HIGH-ORDER METRIC INTERPOLATION FOR CURVED *R*-ADAPTION BY DISTORTION MINIMIZATION

Guillermo Aparicio-Estrems[1] Abel Gargallo-Peiró[2] Xevi Roca[3]

[1] *Barcelona Supercomputing Center, 08034 Barcelona, Spain guillermo.aparicio@bsc.es*
[2] *Barcelona Supercomputing Center, 08034 Barcelona, Spain abel.gargallo@bsc.es*
[3] *Barcelona Supercomputing Center, 08034 Barcelona, Spain xevi.roca@bsc.es*

ABSTRACT

We detail how to use Newton's method for distortion-based curved r-adaption to a discrete high-order metric field. To this end, we consider three existent ingredients. First, a specific-purpose solver for distortion minimization. Second, a log-Euclidean high-order metric interpolation. Third, a point localization procedure for curved high-order meshes. We also extend to discrete metric fields a distortion-based curved r-adaption framework. To extend the framework, we provide, for the log-Euclidean high-order metric interpolation, the first and second derivatives in physical coordinates. These derivatives are required by Newton's method to solve the distortion minimization. The distortion minimization allows properly matching the anisotropic curved features of a discrete high-order metric. This matching capability might be relevant in global and cavity-based curved (straight-edged) high-order mesh adaption.

Keywords: Anisotropy, r-adaption, metric interpolation, curved high-order meshes

1. INTRODUCTION

The capability to relocate mesh nodes without changing the mesh topology, referred to as r-adaptivity, is a key ingredient in many adaptive PDE-based applications [1–3]. In these applications, to improve the solution accuracy, an error indicator or estimator determines the target stretching and alignment of the mesh. Then, to match these target features, an r-adaption procedure modifies the whole mesh (global) [4,5] or a previously remeshed cavity (local) [6–8].

In either case, r-adaptivity contributes to increasing the solution accuracy for a fixed number of degrees of freedom supported on a straight-edged mesh [3,4,6,9,10]. However, straight-edged meshes might not be an efficient support in many applications. Especially in applications where additional straight-edged mesh elements are artificially required to match highly curved solution features [11].

To efficiently match curved features, many practitioners have recently started to exploit curved high-order meshes. These meshes can be stretched and aligned in a pointwise varying fashion through anisotropic procedures [12], geodesic approaches for curved edges [13, 14], shock-tracking methods [15–17], and deformation analogies [18, 19]. Alternatively, the curved r-adaption can be driven, as for straight-edged elements [4, 5], by distortion measures. These measures are defined point-wise and are aware of either a target deformation matrix [20] or a target metric [21].

In adaptivity applications, the target deformations and metrics are not known a priori. These target fields are reconstructed a posteriori from the solution on the last mesh. Specifically, this mesh supports the resulting discrete representation of the target field. This discrete representation is key to interpolate the required field values in the adaptive procedure. Hence, to enable high-order adaptivity, we need the capability to interpolate target fields on a high-order mesh.

To match a deformation matrix, distortion optimization for curved r-adaption to a discrete target field is detailed in [20]. The method is really well-suited for simulation-driven r-adaption [22, 23]. It evaluates the distortion in a physical point by interpolating the target matrix on a discrete field. Although the derivatives of the target matrices are not zero, the method assumes they are zero. Moreover, the second derivatives are also assumed to be zero. Since non-null derivatives are assumed to be zero although the approach implements Newton's method, the curved *r*-adaption minimization corresponds to a quasi-Newton method.

To match a metric, distortion-based curved *r*-adaption to an analytic field can be performed with Newton's minimization [21, 24, 25]. The formulation for an analytic metric is derived in [21], while a specific-purpose globalization and a pre-conditioned Netwon-CG method are proposed in [24,25] to minimize the mesh distortion. Since the method deals with an analytic metric, it does not specify the derivatives for a metric represented by a discrete high-order field.

Regarding a discrete field representation, a convenient approach is to use a log-Euclidean [26] high-order metric interpolation [27]. This metric interpolation drives a cavity-based adaption approach, where the remeshed cavities are improved by locally smoothing the curved quadratic edges. To smooth these edges, the method optimizes the mid-node position. The optimization only uses the first derivatives of the log-Euclidean metric interpolation in terms of the curved edge coordinates. Accordingly, the method does not provide the first and second derivatives of the discrete metric field in physical coordinates.

Considering the previous open issues, our main contribution is to use Newton's optimization for distortion-based curved *r*-adaption to a discrete high-order metric field. We need three existent ingredients. First, to minimize the distortion, we use the specific-purpose solver in [24, 25]. Second, we represent the metric field as a log-Euclidean high-order metric interpolation [27] on a curved high-order mesh. Third, we locate physical points in the curved background mesh similar to the approach in [22]. We also need to extend to discrete metric fields a distortion-based curved *r*-adaption framework [21].

To extend the framework, the main novelty is to provide, for the log-Euclidean high-order metric interpolation, the first and second derivatives in physical coordinates. These derivatives are critical to use Newton's method for distortion minimization. This minimization leads to unprecedented second-order optimization results for curved *r*-adaption for a discrete high-order metric representation on a curved (or straight-edged) mesh.

The remainder of this paper is organized as follows. First, in Section 2 we introduce the metric-aware measures for curved high-order 2D elements. Next, in Section 3 we introduce the high-order log-Euclidean metric interpolation framework and we present the computation of its gradient and Hessian. Following, we present several examples to illustrate the capabilities of the proposed framework, Section 4. To finalize, in Section 5 we present the main conclusions and sum up the future work to develop.

