradients writing:

$$\begin{aligned} \nabla_{\boldsymbol{x}_j} e_f^2(\boldsymbol{\xi}_i) &= 2 e_f \left(\nabla f(\boldsymbol{x}(\boldsymbol{\xi}_i)) \phi_j^{\text{geo}}(\boldsymbol{\xi}_i) - f(\boldsymbol{x}_j)\phi_j(\boldsymbol{\xi}_i) \right), \\ \nabla_{\boldsymbol{x}_j} \mathcal{J}_K(\boldsymbol{\xi}_i) &= \left(\phi_{j,\xi}^{\text{geo}} y_\eta(\boldsymbol{\xi}_i) - y_\xi(\boldsymbol{\xi}_i)\phi_{j,\eta}^{\text{geo}}, \quad x_\xi(\boldsymbol{\xi}_i)\phi_{j,\eta}^{\text{geo}} - \phi_{j,\xi}^{\text{geo}} x_\eta(\boldsymbol{\xi}_i) \right)^T. \end{aligned} \tag{4}$$

A.2 Appendix of Chapter 3

A.2.1 Differential of smooth maps and smooth functions

We show that both concepts of differentials, inroduced in Sections 3.1.1 and 3.1.2, refer to the same quantity for real-valued maps $f : M \to \mathbb{R}$. In coordinates, (3.23) writes:

$$df_{\boldsymbol{p}}(v) \overset{(3.24)}{=} \frac{\partial f}{\partial x^i}(\boldsymbol{p})\, dx^i|_{\boldsymbol{p}}(v) = v^i \frac{\partial f}{\partial x^i}(\boldsymbol{p}) \in \mathbb{R}. \tag{5}$$

In Section 3.1.1, the differential was introduced as a linear map $df_{\boldsymbol{p}} : T_{\boldsymbol{p}}M \to T_{f(\boldsymbol{p})}\mathbb{R}$, whereas in Section 3.1.2 it is introduced as a covector, hence a linear functional $df_{\boldsymbol{p}} : T_{\boldsymbol{p}}M \to \mathbb{R}$. The key ideas are that (i) the images $T_{f(\boldsymbol{p})}\mathbb{R}$ and $\mathbb{R}$ can be identified with each other, i.e., the tangent space to $\mathbb{R}$ at $f(\boldsymbol{p})$ is $\mathbb{R}$ itself, shifted by $f(\boldsymbol{p})$; and (ii) both differentials have the same components in coordinates, namely, the partial derivatives of f:

$$df_{\boldsymbol{p}}(v) \overset{(3.7)}{=} df_{\boldsymbol{p}}\left(v^i \frac{\partial}{\partial x^i}\bigg|_{\boldsymbol{p}}\right) = v^i \frac{\partial f}{\partial x^i}(\boldsymbol{p}) \frac{\partial}{\partial y}\bigg|_{f(\boldsymbol{p})} \in T_{f(\boldsymbol{p})}\mathbb{R}, \tag{6}$$

where y denotes the unique coordinate on $T_{f(\boldsymbol{p})}\mathbb{R}$, and

$$df_{\boldsymbol{p}}(v) \overset{(5)}{=} \frac{\partial f}{\partial x^i}(\boldsymbol{p})\, dx^i|_{\boldsymbol{p}}(v) = v^i \frac{\partial f}{\partial x^i}(\boldsymbol{p}) \in \mathbb{R}. \tag{7}$$

In the former, the partial derivatives appear from the chain rule of the composition of the argument function with f, whereas in the latter the partial derivatives are the very components of the covector.

A.2.2 Introduction to curvature and flatness criterion

We present the introductory example from Lee [56], for which the following result is needed:

Proposition A.1. *Let γ be a curve on M and $v \in T_{\gamma(t_0)}M$ for some t_0. There is a unique vector field V that is parallel along γ, i.e., for which $\nabla_{\gamma'(t)}V = 0$, and such that $V(t_0) = v$.*

V is called the **parallel extension of** v **along** γ. In particular, if v is the zero vector, then V is the zero vector field. This proposition states that V exists and is unique along a curve, but it might not exist in an open subset around $\gamma(t_0)$ where v is defined. One interpretation of curvature is precisely the obstruction for V to exist on open subsets.

In the more abstract setting, curvature is introduced as a measure of how a manifold fails

to be locally isometric to a Euclidean space. This is based on this observation: on $\mathbb{R}^2$, any tangent vector can be extended to a unique parallel vector field, that is simply a constant vector field. Since isometries preserve the Levi-Civita connection and the associated parallel transport, manifolds isometric to $\mathbb{R}^2$ should have the same property locally. To quantify this, we construct a parallel extension of a vector on an arbitrary 2-manifold M following a path, and examine the condition under which this vector fields remains parallel under a different path. Let (x, y) denote local coordinates around $\boldsymbol{p} \in M$ and inducing the coordinate frame (∂_x, ∂_y) in a neighborhood of $\boldsymbol{p}$ (recall that this frame is a vector *field*), and let $z \in T_{\boldsymbol{p}}M$. We call *x-lines* the (curved) coordinate lines parallel to the x-axis, and similarly for y-lines. A vector field Z is defined by parallel transporting z "horizontally" along the x-axis, then "vertically" along each y-line. By construction, $\nabla_{\partial_y} Z = 0$ everywhere since Z was parallel transported along the y-lines, while $\nabla_{\partial_x} Z = 0$ holds only the x-axis. To check Z is fully parallel, we wish to know whether the converse is true, that is, if $\nabla_{\partial_x} Z = 0$ also holds everywhere. By Proposition (A.1), if $\nabla_{\partial_y} \nabla_{\partial_x} Z = 0$ holds, then so does $\nabla_{\partial_x} Z = 0$, because since the vector $\nabla_{\partial_x} Z$ is zero on the x-axis, its unique extension along each y-line is the zero vector field. Since we know that $\nabla_{\partial_y} Z = 0$ everywhere by construction, verifying the symmetry relation $\nabla_{\partial_x} \nabla_{\partial_y} Z = \nabla_{\partial_y} \nabla_{\partial_x} Z$ implies the sought relation $\nabla_{\partial_y} \nabla_{\partial_x} Z = 0$, hence it sums up to verifying whether the mixed derivatives w.r.t. the coordinate frame commute, which is a property of the connection ∇. For the Euclidean connection $\overline{\nabla}$, the covariant derivative w.r.t. the coordinate vector fields $\boldsymbol{e}_i = \partial_i$ reducejacobian to directional derivatives, which commute:

$$\overline{\nabla}_{\partial_y} \overline{\nabla}_{\partial_x} Z = \overline{\nabla}_{\partial_y} \left((\partial_x Z^i) \boldsymbol{e}_i \right) = (\partial_y \partial_x Z^i) \boldsymbol{e}_i = (\partial_x \partial_y Z^i) \boldsymbol{e}_i = \overline{\nabla}_{\partial_x} \overline{\nabla}_{\partial_y} Z. \tag{8}$$

For general vector fields X, Y however, directional derivatives do not commute, even in $\mathbb{R}^2$:

$$\overline{\nabla}_X \overline{\nabla}_Y Z = \overline{\nabla}_X \left(Y^i \frac{\partial Z^j}{\partial x^i} \boldsymbol{e}_j \right) = X^k \frac{\partial}{\partial x^k} \left(Y^i \frac{\partial Z^j}{\partial x^i} \right) \boldsymbol{e}_j = XY(Z^j)\, \boldsymbol{e}_j, \tag{9}$$

and the commutator is:

$$\overline{\nabla}_X \overline{\nabla}_Y Z - \overline{\nabla}_Y \overline{\nabla}_X Z = XY(Z^j)\, \boldsymbol{e}_j - YX(Z^j)\, \boldsymbol{e}_j = \overline{\nabla}_{[X,Y]} Z, \tag{10}$$

with $[X, Y]$ the Lie bracket of X and Y, see Section 3.1.1. The relation (10) holds on $\mathbb{R}^n$ for all vector fields X, Y, Z, and translates the fact that a tangent vector can be uniquely extended to a parallel vector field on $\mathbb{R}^n$. More generally, we say that the Levi-Civita connection on a

smooth manifold M satisfies the **flatness criterion** if for all vector fields X, Y, Z on M:

$$\nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z = \nabla_{[X,Y]} Z. \tag{11}$$

A.2.3 Gradient of the interpolated metric

Let $\partial_j \mathcal{M}$ denote the gradient of $\mathcal{M}$ with respect to the front mesh coordinates, let $L \triangleq \sum_i \phi_i \ln \mathcal{M}_i$ and let $L = P\Lambda P^T$ be its diagonalization. From the definition of the matrix exponential, we write:

$$\begin{aligned} \partial_j \mathcal{M} = \partial_j \exp L &= \partial_j \left(P \, \exp \Lambda \, P^T \right) \\ &= (\partial_j P) \exp \Lambda P^T + P(\partial_j \exp \Lambda) P^T + P \exp \Lambda (\partial_j P^T) \end{aligned} \tag{12}$$

The derivatives of the diagonal matrix of the eigenvalues is simply $\partial_j \exp \Lambda = \exp \Lambda \, \partial_j \Lambda$. We are left with the computation of the derivatives of the eigenvalues $\partial_j \Lambda$ and eigenvectors $\partial_j P = (\partial_j \boldsymbol{v}_1 \; \partial_j \boldsymbol{v}_2)$ of L. Defining $L_k \triangleq L - \lambda_k I$, the eigenvector $\boldsymbol{v}_k$ satisfies $L_k \boldsymbol{v}_k = \mathbf{0}$. Taking the derivative, we obtain:

$$\partial_j (L_k \boldsymbol{v}_k) = (\partial_j L_k) \boldsymbol{v}_k + L_k (\partial_j \boldsymbol{v}_k) = \mathbf{0}. \tag{13}$$

The derivatives of the eigenvalues are obtained by left-multiplying the expression above by $\boldsymbol{v}^T$. The second term writes:

$$\boldsymbol{v}_k^T L_k (\partial_j \boldsymbol{v}_k) = \left[L_k^T \boldsymbol{v}_k \right]^T \partial_j \boldsymbol{v}_k = \left[L_k \boldsymbol{v}_k \right]^T \partial_j \boldsymbol{v}_k = \mathbf{0} \tag{14}$$

since L, and thus L_k, is symmetric. Since $\boldsymbol{v}_k^T \cdot \boldsymbol{v}_k = 1$, the first term writes:

$$\boldsymbol{v}_k^T (\partial_j L_k) \boldsymbol{v}_k = \boldsymbol{v}_k^T \partial_j (L - \lambda_k I) \boldsymbol{v}_k = \boldsymbol{v}_k^T (\partial_j L) \boldsymbol{v}_k - \partial_j \lambda_k = 0. \tag{15}$$