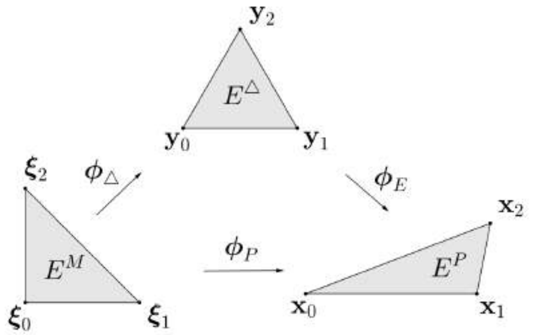


Figure 1: Mappings between the master, the ideal, and the physical elements in the linear case.

2. PRELIMINARIES: METRIC-AWARE MEASURES FOR CURVED HIGH-ORDER ELEMENTS

In this section, we review the definition of the Jacobian-based quality measure for high-order elements equipped with a metric, presented in [21]. To define and compute a Jacobian-based measure for simplices [5], three elements are required: the master, the ideal, and the physical, see Figure 1 for the linear triangle case. The master (E^M) is the element from which the iso-parametric mapping is defined. The equilateral element $\left(E^{\triangle}\right)$ represents the target configuration in the isotropic case. The physical (E^P) is the element to be measured.

To summarize the results in [21], we present the expression of the metric distortion measure in terms of the equilateral element $E^{\triangle}$. First, we need to compute a mapping from the master to the equilateral and physical elements, denoted as $\phi_{\triangle}$ and ϕ_P, respectively. By means of these mappings, we determine a mapping between the equilateral and physical elements by the composition

$$\phi_E : E^{\triangle} \xrightarrow{\phi_{\triangle}^{-1}} E^M \xrightarrow{\phi_P} E^P.$$

As detailed in [21], we define the point-wise distortion measure for a high-order element E^P equipped with a point-wise metric $\mathbf{M}$, at a point $\mathbf{y} \in E^{\triangle}$ as

$$\mathcal{N}\phi_E(\mathbf{y}) = \frac{\operatorname{tr}\left(\mathbf{D}\phi_E(\mathbf{y})^{\mathrm{T}} \cdot \mathbf{M}\left(\phi_E(\mathbf{y})\right) \cdot \mathbf{D}\phi_E(\mathbf{y})\right)}{d\left(\det\left(\mathbf{D}\phi_E(\mathbf{y})^{\mathrm{T}} \cdot \mathbf{M}\left(\phi_E(\mathbf{y})\right) \cdot \mathbf{D}\phi_E(\mathbf{y})\right)\right)^{1/d}}, \tag{1}$$

where the Jacobian of the map ϕ_E is given by

$$\mathbf{D}\phi_E(\mathbf{y}) := \mathbf{D}\phi_P(\phi_{\triangle}^{-1}(\mathbf{y})) \cdot \mathbf{D}\phi_{\triangle}^{-1}(\mathbf{y}).$$

Note that the distortion measure is independent of the computation of the metric $\mathbf{M}\left(\phi_E(\mathbf{y})\right)$, either using an analytical or a discretized representation.

We regularize the determinant in the denominator of Eq. (1) in order to detect inverted elements [28–31]. In particular, we define

$$\sigma_0 = \frac{1}{2}(\sigma + |\sigma|),$$

where

$$\sigma = \det(\mathbf{D}\phi_E(\mathbf{y}))\sqrt{\det(\mathbf{M}(\phi_E(\mathbf{y})))}.$$

Then, we define the point-wise regularized distortion measure of a physical element E^P at a point $\mathbf{y} \in E^{\triangle}$ as

$$\mathcal{N}_0\phi_E(\mathbf{y}) := \frac{\operatorname{tr}(\mathbf{D}\phi_E(\mathbf{y})^{\mathrm{T}} \cdot \mathbf{M}(\phi_E(\mathbf{y})) \cdot \mathbf{D}\phi_E(\mathbf{y}))}{d\sigma_0^{2/d}}, \qquad (2)$$

and its corresponding point-wise quality measure

$$\mathcal{Q}\phi_E(\mathbf{y}) = \frac{1}{\mathcal{N}_0\phi_E(\mathbf{y})}. \qquad (3)$$

Finally, we define the regularized elemental distortion by

$$\eta_{(E^P,\mathbf{M})} := \frac{\left(\int_{E^{\triangle}} (\mathcal{N}_0\phi_E(\mathbf{y}))^2 \, d\mathbf{y}\right)^{1/2}}{\left(\int_{E^{\triangle}} 1 \, d\mathbf{y}\right)^{1/2}} \qquad (4)$$

and its corresponding quality

$$\mathfrak{q}_{(E^P,\mathbf{M})} = \frac{1}{\eta_{0,(E^P,\mathbf{M})}}. \qquad (5)$$

We can improve the mesh configuration by means of relocating the nodes of the mesh according to a given distortion measure [21, 24, 25, 32]. In [21] it is proposed an optimization of the distortion (quality) of a mesh $\mathcal{M}$ equipped with a target metric $\mathbf{M}$ that describes the desired alignment and stretching of the mesh elements. To optimize a given mesh $\mathcal{M}$, first it is defined the mesh distortion by

$$\mathcal{F}(\mathcal{M}) := \sum_{E^P \in \mathcal{M}} \int_{E^{\triangle}} (\mathcal{N}_0\phi_E(\mathbf{y}))^2 \, d\mathbf{y},$$

which allows to pose the following global minimization problem

$$\mathcal{M}^* := \underset{\mathcal{M}}{\operatorname{argmin}} \mathcal{F}(\mathcal{M}), \qquad (6)$$

to improve the mesh configuration according to $\mathcal{F}$. In particular, herein, the degrees of freedom of the minimization problem in Eq. (6) correspond to the spatial coordinates of the mesh nodes.