This is solved to determine the $\partial_j \lambda_k$, which in turn determine the $\partial_j L_k$. Then, the eigenvectors derivatives are computed from (14) by solving:

$$\partial_j \boldsymbol{v}_k = L_k^+ \left(-\partial_j L_k \boldsymbol{v}_k \right), \tag{16}$$

where L_k^+ is the Moore-Penrose pseudo-inverse of L_k, as the matrix is singular by definition. It is computed using Eigen's `CompleteOrthogonalDecomposition (COD)` class. For stability, (16) is solved calling Eigen's `solve()` function rather than `pseudoinverse()`, for example:

```
Eigen::CompleteOrthogonalDecomposition<Eigen::Matrix2d> Lkdec =
```

```
   Lk.completeOrthogonalDecomposition();
Eigen::Vector2d grad_vk = Lkdec.solve(-gradLk * vk);
```

Finally, the derivatives $\partial_j L$, required to compute the derivatives of the eigenvalues in (15), are given by the chain rule:

$$\begin{aligned}\nabla L = \nabla\left(\sum_i \phi_i(\hat{\boldsymbol{\xi}}) \ln \mathcal{M}(\hat{\boldsymbol{X}}_i)\right) &= \sum_i \nabla\phi_i(\hat{\boldsymbol{\xi}}) \ln \mathcal{M}(\hat{\boldsymbol{X}}_i) \\ &= \sum_i \left[\nabla_{\hat{\boldsymbol{\xi}}}\phi_i \nabla\hat{\boldsymbol{\xi}}\right] \ln \mathcal{M}(\hat{\boldsymbol{X}}_i) \\ &= \sum_i \left[\nabla_{\hat{\boldsymbol{\xi}}}\phi_i \left(\nabla_{\hat{\boldsymbol{\xi}}} F_K(\hat{\boldsymbol{\xi}})\right)^{-1}\right] \ln \mathcal{M}(\hat{\boldsymbol{X}}_i),\end{aligned} \tag{17}$$

where we used $\nabla\hat{\boldsymbol{\xi}}(\hat{\boldsymbol{x}}) = \left(\nabla_{\hat{\boldsymbol{\xi}}}\hat{\boldsymbol{x}}(\hat{\boldsymbol{\xi}})\right)^{-1} = \left(\nabla_{\hat{\boldsymbol{\xi}}}\boldsymbol{x}(\hat{\boldsymbol{\xi}})\right)^{-1} = \left(\nabla_{\hat{\boldsymbol{\xi}}} F_K(\hat{\boldsymbol{\xi}})\right)^{-1}$ since the physical coordinates of the front mesh and background mesh are the same.

A.3 Appendix of Chapter 4

A.3.1 Invariants for high-order error estimates

We introduce additional invariants involving third and fourth order symmetric tensors shared by unit linear triangles. These invariants may find some use when deriving high-order error estimates on unit linear elements involving the metric. Let C be a symmetric covariant 3-tensor on $K_\triangle$ with components:

$$C_{111} = a, \quad C_{112} = b, \quad C_{122} = c, \quad C_{222} = d. \tag{18}$$

For three tangent vectors $u, v, w \in T_{\boldsymbol{p}}M$, the tensor evaluation writes:

$$C(u, v, w) = C_{ijk} u^i v^j w^k = w^T \begin{pmatrix} u^T[C_1]v \\ u^T[C_2]v \end{pmatrix}, \tag{19}$$

with the submatrices $[C_m]_{ij} \triangleq C_{ijm}$. We examine the contractions of this tensor on the edges of the triangle $R(\theta)K_\triangle$, whose edges are explicitly given by:

$$E_{12} = R(\theta)\begin{pmatrix}1\\0\end{pmatrix}, \quad E_{23} = R(\theta)\begin{pmatrix}-1/2\\ \sqrt{3}/2\end{pmatrix}, \quad E_{31} = R(\theta)\begin{pmatrix}-1/2\\ -\sqrt{3}/2\end{pmatrix}. \tag{20}$$

The sum of the evaluations of C over all edges combinations does not depend on the edges, but since C is an odd-order tensor, it depends on the rotation of $K_\triangle$, as sines and cosines do

not cancel out. It writes:

$$\begin{aligned} \sum_{i=1}^{3} C(E_i, E_i, E_i) &= \frac{3}{4}\left(\cos 3\theta\,(a-3c) + \sin 3\theta\,(3b-d)\right), \\ &\downarrow \\ \left|\sum_{i=1}^{3} C(E_i, E_i, E_i)\right| &\leq \frac{3}{4}\left(|a-3c| + |3b-d|\right) \triangleq \frac{3}{4}\,\mathrm{Op}(C), \end{aligned} \tag{21}$$

where $\mathrm{Op}(C)$ is an operator applied to C that replaces the trace. On an arbitrary triangle $K = \mathcal{M}^{-1/2} R K_\triangle$ unit for the metric $\mathcal{M}$, this bound evaluates on $K_\triangle$ as:

$$\left|\sum_{i=1}^{3} C(e_i, e_i, e_i)\right| \leq \frac{3}{4}\,\mathrm{Op}(C_{mno}\mathcal{M}^{-1/2}_{mi}\mathcal{M}^{-1/2}_{nj}\mathcal{M}^{-1/2}_{ok}). \tag{22}$$

Note that positive-definiteness is not defined for third-order (or odd-order) tensors, but we can say that C is *copositive* if is satisfies $C(x,x,x) \geq 0$ for positive x.

Similarly, let Q be a symmetric covariant 4-tensor on $K_\triangle$ with components:

$$Q_{1111} = a, \quad Q_{1112} = b, \quad Q_{1122} = c, \quad Q_{1222} = d, \quad Q_{2222} = e. \tag{23}$$

For four tangent vectors $u, v.w, z \in T_{\boldsymbol{p}}M$, we have:

$$Q(u,v,w,z) = Q_{ijk\ell}u^i v^j w^k z^\ell = w^T \left(\begin{array}{c|c} u^T[Q_{11}]v & u^T[Q_{12}]v \\ \hline u^T[Q_{21}]v & u^T[Q_{22}]v \end{array}\right) z, \tag{24}$$

with $[Q_{mn}]_{ij} \triangleq Q_{ijmn}$. As an even-order tensor, the sum of its contractions over the edges is truly invariant:

$$\sum_{i=1}^{3} Q(E_i, E_i, E_i, E_i) = \frac{9}{8}\,(a + 2c + e) = \frac{9}{8}\,\mathrm{tr}\,Q, \tag{25}$$

where $\mathrm{tr}\,Q$ is the double trace w.r.t. the Euclidean metric:

$$\mathrm{tr}\,Q \triangleq \mathrm{tr}_{\overline{\mathcal{M}}}(\mathrm{tr}_{\overline{\mathcal{M}}}\,Q_{ijkl}) = \delta^{ij}(\delta^{kl}Q_{ijkl}) = Q_{iikk} = Q_{ikik} = Q_{ikki}. \tag{26}$$

On $K = \mathcal{M}^{-1/2} R K_\triangle$ unit for the metric $\mathcal{M}$, this bound evaluates on $K_\triangle$ as:

$$\left|\sum_{i=1}^{3} Q(e_i, e_i, e_i, e_i)\right| \leq \frac{9}{8}\,\mathrm{tr}(\underbrace{Q_{mnop}\mathcal{M}^{-1/2}_{mi}\mathcal{M}^{-1/2}_{nj}\mathcal{M}^{-1/2}_{ok}\mathcal{M}^{-1/2}_{p\ell}}_{\triangleq (F^*Q)_{ijk\ell}}). \tag{27}$$

The trace of the pullback tensor F^*Q can be computed e.g. from the flattened form of $[F^*Q]$,

stored in the following $\mathbb{R}^{4\times4}$ matrix, with $I = I^{2\times2}$:

$$[F^*Q]^{4\times4} = \left(\begin{array}{c|c} \mathcal{M}_{11}^{-\frac{1}{2}} I & \mathcal{M}_{12}^{-\frac{1}{2}} I \\ \hline \mathcal{M}_{12}^{-\frac{1}{2}} I & \mathcal{M}_{22}^{-\frac{1}{2}} I \end{array}\right) \left(\begin{array}{c|c} \mathcal{M}^{-\frac{1}{2}}[Q_{11}]\mathcal{M}^{-\frac{1}{2}} & \mathcal{M}^{-\frac{1}{2}}[Q_{12}]\mathcal{M}^{-\frac{1}{2}} \\ \hline \mathcal{M}^{-\frac{1}{2}}[Q_{21}]\mathcal{M}^{-\frac{1}{2}} & \mathcal{M}^{-\frac{1}{2}}[Q_{22}]\mathcal{M}^{-\frac{1}{2}} \end{array}\right) \left(\begin{array}{c|c} \mathcal{M}_{11}^{-\frac{1}{2}} I & \mathcal{M}_{12}^{-\frac{1}{2}} I \\ \hline \mathcal{M}_{12}^{-\frac{1}{2}} I & \mathcal{M}_{22}^{-\frac{1}{2}} I \end{array}\right), \tag{28}$$

and the map $[F^*Q]^{4\times4}_{2(k-1)+i,2(\ell-1)+j} = (F^*Q)_{ijk\ell}$.

A.3.2 Proof of Proposition 4.3

Proposition. *If a curved triangle K is unit w.r.t. $\mathcal{M}$ in the sense of Definition 4.2, then*

$$\ell_{\mathcal{M}}(e_i) = 1, \qquad \forall i = 1, 2, 3. \tag{29}$$

We start by parameterizing the edges of a high-order triangle. Let K be a unit triangle, e_i an edge of K with $i = 1, \dots 3$ and let $\boldsymbol{\gamma}_i(t) = (x_i(t), y_i(t)),\ t \in [0, 1]$ be the parameterization of e_i obtained by considering the restriction of the reference-to-physical transformation $F(\boldsymbol{\xi}(t))$ on the edges of K. On each edge, the composition $\boldsymbol{\xi}_i(t) = (\xi_i(t), \eta_i(t))$ is given by either $(\xi_1, \eta_1) = (t, 0)$ on the edge e_1, $(\xi_2, \eta_2) = (1 - t, t)$ on e_2 or $(\xi_3, \eta_3) = (0, 1 - t)$ on e_3. Thus:

$$\boldsymbol{\gamma}_i(t) = \left(\sum_{\boldsymbol{\alpha}\in\widehat{K}} X_{\boldsymbol{\alpha}}\phi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_i(t)), \sum_{\boldsymbol{\alpha}\in\widehat{K}} Y_{\boldsymbol{\alpha}}\phi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_i(t))\right) \tag{30}$$

and

$$\frac{d\boldsymbol{\gamma}_i}{dt} = \left(\frac{\partial x}{\partial \xi}\frac{d\xi_i}{dt} + \frac{\partial x}{\partial \eta}\frac{d\eta_i}{dt}, \frac{\partial y}{\partial \xi}\frac{d\xi_i}{dt} + \frac{\partial y}{\partial \eta}\frac{d\eta_i}{dt}\right) = [J_K]\Big|_{F(\boldsymbol{\xi}_i(t))}\frac{d\boldsymbol{\xi}_i}{dt}, \tag{31}$$

where J_K is the Jacobian matrix of K and $d\boldsymbol{\xi}_i/dt$ is constant. Now consider the equilateral triangle $K_\triangle$. Its edges $e_i^\triangle$ have unit length with respect to the Euclidean metric:

$$\ell_I(e_i^\triangle) = \int_0^1 \left\|\frac{d\gamma_i^\triangle}{dt}\right\|_I dt = \int_0^1 \sqrt{\left\langle \frac{d\gamma_i^\triangle}{dt}, \frac{d\gamma_i^\triangle}{dt}\right\rangle_I} dt = \int_0^1 \sqrt{\frac{d\boldsymbol{\xi}_i^T}{dt} J_\triangle^T J_\triangle \frac{d\boldsymbol{\xi}_i}{dt}}\, dt = 1. \tag{32}$$