To evaluate the distortion minimization formulation presented in Equation (6), an input metric is required. The reviewed r-adaption procedure has been applied for analytic metrics in [21]. In the following section, we detail the interpolation process that is required to extend the presented framework to dicrete metrics.

3. LOG-EUCLIDEAN METRIC INTERPOLATION

In this section, we formulate a metric interpolation process that allows both the distortion evaluation, Eq. (2), and its optimization, Eq. (6). In Sec. 3.1 we detail the log-Euclidean metric interpolation for linear and high-order elements first presented in [26] and [27,33], respectively. Then, in Sec. 3.2 we present, as a contribution of this work, the gradient and the Hessian of the log-Euclidean interpolation. Their computation will be used for the distortion minimization problem.

3.1 Metric Interpolation

In this section, we introduce the definition of the log-Euclidean metric interpolation at the background mesh. First, we introduce the required notation of the mappings and their parameters with the corresponding diagram. Secondly, we detail the interpolation procedure.

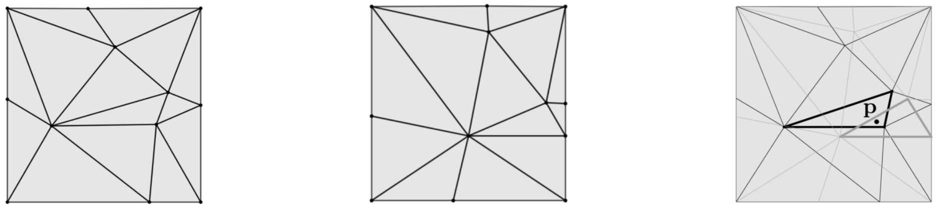
To evaluate the metric-aware distortion measure in Eq. (2) featuring discrete metrics, two meshes are required. On the one hand, the *physical* mesh $\mathcal{M}$, Figure 2(a), is the domain where the elements are deformed in order to solve the problem presented in Equation (6). On the other hand, the background mesh $\hat{\mathcal{M}}$, Figure 2(b), is a mesh that stores discrete metric values as a nodal field.

To evaluate the point-wise metric-aware distortion measure, we need to compute the interpolation of the point-wise metric values. For this, the localization between both meshes is required [22, 34, 35]. In particular, a physical point $\mathbf{p} \in \mathcal{M}$ is located at the background mesh $\hat{\mathcal{M}}$ where the metric is interpolated, see Figure 2(c). In what follows, we introduce the elements and the mappings required for this localization procedure.

We integrate the distortion measure presented in Equation (2) over the equilateral element via the master element E^M. In particular, for the metric evaluation, we map via ϕ_P, each integration point $\xi \in E^M$ to a point $\mathbf{p}$ of the physical element E^P, see Figure 3. To compute the metric at $\mathbf{p}$ we need to locate $\mathbf{p}$ in the background mesh, where the values of the metric are stored, see the intersection between E^P and the background element $E^{\hat{P}}$ in Figure 3. In addition, Figure 3 shows the procedure to obtain the coordinate to interpolate the metric from the quadrature points. In particular, we map a reference point $\xi \in E^M$ to a physical point $\mathbf{p} = \phi_P(\xi) \in E^P$, which we identify it with a point $\hat{\mathbf{p}} \in E^{\hat{P}}$ of the background mesh and its preimage is the background reference point $\hat{\xi} = \phi_{\hat{P}}{}^{-1}(\hat{\mathbf{p}}) \in E^{\hat{M}}$.

Given a physical point $\mathbf{p}$, we find it convenient to denote by ψ any mapping from a background element containing $\mathbf{p}$ that provides the coordinates in the background master element $E^{\hat{M}}$. Using this notation, we understand that any projection of a physical point $\mathbf{p}$ onto a point $\hat{\xi}$ of the background master element $E^{\hat{M}}$ corresponds to the evaluation of the non-linear function $\hat{\xi} = \psi(\mathbf{p})$.

To evaluate this non-linear function, we exploit that the expression of $\psi|_{E^P}$, defined in the intersection of a physical



(a) (b) (c)

Figure 2: Point localization: (a) physical mesh, (b) background mesh, and (c) a point **p** in the corresponding physical and background element (bold edges).

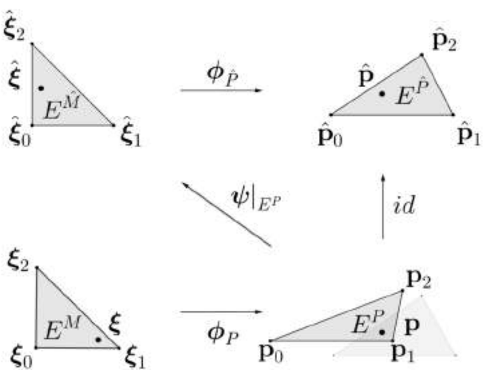


Figure 3: Mappings between the master and the physical elements (below) and their background analogs (above).

element E^P and a fixed background element $E^{\hat{P}}$, is given by

$$\begin{array}{rccc} \psi|_{E^P}: & E^P \cap E^{\hat{P}} & \to & E^{\hat{M}} \\ & \mathbf{p} & \mapsto & \phi_{\hat{P}}^{-1}(\mathbf{p}). \end{array} \qquad (7)$$

Specifically, we solve the non-linear inverse expression in the image term, Equation (7), by applying Newton's minimization to the squared distance. That is, we solve

$$\hat{\xi} = \operatorname*{argmin}_{\hat{\zeta}} \left|\left|\phi_{\hat{P}}\left(\hat{\zeta}\right) - \mathbf{p}\right|\right|^2.$$

The result is a numerical approximation of the point coordinates in the background master element. An alternative approach [22] is to seek the zeros of the vector equation

$$\phi_{\hat{P}}\left(\hat{\xi}\right) - \mathbf{p} = \mathbf{0}.$$

Once the background master coordinates associated to a given physical point have been computed, it is necessary to interpolate the metric supported by the background mesh at the corresponding master coordinate. To do so, we use the log-Euclidean interpolation proposed in [26,27]:

$$\mathbf{M}\left(\hat{\mathbf{N}}\right) := \exp\left(\mathbf{L}(\hat{\mathbf{N}})\right), \quad \mathbf{L}(\hat{\mathbf{N}}) := \sum_{j=1}^{\hat{n}} \hat{N}_j \log \mathbf{M}_j, \qquad (8)$$

where for the j-th node of the master element $E^{\hat{M}}$, $\mathbf{M}_j$ and $\hat{N}_j$ are the corresponding metric value and shape function, respectively. In addition, $\hat{\mathbf{N}}$ denotes all the shape functions, $\hat{n} = \frac{(\hat{p}+1)(\hat{p}+2)}{2}$ is the number of nodes and where $\hat{p}$ is the interpolation degree which corresponds to the polynomial degree of the master element $E^{\hat{M}}$. Finally, $\mathbf{M}(\hat{\mathbf{N}})$ is characterized by the *eigenvalue-based* matrix exponential function

$$\mathbf{M}\left(\hat{\mathbf{N}}\right) = \mathbf{U} \cdot \exp \mathbf{D} \cdot \mathbf{U}^{\mathrm{T}}, \qquad (9)$$

where $\mathbf{D}$, $\mathbf{U}$ are given from the eigenvalue decomposition of the matrix $\mathbf{L}(\hat{\mathbf{N}}) =: \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{U}^{\mathrm{T}}$. Finally, for each physical point $\mathbf{p}$ the metric interpolation is given by $\mathbf{M}\left(\hat{\mathbf{N}}\left(\psi\left(\mathbf{p}\right)\right)\right)$.

3.2 Gradient and Hessian

This section gives formulas of the gradient and Hessian of the metric interpolation over a background mesh in terms of the physical coordinates. For this, we detail first the case for the metric interpolation at a single element and then for the background mesh. In particular, our approach uses the gradient and Hessian of the eigenvalue decomposition presented in [36].

To compute the derivatives of the metric $\mathbf{M}$ we first differentiate the eigenvalue-based exponential matrix function presented in Equation (9) and then we differentiate the $\mathbf{L}$ function presented in Equation (8). By denoting x_j the coordinates of $\mathbf{p}$ and $\partial_j := \frac{\partial}{\partial x_j}$, $\partial_{jk} := \partial_j \partial_k = \frac{\partial}{\partial x_j}\frac{\partial}{\partial x_k}$ the partial derivatives in terms of the physical coordinates of $\mathbf{p}$, we can compute the spatial derivatives of the metric interpolation of Equation (8). In particular, the first-order derivatives are given by

$$\begin{aligned} \partial_j \mathbf{M}(\hat{\mathbf{N}}) = \partial_j \exp \mathbf{L}(\hat{\mathbf{N}}) = \partial_j \left(\mathbf{U} \cdot \exp \mathbf{D} \cdot \mathbf{U}^{\mathrm{T}}\right) = \\ \left(\partial_j \mathbf{U}\right) \cdot \exp \mathbf{D} \cdot \mathbf{U}^{\mathrm{T}} + \mathbf{U} \cdot \left(\partial_j \exp \mathbf{D}\right) \cdot \mathbf{U}^{\mathrm{T}} + \\ \mathbf{U} \cdot \exp \mathbf{D} \cdot \left(\partial_j \mathbf{U}^{\mathrm{T}}\right), \end{aligned}$$

and the second-order derivatives are given by

$$\begin{aligned}\partial_{jk}\mathbf{M}(\hat{\mathbf{N}}) = \partial_{jk}\exp\mathbf{L}(\hat{\mathbf{N}}) = \partial_{jk}\left(\mathbf{U}\cdot\exp\mathbf{D}\cdot\mathbf{U}^{\mathrm{T}}\right) = \\ \left(\partial_{jk}\mathbf{U}\right)\cdot\exp\mathbf{D}\cdot\mathbf{U}^{\mathrm{T}} + \partial_k\mathbf{U}\cdot\left(\partial_j\exp\mathbf{D}\right)\cdot\mathbf{U}^{\mathrm{T}} + \\ \partial_k\mathbf{U}\cdot\exp\mathbf{D}\cdot\left(\partial_j\mathbf{U}^{\mathrm{T}}\right) + \left(\partial_j\mathbf{U}\right)\cdot\partial_k\exp\mathbf{D}\cdot\mathbf{U}^{\mathrm{T}} + \\ \mathbf{U}\cdot\left(\partial_{jk}\exp\mathbf{D}\right)\cdot\mathbf{U}^{\mathrm{T}} + \mathbf{U}\cdot\partial_k\exp\mathbf{D}\cdot\left(\partial_j\mathbf{U}^{\mathrm{T}}\right) + \\ \left(\partial_j\mathbf{U}\right)\cdot\exp\mathbf{D}\cdot\partial_k\mathbf{U}^{\mathrm{T}} + \mathbf{U}\cdot\left(\partial_j\exp\mathbf{D}\right)\cdot\partial_k\mathbf{U}^{\mathrm{T}} + \\ \mathbf{U}\cdot\exp\mathbf{D}\cdot\left(\partial_{jk}\mathbf{U}^{\mathrm{T}}\right).\end{aligned}$$