Since K is unit with respect to the metric $\mathcal{M}(x)$, there exists a rotation $R \in \mathrm{SO}_2$ such that $J_K(x) = \mathcal{M}^{-1/2}(x) R J_\triangle$. We can thus write:

$$\left\|\frac{d\gamma_i}{dt}\right\|_{\mathcal{M}} = \sqrt{\left\langle \frac{d\gamma_i}{dt}, \frac{d\gamma_i}{dt}\right\rangle_{\mathcal{M}(\gamma_i(t))}} = \sqrt{\frac{d\boldsymbol{\xi}^T}{dt} J_K^T \mathcal{M} J_K \frac{d\boldsymbol{\xi}}{dt}}$$

$$= \sqrt{\frac{d\boldsymbol{\xi}^T}{dt} J_{\triangle}^T R^T \mathcal{M}^{-1/2} \mathcal{M} \mathcal{M}^{-1/2} R J_{\triangle} \frac{d\boldsymbol{\xi}}{dt}} = \sqrt{\frac{d\boldsymbol{\xi}^T}{dt} J_{\triangle}^T J_{\triangle} \frac{d\boldsymbol{\xi}}{dt}},$$

hence from (32):

$$\ell_{\mathcal{M}}(e_i) = \int_0^1 \left\| \frac{d\gamma_i}{dt} \right\|_{\mathcal{M}} dt = \int_0^1 \sqrt{\frac{d\boldsymbol{\xi}^T}{dt} J_{\triangle}^T J_{\triangle} \frac{d\boldsymbol{\xi}}{dt}}\, dt = 1, \tag{33}$$

thus the edges of K have unit length with respect to $\mathcal{M}$. □

A.4 Appendix of Chapter 5

A.4.1 Proof of Proposition 5.1.

We show that if $\mathcal{Q}$ satisfies the inequality:

$$\boldsymbol{z}^T \mathcal{Q} \boldsymbol{z} \geq 1, \quad \forall \boldsymbol{z} \in \Lambda(\pi_{\boldsymbol{p}}), \tag{34}$$

then it also satisfies:

$$|\pi_{\boldsymbol{p}}(\boldsymbol{\xi})| \leq \left(\boldsymbol{\xi}^T \mathcal{Q} \boldsymbol{\xi}\right)^{\frac{m}{2}}, \quad \forall \boldsymbol{\xi} \in \mathbb{R}^2. \tag{35}$$

The points in $\Lambda(\pi_{\boldsymbol{p}})$, the level curve 1 of $|\pi_{\boldsymbol{p}}|$, are described as follows. For any $\boldsymbol{\xi} \in \mathbb{R}^2$, the point $\lambda\boldsymbol{\xi}$ is in $\Lambda(\pi_{\boldsymbol{p}})$, with the real number λ defined by:

$$\lambda = \frac{1}{|\pi_{\boldsymbol{p}}(\boldsymbol{\xi})|^{\frac{1}{m}}}. \tag{36}$$

Indeed, since $\pi_{\boldsymbol{p}}$ is homogeneous of degree m, $|\pi_{\boldsymbol{p}}(\lambda\boldsymbol{\xi})| = |\lambda^m \pi_{\boldsymbol{p}}(\boldsymbol{\xi})| = |\pi_{\boldsymbol{p}}(\boldsymbol{\xi})|/|\pi_{\boldsymbol{p}}(\boldsymbol{\xi})| = 1$. Thus for any $\boldsymbol{\xi} \in \mathbb{R}^2$, $\mathcal{Q}$ satisfies $\boldsymbol{z}^T \mathcal{Q} \boldsymbol{z} = (\lambda\boldsymbol{\xi})^T \mathcal{Q} (\lambda\boldsymbol{\xi}) \geq 1$. It follows that $\mathcal{Q}$ satisfies:

$$\lambda^2 \boldsymbol{\xi}^T \mathcal{Q} \boldsymbol{\xi} = \frac{\boldsymbol{\xi}^T \mathcal{Q} \boldsymbol{\xi}}{|\pi_{\boldsymbol{p}}(\boldsymbol{\xi})|^{\frac{2}{m}}} \geq 1 \Leftrightarrow \left(\boldsymbol{\xi}^T \mathcal{Q} \boldsymbol{\xi}\right)^{\frac{m}{2}} \geq |\pi_{\boldsymbol{p}}(\boldsymbol{\xi})|. \tag{37}$$

The same reasoning can be applied starting from the other inequality, proving the equivalence.

A.4.2 Computation of the parameters μ_π, α_π and λ_α

Computation of μ_π

To determine μ_π, consider the circles tangent to $\pi(x, y)$. These circles have equation $x^2+y^2 = r^2$, and the tangency condition writes $y\partial_x\pi - x\partial_y\pi = \overline{\pi}(x, y) = 0$, with $\overline{\pi} = -bx^3 + (3a - 2c)x^2y + (2b - 3d)xy^2 + cy^3 \in \mathbb{H}^3$. Since $\overline{\pi}$ is homogeneous, if $(x^*, y^*) \in \mathbb{R}^2$ is such that

$\overline{\pi}(x^*, y^*) = 0$, then $\overline{\pi}(\lambda x^*, \lambda y^*) = \lambda^3 \overline{\pi}(x^*, y^*) = 0$ for any $\lambda \in \mathbb{R}$, thus the roots of $\overline{\pi}$ are along the line $(\lambda x^*, \lambda y^*)$. There are at most three such lines, and on each of these lines lies the tangency point of a circle of radius r with the level curve 1 of π, i.e., there exists $\lambda_r > 0$ such that:

$$\begin{cases} |\pi(\lambda_r x^*, \lambda_r y^*)| = \lambda_r^3 |\pi(x^*, y^*)| = 1, \\ \lambda_r^2 (x^*)^2 + \lambda_r^2 (y^*)^2 = r^2, \end{cases} \tag{38}$$

and $\mu_\pi^{-1/2}$ is the smallest of these radii. To avoid factoring the bivariate polynomial $\overline{\pi}(x, y)$, consider the root $(x^*, 1)$ with x^* real such that $\overline{\pi}(x^*, 1) = \overline{\pi}_1(x^*) = 0$, where $\overline{\pi}_1(x) = -bx^3 + (3a - 2c)x^2 + (2b - 3d)x + c$ is a univariate and nonhomogeneous polynomial of degree 3. If the n leading coefficients of $\overline{\pi}_1$ are 0, the first n roots are set to ε^{-1} for a small $\varepsilon > 0$, resulting in roots of $\overline{\pi}$ close to the x axis. For each real root, the tangency point is given by $(\lambda_r x^*, \lambda_r)$ satisfying:

$$\begin{cases} |\pi(\lambda_r x^*, \lambda_r)| = \lambda_r^3 |\pi(x^*, 1)| = 1, \\ \lambda_r^2 \left((x^*)^2 + 1 \right) = r^2, \end{cases} \tag{39}$$

yielding $\lambda_r = |\pi(x^*, 1)|^{-1/3}$ and finally:

$$\mu_\pi = r_{\min}^{-2} = \frac{1}{\max_{x^*} \lambda_r^2 \left((x^*)^2 + 1 \right)}. \tag{40}$$

Note that this is equivalent to considering the unit vector $\hat{z} = (x^*, 1)/\|(x^*, 1)\|$, $\lambda_r = |\pi(\hat{z})|^{-1/3}$ and:

$$\mu_\pi = \max_{\hat{z}} |\pi(\hat{z})|^{\frac{2}{3}}. \tag{41}$$

The eigenvector $\hat{z}_\pi$ is the argument of the maximum in the above expression.

Computation of α_π

We now turn to α_π. The matrix $\mathcal{Q}_{\pi,\alpha}$ at $\alpha = \alpha_\pi$ satisfies both 2. and 3. of Proposition 5.4, that is, its eigenvectors are $\hat{z}_\pi$ and $\hat{\boldsymbol{z}}_\pi^\perp$, and it is quadri-tangent to $\Lambda(\pi)$ and can be written as a transformation ϕ_π^{-1} of a diagonal matrix $H_{\lambda_{\alpha_\pi}}$:

$$\mathcal{Q}_{\pi,\alpha_\pi} = U_\pi^T \begin{pmatrix} \mu_\pi & 0 \\ 0 & \alpha_\pi \end{pmatrix} U_\pi = \phi_\pi^{-T} H_{\lambda_{\alpha_\pi}} \phi_\pi^{-1}, \tag{42}$$

where the expression of the diagonal matrix $H_{\lambda_{\alpha_\pi}}$ depends on the discriminant of π. Only the middle expression involving U_π is a diagonalized form of $\mathcal{Q}_{\pi,\alpha_\pi}$ since ϕ_π is not a rotation

in general. μ_π and α_π are eigenvalues for the eigenvectors $\hat{z}_\pi$ and $\hat{\boldsymbol{z}}_\pi^\perp$ respectively, i.e.:

$$\begin{aligned}\mathcal{Q}_{\alpha_\pi}\hat{z}_\pi &= \phi_\pi^{-T} H_{\lambda_{\alpha_\pi}} \phi_\pi^{-1}\hat{z}_\pi = \mu_\pi \hat{z}_\pi,\\ \mathcal{Q}_{\alpha_\pi}\hat{\boldsymbol{z}}_\pi^\perp &= \phi_\pi^{-T} H_{\lambda_{\alpha_\pi}} \phi_\pi^{-1}\hat{\boldsymbol{z}}_\pi^\perp = \alpha_\pi \hat{\boldsymbol{z}}_\pi^\perp.\end{aligned} \tag{43}$$

Defining:

$$\begin{aligned}\boldsymbol{z}_\pi^I &\triangleq \phi_\pi^{-1}\hat{z}_\pi, & \boldsymbol{z}_\pi^T &\triangleq \phi_\pi^T\hat{z}_\pi,\\ \boldsymbol{z}_\pi^{\perp,I} &\triangleq \phi_\pi^{-1}\hat{\boldsymbol{z}}_\pi^\perp, & \boldsymbol{z}_\pi^{\perp,T} &\triangleq \phi_\pi^T\hat{\boldsymbol{z}}_\pi^\perp,\end{aligned} \tag{44}$$

in which we have removed the "hat" for readability, these relations write:

$$H_{\lambda_{\alpha_\pi}}\boldsymbol{z}_\pi^I = \mu_\pi \boldsymbol{z}_\pi^T \quad \text{and} \quad H_{\lambda_{\alpha_\pi}}\boldsymbol{z}_\pi^{\perp,I} = \alpha_\pi \boldsymbol{z}_\pi^{\perp,T}, \tag{45}$$

which we solve for λ_{α_π} and α_π. If $\pi \circ \phi_\pi = \tilde{\pi}_2 = x^2 y$, then $H_{\lambda_{\alpha_\pi}} = \mathrm{diag}(\lambda_{\alpha_\pi}, 4/27\lambda_{\alpha_\pi}^2)$ and the equations for the first component of each eigenvectors are readily solved:

$$\lambda_{\alpha_\pi} = \mu_\pi \frac{z_{\pi,x}^T}{z_{\pi,x}^I}, \qquad \alpha_\pi = \lambda_{\alpha_\pi}\frac{z_{\pi,x}^{\perp,I}}{z_{\pi,x}^{\perp,T}}. \tag{46}$$