Note that, since the matrix $\mathbf{D}$ is diagonal, we have

$$\begin{aligned}\partial_j\exp\mathbf{D} &= \exp\left(\mathbf{D}\right)\cdot\partial_j\mathbf{D}, \\ \partial_{jk}\exp\mathbf{D} &= \exp\left(\mathbf{D}\right)\cdot\left(\partial_k\mathbf{D}\cdot\partial_j\mathbf{D} + \partial_{jk}\mathbf{D}\right).\end{aligned}$$

The presented first and second-order derivatives of the metric require the first and second-order spatial derivatives of the eigenvalue decomposition (eigenvalues and eigenvectors), respectively. Their computation is appended in Section 7.

In addition, the derivatives of the eigenvalues and eigenvectors depend on the derivatives of the $\mathbf{L}$ function presented in Equation (8). In particular, they are given by

$$\nabla\mathbf{L} = \sum_j \left(\log\mathbf{M}_j\right)\nabla\hat{N}_j, \;\; \nabla^2\mathbf{L} = \sum_j \left(\log\mathbf{M}_j\right)\nabla^2\hat{N}_j,$$

where ∇ is the gradient with respect to physical coordinates. Therefore, to differentiate the metric interpolation $\mathbf{M}\left(\hat{\mathbf{N}}\left(\psi\left(\mathbf{p}\right)\right)\right)$ at a physical point $\mathbf{p}$, the derivatives of the map ψ presented in Equation (7) and of the shape functions $\hat{\mathbf{N}}$ are required.

The derivatives of $\psi|_{E^P}$ are given, at each patch $E^P\cap E^{\hat{P}}$, by the ones of the inverse of the physical map $\phi_{\hat{P}}^{-1}$ corresponding to the background mesh. To obtain the derivatives of the shape functions $\hat{\mathbf{N}}$ in terms of the physical coordinates $\mathbf{p}$, we consider the chain rule for the composition $\hat{\mathbf{N}}\circ\psi|_{E^P}$ and the restriction of the map $\psi|_{E^P}$ at each patch $E^P\cap E^{\hat{P}}$. We finally obtain the gradient

$$\nabla\hat{\mathbf{N}} = \nabla_{\hat{\xi}}\hat{\mathbf{N}}\cdot\nabla\phi_{\hat{P}}^{-1}, \tag{10}$$

where $\nabla_{\hat{\xi}}$ is the gradient with respect to $\hat{\xi}$ coordinates, and the Hessian

$$\nabla^2\hat{N}_j = \left(\nabla\phi_{\hat{P}}^{-1}\right)^{\mathrm{T}}\cdot\nabla_{\hat{\xi}}^2\hat{N}_j\cdot\nabla\phi_{\hat{P}}^{-1} + \nabla_{\hat{\xi}}\hat{N}_j\cdot\nabla^2\phi_{\hat{P}}^{-1}, \tag{11}$$

where

$$\begin{aligned}\nabla\phi_{\hat{P}}^{-1} &= \left(\nabla_{\hat{\xi}}\phi_{\hat{P}}\right)^{-1}, \\ \nabla^2\phi_{\hat{P}}^{-1} = \nabla\left(\left(\nabla_{\hat{\xi}}\phi_{\hat{P}}\right)^{-1}\right) &= -\nabla\phi_{\hat{P}}^{-1}\cdot\nabla_{\hat{\xi}}^2\phi_{\hat{P}}\cdot\nabla\phi_{\hat{P}}^{-1}.\end{aligned}$$

4. RESULTS

In this section, we present a 2D and a 3D example to illustrate the applicability of our distortion minimization framework for curved r-adaption to a high-order metric interpolation. First, we generate a background mesh $\hat{\mathcal{M}}$ and we evaluate the analytical metric $\mathbf{M}$ at the background mesh nodes. Second, we generate an initial physical mesh $\mathcal{M}$ and we measure its distortion (quality) by interpolating the metric. Finally, by relocating the nodes, we minimize the mesh distortion problem presented in Equation (6) using the framework presented herein.

To summarize the results, we present a table of the quality statistics, and the figures for the initial and optimized meshes, respectively. Specifically, we show the minimum quality, the maximum quality, the mean quality and the standard deviation of the initial and optimized meshes. We highlight that in all cases, the optimized mesh increases the minimum element quality and it does not include any inverted element. In addition, the meshes resulting after the optimization are composed of elements aligned and stretched to match the target metric tensor. In all figures, the meshes are colored according to the point-wise quality presented in Equation (3).

As a proof of concept, a mesh optimizer has been developed in Julia 1.4.2 [37] with the additional packages: Arpack v0.5.0, Einsum v0.4.1, EllipsisNotation v1.0.0, ILUZero v0.1.0, JLD v0.12.1, Plots v1.9.0, Setfield v0.7.0, SpecialFunctions v1.2.1, StatsBase v0.33.2, TensorOperations v3.1.0 and WriteVTK v1.8.0. In addition, we have used the MATLAB PDE Toolbox [38] to generate the initial isotropic linear unstructured 2D and 3D meshes (the structured meshes are generated by subdivision).

The Julia prototyping code is multithreaded, it corresponds to the implementation of the method presented in this work and the one presented in [21,24,25]. In all the examples, the optimization corresponds to finding a minimum of a nonlinear unconstrained multi-variable function. In particular, the mesh optimizer uses an unconstrained line-search globalization with an iterative preconditioned conjugate gradients linear solver. The stopping condition is set to reach an absolute root mean square residual, defined as $\frac{\|\nabla f(x)\|_{\ell^2}}{\sqrt{n}}$ for $x\in\mathrm{R}^n$, smaller than 10^{-4} or a length-step smaller than 10^{-4}. Each optimization process has been performed in a single node of a computing machine. Each node contains two Intel Xeon Platinum 8160 CPU with 24 cores, each at 2.10 GHz, and 96 GB of RAM memory.

We regularize the objective function to ensure infinite values for inverted configurations. Furthermore, to globalize the minimization, we equip Newton's method with a backtracking line-search. Whenever the Newton's update provides an inverted configuration, the objective function becomes infinity and thus, the backtracking line-search shortens the update until a valid configuration is reached.

Following, we first present the target domains to be meshed, and the considered metrics on the domain, Section 4.1. Next, in Section 4.2 we present the optimization results comparing both the proposed discrete based-interpolation procedure and the analytical one from [21, 24, 25]. Finally, in Sections 4.3 and 4.4, we show the application of the discrete metric approach to optimize an anisotropic mesh adapted to a given metric generated by the MMG algorithm presented in [39].

4.1 Domains and metrics

We consider the quadrilateral domain $\Omega = [-\frac{1}{2}, \frac{1}{2}]^2$ for the two-dimensional examples and the hexahedral domain $\Omega = [-\frac{1}{2}, \frac{1}{2}]^3$ for the three-dimensional ones. Each domain is equipped with a metric matching a boundary layer. In particular, our target metric $\mathbf{M}$ is characterized by a diagonal boundary layer metric $\mathbf{D}$ and a deformation map φ by the following expression

$$\mathbf{M} = \nabla\varphi^{\mathrm{T}} \cdot \mathbf{D} \cdot \nabla\varphi. \tag{12}$$

In what follows, we first detail the boundary layer metric $\mathbf{D}$ and then the deformation map φ.

The boundary layer aligns with the x-axis (xy-plane) in the 2D case (3D case). It determines a constant unit element size along the x-direction (xy-directions), and a non-constant element size along the y-direction (z-direction). This vertical element size grows linearly with the distance to the x-axis (xy-plane), with a factor $\gamma = 2$, and starts with the minimal value $h_{\min} = 0.1$. Thus, the stretching ratio blends from $1:10$ to $1:1$ between $y = -0.5$ and $y = 0.5$ (between $z = -0.5$ and $z = 0.5$). We define the metric for the 2D case as:

$$\mathbf{D} := \begin{pmatrix} 1 & 0 \\ 0 & 1/h(y)^2 \end{pmatrix} \tag{13}$$

where the function h is defined by

$$h(x) := h_{\min} + \gamma|x|.$$

Similarly, the metric for the 3D case is

$$\mathbf{D} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/h(z)^2 \end{pmatrix}. \tag{14}$$

The deformation map φ in Eq. (12) aligns the stretching of $\mathbf{D}$ according to a given curve in the 2D examples and at a given surface in the 3D examples. In the 2D case, we define the map φ by

$$\varphi(x,y) = \left(x, \frac{10y - \cos(2\pi x)}{\sqrt{100 + 4\pi^2}}\right),$$

and, in the 3D case by

$$\varphi(x,y,z) = \left(x, y, \frac{10z - \cos(2\pi x)\cos(2\pi y)}{\sqrt{100 + 8\pi^2}}\right).$$

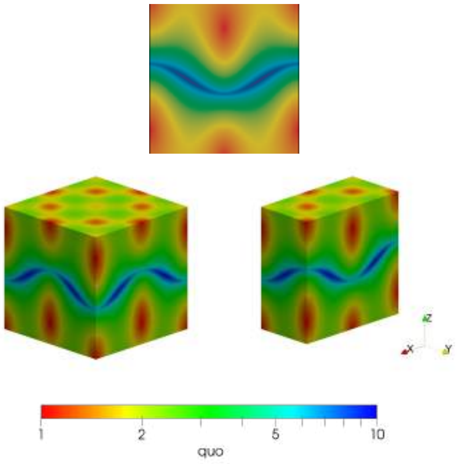


Figure 4: Anisotropic quotient values in logarithmic scale of the target metrics.

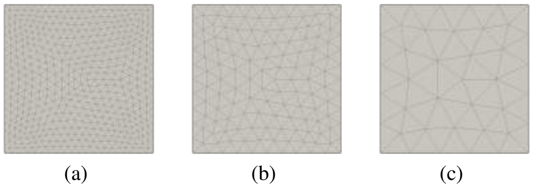

(a) (b) (c)

Figure 5: Background triangular meshes of polynomial degree 1, 2 and 4.

Figure 4 shows the anisotropic quotient [40] of the metric presented in Equations (13) and (14). Specifically, the anisotropic quotient of a metric tensor $\mathbf{M} \in \mathbb{R}^{d\times d}$ is given by

$$\mathrm{quo} = \max_{i=1,\dots,d} \sqrt{\frac{\det(\mathbf{M})}{\lambda_i^d}}$$

where $\lambda_i,\ i = 1, \dots, d$ are the eigenvalues of $\mathbf{M}$. The considered metric $\mathbf{M}$ attains the highest level of anisotropy, close to the curve described by the points $(x,y) \in \Omega$ such that $\varphi(x,y) = (x,0)$ in 2D, and close the surface described by the points $(x,y,z) \in \Omega$ such that $\varphi(x,y,z) = (x,y,0)$ in 3D.