If $\pi \circ \phi_\pi = x(x^2 \pm 3y^2)$ ($\tilde{\pi}_3$ or $\tilde{\pi}_4$), then $H_{\lambda_{\alpha_\pi}} = \mathrm{diag}((4 \pm \lambda_{\alpha_\pi}^3)/3\lambda_{\alpha_\pi}^2, \lambda_{\alpha_\pi})$. Solving the equations for the second component of $\hat{z}_\pi$ and $\hat{\boldsymbol{z}}_\pi^\perp$, we have:

$$\lambda_{\alpha_\pi} = \mu_\pi \frac{z_{\pi,y}^T}{z_{\pi,y}^I}, \qquad \alpha_\pi = \lambda_{\alpha_\pi}\frac{z_{\pi,y}^{\perp,I}}{z_{\pi,y}^{\perp,T}}. \tag{47}$$

Computation of λ_α

Finally, we determine λ_α by imposing $\det(\mathcal{Q}_{\pi,\alpha} - \alpha I) = 0$. Using the identity:

$$\det(A+B) = \det A + \det B + \det A\ \mathrm{tr}(A^{-1}B) \tag{48}$$

with $A = -\alpha I$ and $B = \mathcal{Q}_{\pi,\alpha} = \phi_\pi^{-T} H_{\lambda_\alpha}\phi_\pi^{-1}$, we have:

$$\begin{aligned}\det(-\alpha I + \mathcal{Q}_{\pi,\alpha}) &= \det(-\alpha I) + \det(\mathcal{Q}_{\pi,\alpha}) + \det(-\alpha I)\ \mathrm{tr}\left(-\frac{\mathcal{Q}_{\pi,\alpha}}{\alpha}\right)\\ &= \alpha^2 + \det(\phi_\pi^{-1})^2 \det(H_{\lambda_\alpha}) - \alpha\,\mathrm{tr}(\mathcal{Q}_{\pi,\alpha}).\end{aligned} \tag{49}$$

Let $C_1 = (\phi_{\pi,11}^{-1})^2 + (\phi_{\pi,12}^{-1})^2$, $C_2 = (\phi_{\pi,21}^{-1})^2 + (\phi_{\pi,22}^{-1})^2$ and $D = \det(\phi_\pi^{-1})^2$, then:

$$\begin{aligned} \det(H_{\lambda_\alpha}) = \frac{4}{27\lambda_\alpha} \text{ and } \operatorname{tr}(\mathcal{Q}_{\pi,\alpha}) = \lambda_\alpha C_1 + \frac{4}{27\lambda_\alpha^2} C_2 \quad & \text{if } \pi \circ \phi = \tilde{\pi}_2, \\ \det(H_{\lambda_\alpha}) = \frac{4 \pm \lambda_\alpha^3}{3\lambda_\alpha} \text{ and } \operatorname{tr}(\mathcal{Q}_{\pi,\alpha}) = \frac{4 \pm \lambda_\alpha^3}{3\lambda_\alpha^2} C_1 + \lambda_\alpha C_2 \quad & \text{if } \pi \circ \phi = \tilde{\pi}_3 \text{ or } \tilde{\pi}_4. \end{aligned} \tag{50}$$

This yields the following polynomials of order 3 and 4 in λ_α:

$$\begin{aligned} -27\alpha C_1 \lambda_\alpha^3 + 27\alpha^2 \lambda_\alpha^2 + 4D\lambda_\alpha - 4\alpha C_2 = 0 \quad & \text{if } \pi \circ \phi = \tilde{\pi}_2, \\ \pm D\lambda_\alpha^4 - \alpha(3C_2 \pm C_1)\lambda_\alpha^3 + 3\alpha^2 \lambda_\alpha^2 + 4D\lambda_\alpha - 4\alpha = 0 \quad & \text{if } \pi \circ \phi = \tilde{\pi}_3 \text{ or } \tilde{\pi}_4. \end{aligned} \tag{51}$$

The optimal λ_α is chosen as the strictly positive root if $\pi \circ \phi = \tilde{\pi}_2$ (resp. in $(0, 2]$ for $\tilde{\pi}_3$ and in $(0, 1]$ for $\tilde{\pi}_4$) that maximizes the ellipse area, hence that minimizes $\det(\mathcal{Q}_{\pi,\alpha})$. Since $\det(\phi_\pi^{-1})^2$ is constant for a given polynomial π, it is more stable to minimize $\det(H_{\lambda_\alpha})$ for very large or very small values of λ_α as $\det(\mathcal{Q}_{\pi,\alpha})$ becomes less accurate or even negative.

A.5 Appendix of Chapter 6

A.5.1 Signed curvatures

The signed curvatures of γ_{iso} and γ_{grad} can be obtained by imposing that f on γ_{iso} is approximated to second order and that ∇f is aligned with γ'_{grad} up to first order. Writing the quadratic expansion around $\boldsymbol{x}_0$ at $\boldsymbol{x} = \gamma(s)$:

$$f(\boldsymbol{x}) = f(\boldsymbol{x}_0) + \nabla f(\boldsymbol{x}_0) \cdot (\boldsymbol{x} - \boldsymbol{x}_0) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^T H_f(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0) + \mathcal{O}(\|\boldsymbol{x} - \boldsymbol{x}_0\|^3), \tag{52}$$

we have after substituting $\gamma_{\text{iso}}(s) - \boldsymbol{x}_0$:

$$\begin{aligned} f(\gamma_{\text{iso}}(s)) - f(\boldsymbol{x}_0) = 0 = \left(\nabla f(\boldsymbol{x}_0) \cdot k_{\text{iso}} \boldsymbol{g} + \boldsymbol{g}^\perp \cdot H_f(\boldsymbol{x}_0) \cdot \boldsymbol{g}^\perp\right) \frac{s^2}{2} + \mathcal{O}(s^3) \\ \downarrow \\ k_{\text{iso}} = -\frac{\boldsymbol{g}^\perp \cdot H_f(\boldsymbol{x}_0) \cdot \boldsymbol{g}^\perp}{\nabla f(\boldsymbol{x}_0) \cdot \boldsymbol{g}}, \end{aligned} \tag{53}$$

and similarly for the linear expansion of $\nabla f(\boldsymbol{x})$ around $\boldsymbol{x}_0$:

$$\begin{aligned} \gamma'_{\text{grad}}(s) \propto \nabla f(\boldsymbol{x}) &= \nabla f(\boldsymbol{x}_0) + H_f(\boldsymbol{x}_0) \cdot (\boldsymbol{x} - \boldsymbol{x}_0) + \mathcal{O}(\|\boldsymbol{x} - \boldsymbol{x}_0\|^2) \\ &\downarrow \\ \boldsymbol{g} + sk_{\text{grad}}\boldsymbol{g}^{\perp} &\propto \nabla f(\boldsymbol{x}_0) + sH_f(\boldsymbol{x}_0) \cdot \boldsymbol{g} + \mathcal{O}(s^2). \end{aligned} \tag{54}$$

The vector equality is obtained by normalizing the right-hand side by $\|\nabla f(\boldsymbol{x}_0)\|$, yielding:

$$k_{\text{grad}} = \frac{\boldsymbol{g}^{\perp} \cdot H_f(\boldsymbol{x}_0) \cdot \boldsymbol{g}}{\|\nabla f(\boldsymbol{x}_0)\| \, \boldsymbol{g}^{\perp} \cdot \boldsymbol{g}^{\perp}} = \frac{\boldsymbol{g}^{\perp} \cdot H_f(\boldsymbol{x}_0) \cdot \boldsymbol{g}}{\|\nabla f(\boldsymbol{x}_0)\|}. \tag{55}$$

APPENDIX B HIGH-ORDER DERIVATIVES RECOVERY

Anisotropic adaptation requires a posteriori error estimators, which rely on derivatives of order $m \geq 2$ of the field of interest. For linear interpolation, for instance, the mesh adaptation process is driven by the Hessian of the numerical solution, whereas derivatives of order 3 or higher are required for high-order interpolation schemes on straight meshes. Because of the popularity of linear elements, there is a rich literature on Hessian recovery methods, see for example [133–135] for a review and comparison of such operators. The recovery of higher derivatives, in contrast, is less documented, but we can mention the work of Kuate [90] and Mbinky [10]. In this thesis, derivatives of arbitrary order are recovered with the least-squares projection of Zhang and Naga [85], a patch-based recovery similar to Zhu and Zienkiewicz's postprocessing [84].

Zhang and Naga's Polynomial Preserving Recovery (PPR) method is briefly presented in this Appendix. This includes the practical computation of the least-squares polynomial, the construction of element and vertex patches around mesh vertices as well as some convergence properties of the PPR operator for smooth functions. Some modifications applied to improve its behaviour on anisotropic meshes are presented, as well as our implementation on $\mathcal{P}^2$ triangulations. An immediate high-order ($m = 3, 4$) extension of the PPR-based Hessian recovery [135] is described. To our best knowledge, the use of this scheme to recover derivatives of order $m \geq 3$ is reported for the first time. The presentation concludes with numerical tests showing the approximation capabilities of the PPR with respect to high-order derivatives on both regular and anisotropic meshes.

Notations

We introduce some notations which mostly follow [136]. For a triangulation $\mathcal{T}$ in the physical space Ω, recall the space V_h^k of continuous and piecewise polynomial functions of degree up to k is:

$$V_h^k = \mathbb{P}^k(\mathcal{T}) = \left\{ f_h \in L^2(\Omega) \mid f_h|_K \in \mathbb{P}^k(K),\ \forall K \in \mathcal{T} \right\}. \tag{56}$$

These functions are polynomial in the physical space. The set of mesh vertices and interior vertices are denoted by $\mathcal{N}_h$ and $\mathring{\mathcal{N}}_h \triangleq \mathcal{N}_h \backslash (\mathcal{N}_h \cap \partial\Omega)$, respectively. A **patch of elements** surrounding a mesh vertex $\boldsymbol{x}$ is denoted by $\omega_{\boldsymbol{x}}$. The precise definition of $\omega_{\boldsymbol{x}}$ is given further and changes if $\boldsymbol{x}$ is an interior, boundary or edge vertex, the latter occurring only for $\mathcal{P}^2$ meshes. The **patch of vertices** surrounding $\boldsymbol{x}$ is defined by $\omega_{\boldsymbol{x}}^v \triangleq \omega_{\boldsymbol{x}} \cap \mathcal{N}_h$.

B.1 The Polynomial Preserving Recovery

The polynomial preserving recovery (PPR) was introduced in a series of papers by Zhang and Naga [85, 136, 137] as an improvement of the Zhu and Zienkiewicz's gradient recovery, which it outperforms on selected mesh patterns. Given a discrete solution $f_h \in V_h^k$, the PPR is an operator $G_h : V_h^k \to V_h^k \times V_h^k$ first defined by its values at the mesh vertices $G_h f_h(\boldsymbol{x})$, then extended to the domain using the basis of V_h^k:

$$G_h f_h \triangleq \sum_{\boldsymbol{x} \in \mathcal{N}_h} G_h f_h(\boldsymbol{x}) \phi(\boldsymbol{x}). \tag{57}$$

The nodal values are determined by fitting a least-squares polynomial to the discrete solution f_h on a patch surrounding each vertex. We let $p_{\boldsymbol{x}} \in \mathbb{P}^{k+1}(\omega_{\boldsymbol{x}})$ denote the polynomial that best fits the discrete solution f_h in a least-squares sense on the patch $\omega_{\boldsymbol{x}}$, that is:

$$p_{\boldsymbol{x}} = \arg \min_{p \in \mathbb{P}^{k+1}(\omega_{\boldsymbol{x}})} \sum_{\boldsymbol{x} \in \omega_{\boldsymbol{x}}^v} (f_h(\boldsymbol{x}) - p(\boldsymbol{x}))^2. \tag{58}$$

The value of $G_h f_h$ at the mesh vertices is then defined as the gradient of $p_{\boldsymbol{x}}$ evaluated at $\boldsymbol{x}$:

$$G_h f_h(\boldsymbol{x}) \triangleq \nabla p_{\boldsymbol{x}}(\boldsymbol{x}). \tag{59}$$

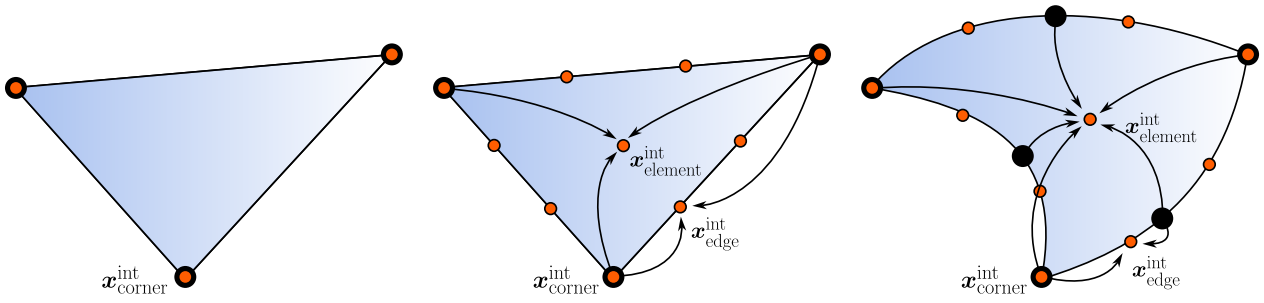
On $\mathcal{P}^1$ triangles, this completely defines $G_h u_h$ for linear interpolants whose interpolation nodes coincide with the mesh vertices, whereas nodal values for higher-order elements, which do not have an associated mesh vertex, are obtained by averaging the evaluation of adjacent polynomials. Note that this is only required when the recovery operator has to be evaluated inside the triangles, to compute norms and error, for instance. Riemannian metrics, as discussed in Chapter 5, are computed at the mesh vertices and do not require additional work. If $\boldsymbol{x}_{\text{edge}}^{\text{int}}$ is the position of an interpolation node on the edge $\boldsymbol{x}_1\boldsymbol{x}_2$, $G_h f_h$ is defined by:

$$G_h f_h(\boldsymbol{x}_{\text{edge}}^{\text{int}}) \triangleq \alpha \nabla p_{\boldsymbol{x}_1}(\boldsymbol{x}_{\text{edge}}^{\text{int}}) + (1-\alpha) \nabla p_{\boldsymbol{x}_2}(\boldsymbol{x}_{\text{edge}}^{\text{int}}), \tag{60}$$

with $\alpha = d_2/(d_1+d_2)$ the ratio of distances $d_i = \|\boldsymbol{x}_{\text{edge}}^{\text{int}} - \boldsymbol{x}_i\|$. The values of $G_h f_h$ at element nodes $\boldsymbol{x}_{\text{element}}$ are obtained in the same way, using barycentric coordinates α_j as weights:

$$G_h f_h(\boldsymbol{x}_{\text{element}}^{\text{int}}) \triangleq \sum_{j=1}^{3} \alpha_j \nabla p_{\boldsymbol{x}_j}(\boldsymbol{x}_{\text{element}}^{\text{int}}). \tag{61}$$

To avoid instabilities when fitting a least-squares polynomial on small patches, the compu-



$(\mathcal{P}^1, \Pi^1)$ $(\mathcal{P}^1, \Pi^3)$ $(\mathcal{P}^2, \Pi^3)$

Figure B.1 Recoveries are computed at mesh vertices (in black), then evaluated and averaged at interpolation nodes (in orange).

tation is performed on a scaled patch defined by $\hat{\omega}_{\boldsymbol{x}} = (\omega_{\boldsymbol{x}} - \boldsymbol{x})/h_{\boldsymbol{x}}$, with $h_{\boldsymbol{x}}$ the dimension of the bounding box around $\omega_{\boldsymbol{x}}$. To obtain better results with highly anisotropic meshes, we consider as in Briffard [138] the anisotropic scaling:

$$\hat{\omega}_{\boldsymbol{x}} = \left(\frac{\omega_{\boldsymbol{x}} - x^1}{h_{\boldsymbol{x},1}}, \frac{\omega_{\boldsymbol{x}} - x^2}{h_{\boldsymbol{x},2}} \right) \triangleq F_{\boldsymbol{x}}(\omega_{\boldsymbol{x}}) \quad \text{with} \quad h_{\boldsymbol{x},i} = \max_{\boldsymbol{x}_j \in \omega^v_{\boldsymbol{x}}} |x^i_j - x^i|, \tag{62}$$

so that we search $\hat{p}_{\boldsymbol{x}} \in \mathbb{P}^{k+1}(\hat{\omega}_{\boldsymbol{x}})$ such that $p_{\boldsymbol{x}} = \hat{p}_{\boldsymbol{x}} \circ F_{\boldsymbol{x}}$. The scaled polynomial writes $\hat{p}_{\boldsymbol{x}}(\hat{\boldsymbol{y}}) = \hat{\boldsymbol{P}}^T \boldsymbol{c}_{\boldsymbol{x}}$, where $\boldsymbol{P} = (p_1(\hat{\boldsymbol{y}}), ..., p_{n_{k+1}}\hat{\boldsymbol{y}})^T$ are the monomials on the scaled patch and $\boldsymbol{c}_{\boldsymbol{x}} = (c_1, ..., c_{n_{k+1}})$ the coefficients to determine. The dimension of each vector is $n_{k+1} = |\widehat{K}_2^{k+1}| = (k+2)(k+3)/2$. These coefficients are obtained by solving the least-squares problem:

$$A^T A \boldsymbol{c}_{\boldsymbol{x}} = A^T \boldsymbol{b}_{\boldsymbol{x}}, \tag{63}$$

with the matrix and right-hand side:

$$A = \begin{pmatrix} \boldsymbol{P}(\hat{\boldsymbol{x}}_1)^T \\ \vdots \\ \boldsymbol{P}(\hat{\boldsymbol{x}}_{N(\omega^v_{\boldsymbol{x}})})^T \end{pmatrix} \text{ and } \boldsymbol{b}_{\boldsymbol{x}} = \begin{pmatrix} u_h(\boldsymbol{x}_1) \\ \vdots \\ u_h(\boldsymbol{x}_{N(\omega^v_{\boldsymbol{x}})}) \end{pmatrix}, \tag{64}$$

where $N(\omega^v_{\boldsymbol{x}})$ is the number of mesh vertices in the patch. The matrix $A^T A$ is a real SPD matrix of order $n_{k+1} \times n_{k+1}$. For (63) to have a unique solution, the rank of $A^T A$ must be the dimension of the coefficient vector, i.e., n_{k+1}. The rank of $A^T A$ depends only on the position of the vertices in the patch, thus some patches might have to be enlarged to obtain a well-posed problem.

Extension to $\mathcal{P}^2$ meshes

On $\mathcal{P}^2$ triangles, the derivatives are additionally recovered at the edge vertices, to yield more accurate interpolated metrics. The only difference between edge and $\mathcal{P}^1$ (corner) vertices is the definition of their patch of elements for the least-squares projection. Then, the PPR operator is defined at the interpolation nodes of the chosen interpolant. Linear (Π^1) and quadratic (Π^2) interpolation does not require additional work as the nodal values $G_h f_h(\boldsymbol{x})$ are available and given by (59). Moreover, the input numerical solution f_h is stored at these vertices. Starting with cubic interpolation, f_h needs to be interpolated to evaluate the least-squares right-hand side at edge and element nodes, and averaging is required to define the PPR operator at these nodes, see Fig. B.1. If $\boldsymbol{x}_{\text{edge}}^{\text{int}}$ is an interpolation node on an edge, the evaluation at $\boldsymbol{x}_{\text{edge}}^{\text{int}}$ of the two nearest polynomials on the edge are weighted by the distance, similarly to (60). For the interpolation nodes lying inside the element, we interpolate using Lagrange quadratic functions, as there is no immediate extension of barycentric coordinates for $\mathcal{P}^2$ triangles:

$$G_h f_h(\boldsymbol{x}_{\text{element}}^{\text{int}}) \triangleq \sum_{j=1}^{6} \phi_j^{\text{geo}}(\boldsymbol{x}_{\text{element}}^{\text{int}}) \nabla p_{\boldsymbol{x}_j}(\boldsymbol{x}_{\text{element}}^{\text{int}}). \tag{65}$$

Although three weights are negative at the barycenter, we observed in Section B.5 that this averaging does not hinder the convergence of the recoveries in practice. An alternative would be to use the quadratic Bernstein polynomials, which are everywhere positive, however it would require recovering the numerical solution at Bézier control points or using a Lagrange-to-Bézier transformation [17].