4.2 Distortion minimization: initial isotropic straight-edged meshes

In this section, we present the optimization results for initially isotropic meshes on the domain equipped with the metrics presented in Section 4.1. We describe first the initial meshes $\mathcal{M}$ together with the background meshes $\hat{\mathcal{M}}$ where the metric is interpolated. Next, we present the optimized meshes $\mathcal{M}^*$ and to conclude, we present the results obtained from the optimization process. Herein, both the background and physical meshes are meshes of the same polynomial degree.

The initial meshes $\mathcal{M}$ are of polynomial degree 1, 2 and 4.

Table 1: Quality Statistics for the initial and optimized meshes with interpolated 2D metric.

Mesh	Minimum		Maximum		Mean		Standard deviation	
degree	Initial	Final	Initial	Final	Initial	Final	Initial	Final
1	0.2066	0.4481	0.9973	0.9853	0.6435	0.7558	0.2149	0.0890
2	0.2608	0.5609	0.9890	0.8647	0.6352	0.7706	0.2087	0.0708
4	0.3504	0.6834	0.9156	0.8268	0.6095	0.7661	0.1877	0.0450

Table 2: Quality Statistics for the initial and optimized meshes with analytical 2D metric.

Mesh	Minimum		Maximum		Mean		Standard deviation	
degree	Initial	Final	Initial	Final	Initial	Final	Initial	Final
1	0.2058	0.4510	0.9972	0.9846	0.6443	0.7556	0.2145	0.0823
2	0.2590	0.5648	0.9890	0.8734	0.6351	0.7703	0.2089	0.0715
4	0.3485	0.6838	0.9155	0.8417	0.6096	0.7735	0.1873	0.0530

Table 3: Quality Statistics for the initial and optimized meshes with interpolated 3D metric.

Mesh	Minimum		Maximum		Mean		Standard deviation	
degree	Initial	Final	Initial	Final	Initial	Final	Initial	Final
1	0.0875	0.2467	0.9841	0.9594	0.5636	0.6240	0.2199	0.1203
2	0.0980	0.4524	0.9810	0.9118	0.5739	0.6763	0.2214	0.0944
4	0.1929	0.5139	0.9228	0.8289	0.5847	0.7002	0.1998	0.0691

The three meshes feature the same number of nodes and they have the same resolution over the domain. In particular, in 2D the three initial meshes are composed of 481 nodes and 224, 56, and 14 elements, respectively. In 3D, they are composed of and 1577 nodes and 7296, 912, and 114 elements, respectively. In Figures 6 and 8 we show the initial meshes, they are colored according to the point-wise stretching and alignment quality measure, presented in Equation (3). Points in blue color have low quality and points with red color have high quality. As we observe, the elements lying in the region of highest stretching ratio have less quality than the elements lying in the isotropic region.

We equip each mesh with the metric presented in Equation (12). We obtain the metric values from the log-Euclidean interpolation method presented in Section 3. In particular, we interpolate the metrics from a background mesh $\hat{\mathcal{M}}$, see Figure 5 for the 2D cases. The background meshes are of polynomial degree 1, 2 and 4 according to the polynomial degree of the initial meshes $\mathcal{M}$. The three background meshes feature the same number of nodes and they have the same resolution over the domain. In particular, in 2D the three background meshes are composed of 521 nodes and 960, 240 and 60 elements, respectively. In 3D the they are composed of and 11411 nodes and 59456, 7432 and 929 elements, respectively.

To obtain an optimal configuration $\mathcal{M}^*$ we minimize the mesh distortion by relocating the mesh nodes while preserving their connectivity, as detailed in Section 2. The coordinates of the inner nodes, and the coordinates tangent to the boundary, are the design variables. Thus, the inner nodes are free to move, the vertex nodes are fixed, while the rest of boundary nodes are enforced to slide along the boundary facets of the domain Ω. The total amount of degrees of freedom for the 2D and 3D meshes is 222 and 3957, respectively. In Figure 6 we illustrate the optimized 2D meshes. In the 3D case, Figure 8 shows the interior and exterior of the meshes. We align the axes according to the ones of Figure 4. We observe that the elements lying in the anisotropic region are compressed to attain the stretching and alignment prescribed by the metric.

Tables 1 and 3 show the quality statistics of both the initial and optimized meshes for the 2D and 3D cases, respectively. In all the optimized meshes the minimum is improved and the standard deviation of the element qualities is reduced when compared with the initial configuration.

To validate the proposed method, we compare 2D curved *r*-adaption results for the high-order metric interpolation with the results corresponding to an analytic metric evaluation. Considering the initial meshes presented in this section, we optimize the distortion measure by evaluating the analyti-

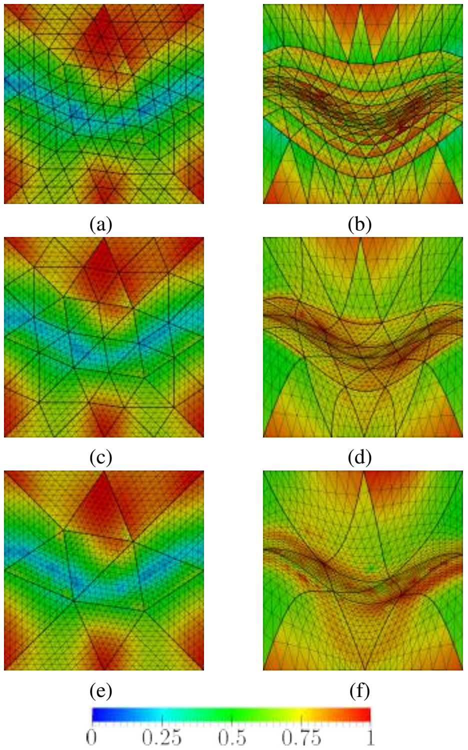


Figure 6: Point-wise distortion for triangular meshes of polynomial degree 1, 2 and 4 in rows. Initial straight-sided isotropic meshes and optimized meshes from initial meshes in columns. The interpolation of the metric has been used for the distortion minimization. The sub-triangular elements are the visualization elements. These element vertices are not the high-order degrees of freedom.

cal metric expression, instead of interpolating it in the background mesh. In Figure 7 we show the initial and optimized meshes. They colored according the point-wise quality measure of Equation (3) using the analytical metric expression.