Patches of elements

We define the patches $\omega_{\boldsymbol{x}}$ for interior, boundary and edge vertices. Following [136], these are defined using the union of elements in the first n layers around a mesh vertex $\boldsymbol{x}$:

$$\mathcal{L}(\boldsymbol{x}, n) \triangleq \begin{cases} \boldsymbol{x} & \text{if } n = 1, \\ \bigcup \{K \in \mathcal{T}_h \mid K \cap \mathcal{L}(\boldsymbol{x}, n-1) \neq \varnothing\} & \text{otherwise.} \end{cases} \tag{66}$$

Since we fit a polynomial $p_{\boldsymbol{x}} \in \mathbb{P}^{k+1}(\omega_{\boldsymbol{x}})$ in a space of dimension $n_{k+1} = (k+2)(k+3)/2$, the patch $\omega_{\boldsymbol{x}}$ must have at least n_{k+1} vertices to obtain a unique solution. The element patch $\omega_{\boldsymbol{x}}$ around an interior vertex is thus defined by:

$$\omega_{\boldsymbol{x},\text{interior}} \triangleq \mathcal{L}(\boldsymbol{x}, n_0), \tag{67}$$

where n_0 is the smallest integer such that $N(\mathcal{L}(\boldsymbol{x}, n_0)) \geq n_{k+1}$, see Fig. B.2. The vertices in $\omega_{\boldsymbol{x}}$ may be placed in such a way that the least-squares matrix $A^T A$ is not of full rank. When it is the case, the patch size is increased to the minimum number of layers required to have $\text{rank}(A^T A) = n_{k+1}$. Because the length of incident edges to a vertex $\boldsymbol{x}$ are expected to vary greatly from one direction to another in anisotropic meshes, this definition of $\omega_{\boldsymbol{x},\text{interior}}$ is preferred to the original definition in [85] based on the ball of radius equal to the length of the longest incident edge to $\boldsymbol{x}$. Considering only the longest edge leads to large vertex patches that span along both directions of anisotropy, yielding undesired smoothing, as the information may be taken far from the feature to recover. In contrast, $\omega_{\boldsymbol{x},\text{interior}}$ mimics the anisotropic unit ball for the metric induced by the input mesh. In an adaptation loop, this mesh is expected to be a unit, with all incident edges of quasi-unit length. Considering the first layers of adjacent elements thus amounts to consider elements inside roughly an integer multiple of the unit ball.

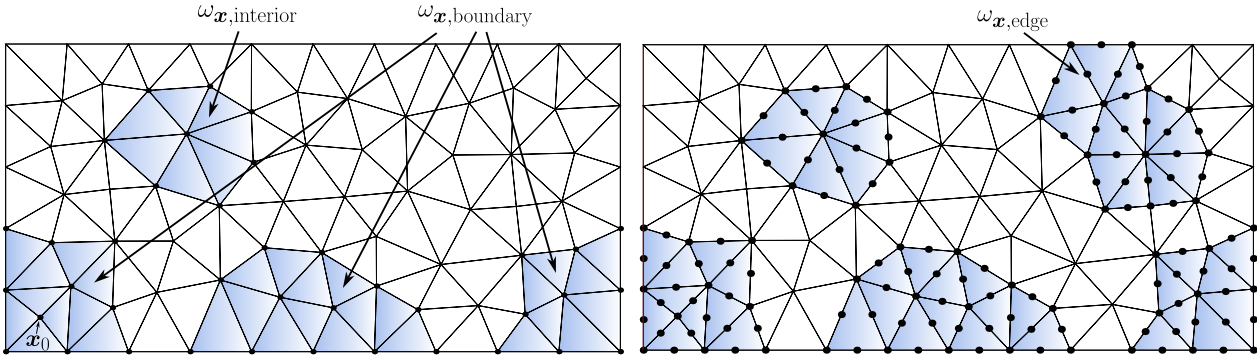


Figure B.2 Element patches $\omega_{\boldsymbol{x}}$ and vertex patches $\omega^v_{\boldsymbol{x}}$ for interior, boundary and edge vertices. Left: patches to fit a quadratic polynomial on a $\mathcal{P}^1$ mesh ($n_{k+1} = 6$ vertices required). The initial patch of the interior vertex $\boldsymbol{x}_0$ contains only 5 vertices, and had to be enlarged. Right: patches to fit a cubic polynomial on a $\mathcal{P}^2$ mesh ($n_{k+1} = 10$ vertices required).

The patches for boundary vertices are defined from the *Strategy 2* in Guo et al. [139]. It uses an intermediate patch $\omega_{m_0} = \mathcal{L}(\boldsymbol{x}, m_0)$, with m_0 the smallest integer such that ω_{m_0} contains at least one interior vertex. Typically, $m_0 = 1$ except maybe at corners of the domain. Then, the $\omega_{\boldsymbol{x}}$ is defined as the union of the patches of the interior vertices in ω_{m_0}, supplemented by the first layer of elements surrounding each boundary vertex in ω_{m_0}. This formally writes as:

$$\omega_{\boldsymbol{x},\text{boundary}} \triangleq \left(\bigcup_{\bar{\boldsymbol{x}} \in \omega_{m_0} \cap \mathring{\mathcal{N}}_h} \omega_{\bar{\boldsymbol{x}}} \right) \bigcup \left(\bigcup_{\bar{\boldsymbol{x}} \in \omega_{m_0} \cap \mathcal{N}_h \cap \partial\Omega} \mathcal{L}(\bar{\boldsymbol{x}}, 1) \right). \tag{68}$$

Finally, for $\mathcal{P}^2$ meshes, we simply define $\omega_{\boldsymbol{x}}$ at an edge vertex as the union of the patches of

the extremities $\boldsymbol{x}_1, \boldsymbol{x}_2$ of the edge:

$$\omega_{\boldsymbol{x},\text{edge}} \triangleq \omega_{\boldsymbol{x}_1} \cup \omega_{\boldsymbol{x}_2}. \tag{69}$$

Fig. B.2 illustrates these three definitions for different kind of vertices.

B.2 Properties of the PPR operator

From the analyses of [85, 136, 137, 140], we briefly detail the properties of the PPR operator G_h. The proofs are given in these papers and the references therein.

1. **Polynomial preserving property.** G_h preserves polynomials of degree up to $k+1$ on arbitrary meshes, that is, $G_h f_h = \nabla f_h$ for any $f_h \in \mathbb{P}^{k+1}(\Omega)$, since the least-squares recovery yields the exact f_h in that case. This yields the unconditional superconvergence property when applying the PPR operator to the interpolant of f ([136]):

$$\|\nabla u - G_h \Pi^k u\|_{L^2(\Omega)} \leq C h^{k+1} |u|_{H^{k+2}(\Omega)}, \forall u \in H^{k+2}(\Omega), \tag{70}$$

   Polynomials of degree $k+2$ are also preserved at mesh vertices that are symmetry centers for the nodes of their patch, although this is not expected on arbitrary meshes.

2. **Superconvergence.** If conditions on the mesh are satisfied, namely G_h is *bounded* (in the sense of linear operators) and satisfies the *superconvergence condition*, then the recovered gradient $G_h f_h$ *superconverges* to ∇f. These two conditions are respected by a wide range of meshes, notably meshes generated by the Delaunay triangulation [141, 142]. For linear interpolation and for $f \in W^{\infty,3}(\Omega)$ ([137], Th. 4.1):

$$\|\nabla f - G_h f_h\|_{L^2(\Omega)} \leq C h^{1+\rho} \|f\|_{3,\infty,\Omega}, \quad \text{with} \quad \rho \geq 0. \tag{71}$$

   From [136, 137], we can infer the general result for $f \in W^{\infty,k+2}(\Omega)$:

$$\|\nabla f - G_h f_h\|_{L^2(\Omega)} \leq C h^{k+\rho} \|f\|_{k+2,\infty,\Omega}, \quad \text{with} \quad \rho \geq 0. \tag{72}$$

   Superconvergence results also hold on anisotropic meshes under some conditions ([142], Th 3.2 and [140], Th. 3.3 and 3.4). The convergence results are more involved, but show that the convergence rate for $\|\nabla f - G_h f_h\|_{L^2(\Omega)}$ can reach $\mathcal{O}(h^{k+1})$ on anisotropic meshes.

3. **Asymptotically exact error estimator.** When the recovered gradient obtained with

the PPR operator superconverges to the exact gradient, it can be used to derive an error estimator for the gradient of $e_h = f - f_h$, see [137].

To summarize, we can expect the following behaviour from the recovered gradient applied to sufficiently smooth functions:

- Convergence of order $k+1$ when applying to the interpolant $\Pi^k f$ on mildly structured meshes, and of order $k+2$ on uniform meshes with k even;
- Convergence of order $k+\rho$, $\rho \geq 0$ when applying to the finite element solution f_h on mildly structured meshes.

B.3 Hessian matrix recovery

The gradient recovery operator can be applied twice to recover the Hessian matrix, as suggested in Guo et al. [135]. The recovered Hessian matrix $H_h f_h$ of degree k defined this way is:

$$H_h f_h = \begin{pmatrix} G_h^x(G_h^x f_h) & \dfrac{G_h^x(G_h^y f_h) + G_h^y(G_h^x f_h)}{2} \\ \text{Sym} & G_h^y(G_h^y f_h) \end{pmatrix}. \tag{73}$$

The Hessian is obtained by applying the gradient recovery operator twice and averaging the mixed derivatives. Other recovery techniques, such as Zhu and Zienkiewicz's recovery, exist in the literature and we could consider mixing techniques when recovering the derivatives of f_h. Numerical results in [135] show however that the PPR-PPR combination, i.e., using the PPR to recover the gradient of both f_h and $G_h f_h$, is more accurate than other combinations by up to one-half order of convergence. The recovered Hessian enjoys approximation properties similar to those of the recovered gradient:

1. **Polynomial preserving.** H_h preserves polynomials of degree up to $k+1$ on arbitrary meshes, that is, $H_h \Pi^k f = Hf$ for $f \in \mathbb{P}^{k+1}(\Omega)$. This yields the local superconvergence property for the Hessian of the interpolant and for $f \in W^{\infty,k+2}(\omega_{\boldsymbol{x}})$ ([135], Th. 3.9):

$$\|Hu - H_h \Pi^k f\|_{L^\infty(\omega_{\boldsymbol{x}})} \leq C h^k |f|_{W_\infty^{k+2}(\omega_{\boldsymbol{x}})}. \tag{74}$$

2. **Superconvergence.** For smooth enough functions f and certain classes of meshes, we have the following results for linear and quadratic interpolation ([135], Th. 4.3 and 4.5):

- *Linear elements.* If the mesh satisfies the (α, σ) condition (see e.g. [137]), then:

$$\|Hf - H_h f_h\|_{L^2(\Omega)} \leq h^\rho \|f\|_{W^3_\infty(\Omega)}, \quad \text{with} \quad \rho \geq 0. \tag{75}$$

- *Quadratic elements.* If the mesh is uniform or *strongly regular* ($\alpha = 1, \sigma = \infty$) [137], then:

$$\|Hf - H_h f_h\|_{L^2(\Omega)} \leq h^2 \|f\|_{H^4(\Omega)}. \tag{76}$$

Thus, we can expect convergence of order k for the Hessian of the interpolant and of order $k - 1 + \rho$ for the Hessian of the numerical solution.

B.4 Higher-order derivatives recovery

In a similar fashion, we propose to define the array of third-order derivatives according to the following illustration:

$$D_h^{(3)} f_h = \begin{pmatrix} G_h^x(G_h^x(G_h^x f_h)) & \overline{G}_h^{xxy}(f_h) & \overline{G}_h^{xyy}(f_h) \\ & \overline{G}_h^{xxy}(f_h) & G_h^y(G_h^y(G_h^y f_h)) \\ \text{Sym} & & \overline{G}_h^{xyy}(f_h) \end{pmatrix},$$

with for example:

$$\overline{G}_h^{xxy}(f_h) = \frac{G_h^x(G_h^x(G_h^y f_h)) + G_h^x(G_h^y(G_h^x f_h)) + G_h^y(G_h^x(G_h^x f_h))}{3}. \tag{77}$$

Higher-order derivatives are obtained by applying the gradient recovery operator n times and averaging the mixed components. To our best knowledge, the use of the PPR operator to recover derivatives of order higher than 2 is not documented in the literature. We can still affirm, however, that they preserve polynomials of order up to $k+1$ on arbitrary meshes, that is, $D_h^{(m)} \Pi^k f = d^{(m)} f$ for any $f \in \mathbb{P}^{k+1}(\Omega)$, by applying recursively the proof of Theorem 3.5 in [135]. From (70) and (74), we can infer that one order of convergence is lost per derivative order when applying the PPR to the interpolant $\Pi^k f$ on unstructured meshes, and similarly for f_h.

B.5 Verification

We consider two test cases to verify the implementation of the polynomial preserving recovery and its capabilities to recover higher-order derivatives:

Test 1. We solve on the unit square $\Omega = [0,1]^2$ the Poisson problem:

$$\begin{aligned} -\Delta f &= g \quad \text{in } \Omega, \\ f &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{78}$$

with homogeneous boundary condition and a manufactured source term chosen so that the analytic solution is $f_1(x,y) = \sin(\pi x)\sin(\pi y)$.

Test 2. We solve a Poisson problem on the square $\Omega = [-1,1]^2$ with nonhomogeneous Dirichlet boundary condition. The source term and boundary conditions are set so that the analytic solution is:

$$f_2(x,y) = \frac{1}{2}\left[1 + \tanh\left(a\left(bx - \frac{\sin(c\pi y)}{d}\right)\right)\right], \quad \text{with} \quad a = 10, b = 1/2, c = 2, d = 4. \tag{79}$$

For each test, the derivatives of order up to 4 of f_h and $\Pi^k f$ are recovered by successively applying the PPR operator.

Influence of the boundary. Recovery methods are known to deteriorate near the boundaries. Following the methodology in [85, 135, 136, 139], we examine seperately the behaviour of the PPR on the full domain and on an inner domain excluding the boundary. To do so, we consider the set $\mathcal{N}_{h,1} \triangleq \{\boldsymbol{x} \in \mathcal{N}_h \mid \operatorname{dist}(\boldsymbol{x}, \partial\Omega) \geq \ell\}$ of vertices at a distance ℓ from the boundary. The domain is partitioned into $\overline{\Omega} = \Omega_1 \cup \Omega_2$, where:

$$\Omega_1 = \bigcup\{K \in \mathcal{T}_h \mid K \text{ has all its vertices in } \mathcal{N}_{h,1}\}, \tag{80}$$

so that Ω_1 contains only triangles with all vertices at prescribed distance from the boundary. For both tests, we set $\ell = 0.125$ and define a discrete maximum norm $\|\cdot\|_{\infty,\mathcal{N}_{h,1}}$ over the mesh vertices in $\mathcal{N}_{h,1}$ and a continuous norm $\|\cdot\|_{L^2(\Omega_1)}$ by:

$$\|\nabla f - G_h f_h\|_{\infty,\mathcal{N}_{h,1}} \triangleq \max_{\boldsymbol{x}\in\mathcal{N}_{h,1}} \max_{j=1,2} \left(\frac{\partial f}{\partial x_j}(\boldsymbol{x}) - G_h^j f_h(\boldsymbol{x})\right), \tag{81}$$

$$\|\nabla f - G_h f_h\|_{L^2(\Omega_1)} \triangleq \sqrt{\int_{\Omega_1} \sum_{j=1,2} \left(\frac{\partial f}{\partial x_j} - G_h^j f_h \right)^2 dx}. \tag{82}$$

We also define the norms $\|\cdot\|_\infty$ and $\|\cdot\|_{L^2}$ to include the boundary by replacing Ω_1 by Ω in the expressions above. The norms associated to the Hessian and higher-order derivatives are computed similarly, considering all 4, 8 and 16 scalar components respectively. In the following, all convergence results with a rate superior to $k-(d-1)$, with d the order of the recovered derivative, are said to be *superconvergence* results, such as an order $2 > (1-(1-1)) = 1$ for the recovered gradient ($d=1$) on linear ($k=1$) elements. Since results are presented in 2D for both uniform and anisotropic meshes, all error graphs are plotted against the square root of the number of $\mathcal{P}^1$ mesh vertices N_v.

Families of meshes. We consider the families of meshes shown on Fig. B.3: uniform meshes, Delaunay meshes and straight adapted anisotropic meshes. The straight anisotropic meshes are adapted to the optimal metric described in Chapter 5. These metrics require the high-order derivatives of f: analytical derivatives were used to remove the effect of the recovery method on the adapted meshes. We present results for $(\mathcal{P}^1, \Pi^3)$ interpolation on uniform meshes (Fig. B.4), for $(\mathcal{P}^1, \Pi^3)$ interpolation on Delaunay meshes (Fig. B.5), and for $(\mathcal{P}^1, \Pi^3)$ interpolation on straight anisotropic meshes (Fig. B.6)

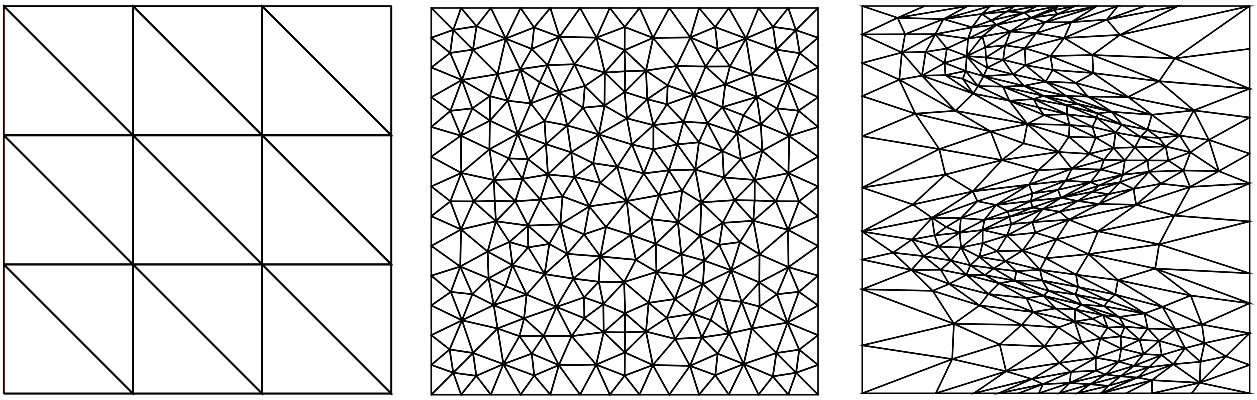


Figure B.3 Families of meshes considered for convergence studies. From left to right: uniform meshes with the "regular" pattern [85], Delaunay meshes and straight anisotropic meshes adapted to the solution f_2.

Results. The results for u_h and $\Pi^k u$ being mostly identical, the following comments hold for both. For all results, the $\mathcal{O}(h^k)$ convergence of the H^1-seminorm of the error on the finite element solution is presented as a baseline. On the interior of the uniform meshes (Fig. B.4,

bottom 4 figures), all derivatives exhibit $\mathcal{O}(h^{k+1}) = \mathcal{O}(h^4)$ superconvergence in both pointwise and L^2 norms. When the boundary is included (top 2 figures), the recovered derivatives converge as expected and an order of convergence is lost per derivative. In particular, the gradient superconverges as $\mathcal{O}(h^{k+1})$ pointwise, the Hessian as $\mathcal{O}(h^k)$, etc.

Results on Delaunay meshes, Fig. B.5, show the behaviour of the PPR operator on nonuniform meshes, which are more common in finite element analyses. On these meshes, the convergence order for $G_h\Pi^k u$ is expected from the polynomial preserving property, while the superconvergence of $G_h u_h$ is ensured from the fact that the Delaunay meshes satisfy the (α, σ) condition [137]. Superconvergence is observed for all derivatives including the boundaries. For test 1, derivatives of increasing order converge in L^2 norm (top right figure) with rates 3.9 (vs. $k+1=4$), 3.5 (vs. $k=3$), 2.5 (vs. $k-1=2$) and 1.4 (vs $k-2=1$). Inside the domain, the $\mathcal{O}(h^4)$ rate granted by the symmetry of the regular pattern is no longer observed, and we lose one order of convergence per derivative instead. Unlike uniform meshes where all derivatives converge even with low order interpolant, derivatives of order $m > k+1$ do not converge in general on Delaunay meshes (not shown).

Lastly, the recovered derivatives on straight anisotropic meshes exhibit a very similar behaviour as on Delaunay meshes, see Fig. B.6. This confirms that the definition of element patches is adequate for anisotropic meshes. The derivatives of order 4, required to compute the optimal metric, converge in L^2 norm with rate 1.4 including the boundary (top right) and 1.7 inside the domain (middle right). This rate goes up to 2.9 inside the domain for the test 2 (bottom right), thus the fourth order derivatives converge with the same rate as the H^1 seminorm of the error.

B.6 Conclusion

We presented the least-squares operator used to recover high-order derivatives in this thesis. It is a critical ingredient in the mesh adaptation loop, as the quality of the high-order derivatives directly affect the error estimates used in adaptive meshing. This operator consists in m applications of the Zhang ang Naga gradient recovery operator to the numerical solution f_h. The convergence properties of this operator were also recalled. Namely, the gradient and Hessian recovered with the PPR converge with accelerated rate to the exact derivatives on uniform meshes, and may superconverge on general meshes under some geometric conditions. Although we did not explicitly validate those geometric conditions, this superconvergence was well identified in our numerical experiments. Generally speaking, superconvergence was ob-

served for derivatives of all tested orders (up to 4) on all tested meshes, including anisotropic triangulations. On the interior of uniform meshes, a convergence rate of $k+1$ was obtained for derivatives of all orders. On unstructured meshes, the gradient usually superconverges with order $k+\rho$, with $\rho \geq 0$, and we lose one order of convergence per subsequent recovered derivative. In particular, we are able to recover derivatives of order 4 on cubic interpolants with $\mathcal{O}(h^r)$ convergence, with $1 \leq r \leq 3$. This is encouraging, as these derivatives control the adaptation process for cubic elements. It is worth noting that the convergence rates obtained on meshes adapted to a strongly anisotropic field, test 2, do not differ drastically from the rates on uniform meshes, hinting similarities between a uniform mesh and a nonuniform mesh correctly adapted to the features of the solution.

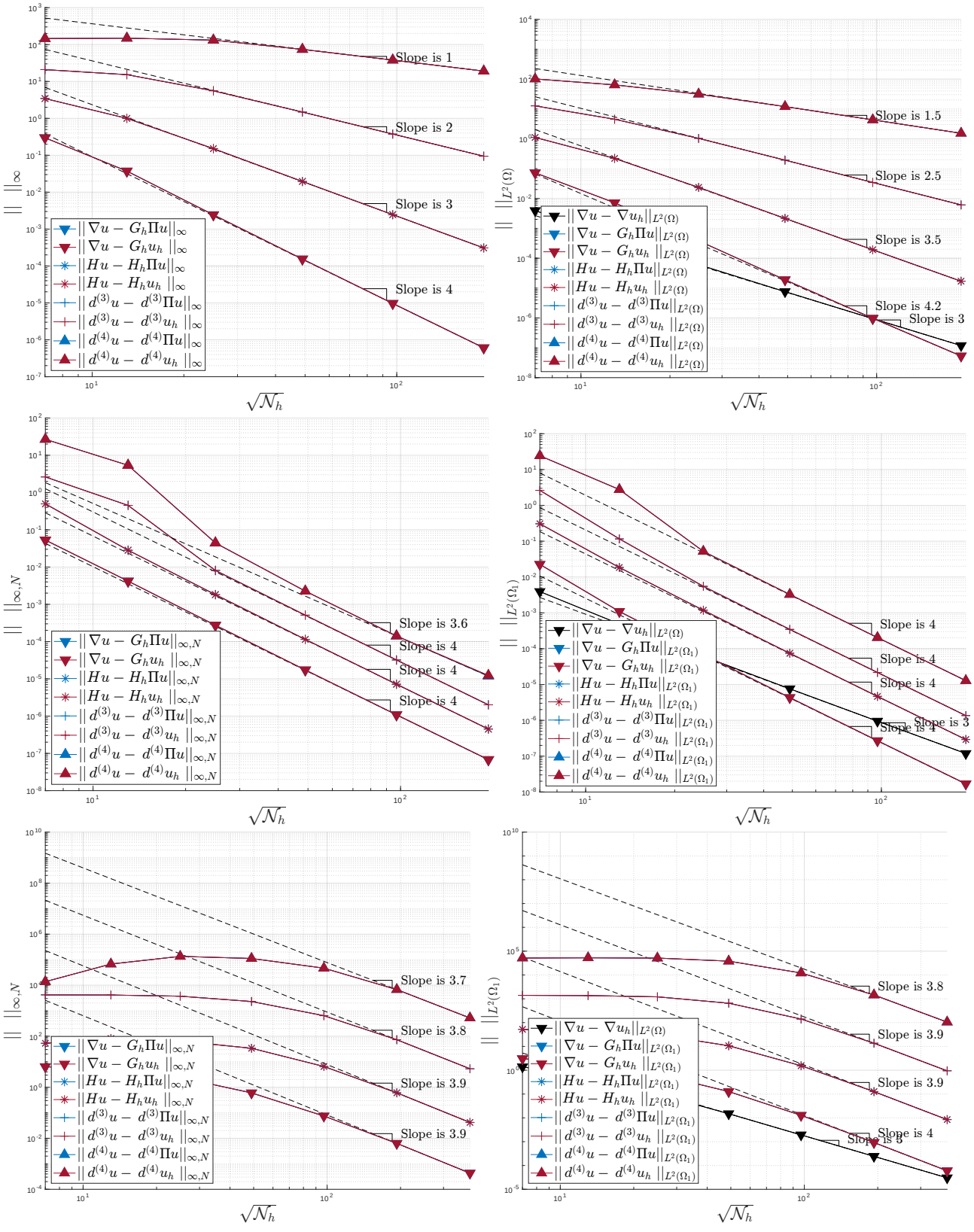


Figure B.4 Uniform meshes. Convergence results for the $(\mathcal{P}^1, \Pi^3)$ case for test 1 with (top) and without (middle) boundary, and for test 2 without boundary (bottom).

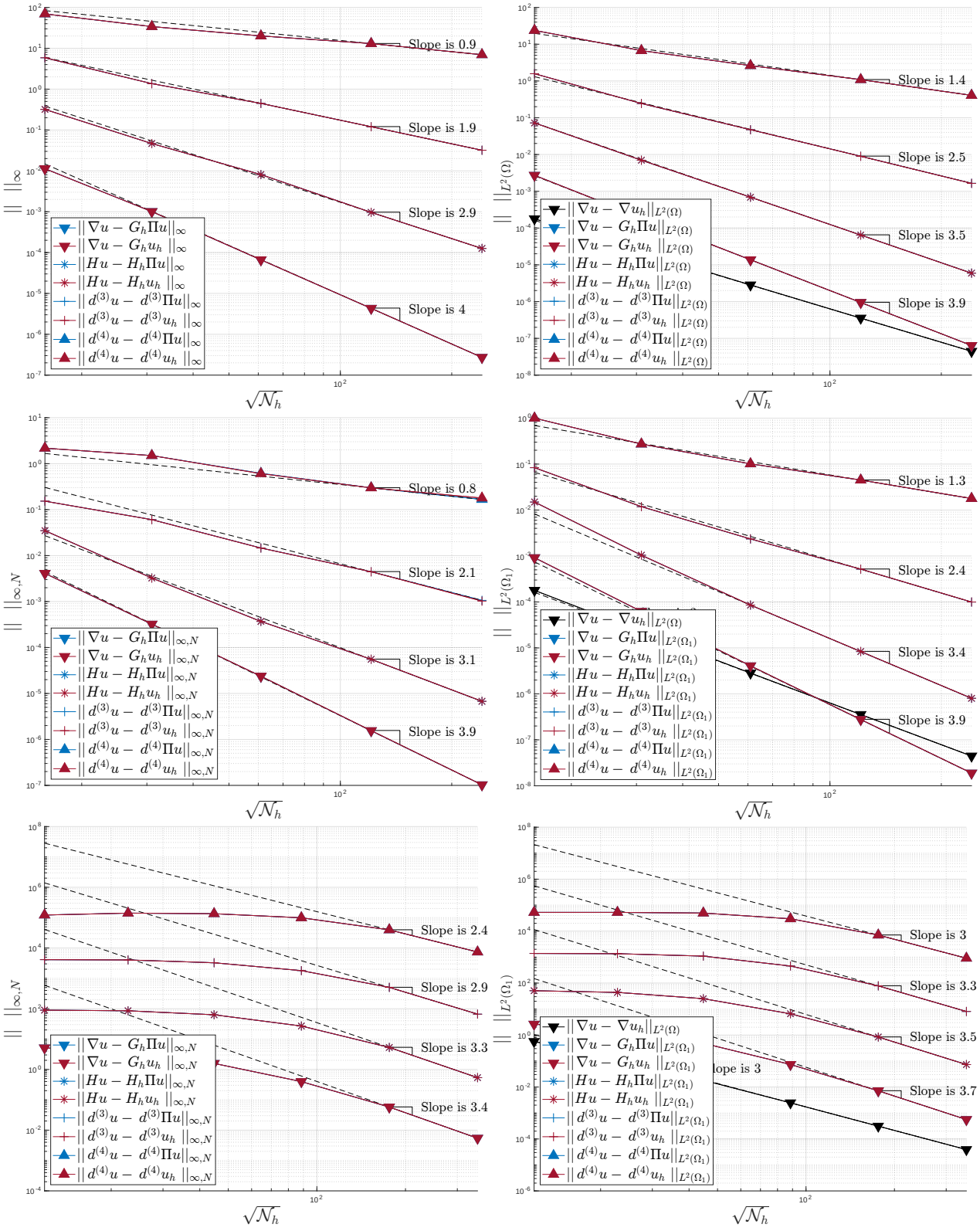


Figure B.5 Delaunay meshes. Convergence results for the $(\mathcal{P}^1, \Pi^3)$ case for test 1 with (top) and without (middle) boundary, and for test 2 without boundary (bottom).

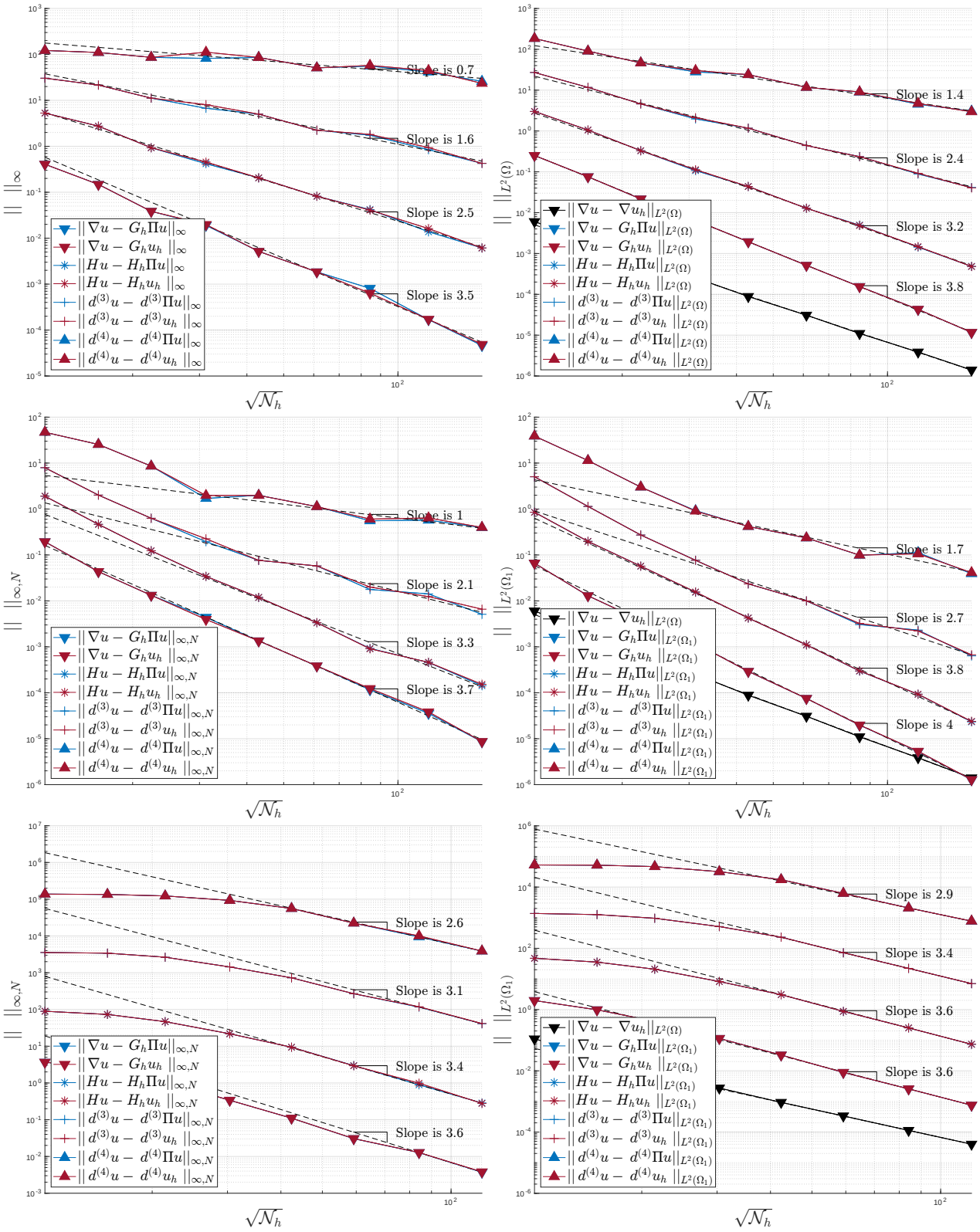


Figure B.6 Straight anisotropic meshes. Convergence results for the $(\mathcal{P}^1, \Pi^3)$ case for test 1 with and without boundary in L^2 norm only (top), and for test 2 with (middle) and without boundary (bottom).