To compare quantitatively both results, we compute the relative distance of the node coordinates of the optimized configurations. The relative distance is around 10^{-2} for all the tested cases, obtaining comparable nodal configurations, as it can be observed when comparing Figures 6 and 7.

In Table 2 we present the quality statistics of the initial and optimized meshes using the analytical metric evaluation. To compare the quality improvement of both approaches, we compute the difference between the analyzed quality statistics, obtaining a value for all the statistics below 10^{-2}. Thus, the quality improvement driven by the optimization using the proposed metric interpolation procedure is analogous to the one given by the analytical metric, obtaining in all cases high-quality configurations with a minimum quality over 0.4.

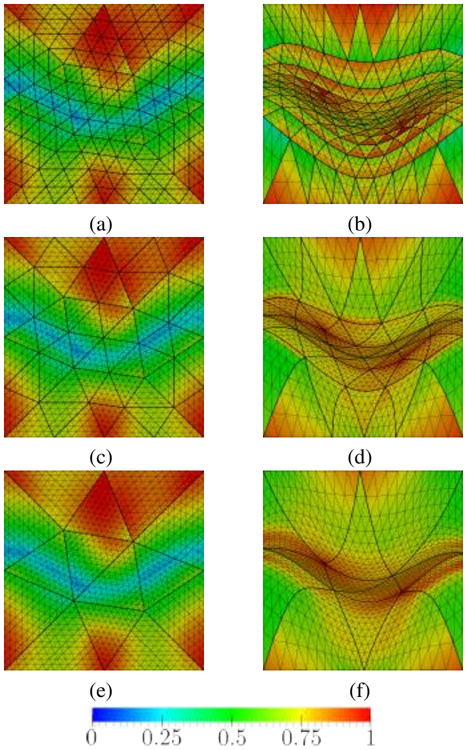


Figure 7: Point-wise distortion for triangular meshes of polynomial degree 1, 2 and 4 in rows. Initial straight-sided isotropic meshes and optimized meshes from initial meshes in columns. The analytic evaluation of the metric has been used for the distortion minimization.

4.3 Distortion minimization: initial anisotropic straight-edged meshes

The results presented in Section 4.2 show the application of the metric interpolation procedure to optimize isotropic meshes in a domain equipped with a metric. However, in practice, anisotropic meshes are generated combining topological mesh operations that modify the mesh connectivity and mesh r-adaption procedures [6]. To illustrate a practical example, we consider an initial straight-sided mesh adapted by the MMG algorithm presented in [39]. Then, we apply the anisotropic r-adaption method presented in this work.

First, we consider the target metric presented in Equation (12) with $h_{\min} = 0.01$. Second, we generate a linear isotropic triangular mesh of input size $h_{\min}/2 = 0.005$ with MATLAB. Then, we couple such mesh with the target metric evaluated at the mesh vertices and normalized according to different sizes. These sizes are chosen in order to obtain a comparable mesh resolution according to the mesh polynomial degree. Specifically, they are given by 0.0625, 0.125 and 0.25 for the linear, quadratic and quartic case, respectively. We apply the MMG algorithm to obtain a straight-sided anisotropic mesh of polynomial degree 1, 2 and 4, see Figure 9. In particular, they are composed by 1161 nodes and 2137 triangles,

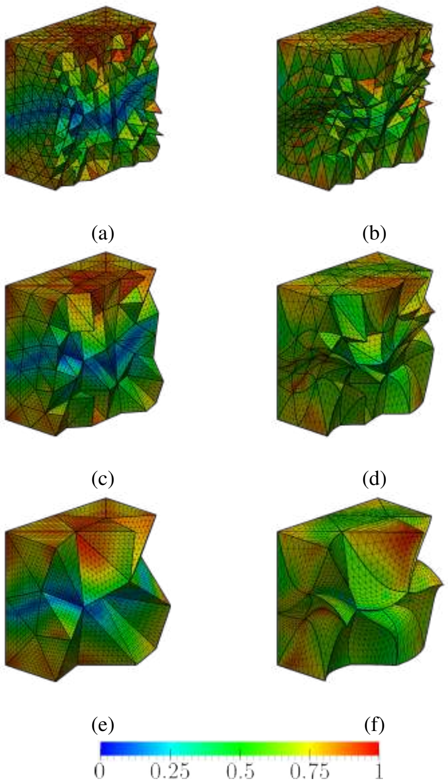

(a) (b) (c) (d) (e) (f)

Figure 8: Tetrahedral meshes of polynomial degree 1, 2 and 4 in rows. Initial straight-sided isotropic meshes and optimized meshes from initial meshes in columns.

1333 nodes and 624 triangles and, 1525 nodes and 180 triangles, respectively.

The generated meshes are then optimized using the metric interpolation approach presented in this work. In Figure 9 we illustrate the optimized meshes. We observe that the elements lying in the anisotropic region are compressed to attain the stretching and alignment prescribed by the metric. In Table 4 we show the quality statistics of both the initial and optimized meshes. In all the optimized meshes the minimum is improved and the standard deviation of the element qualities is reduced when compared with the initial configuration. We conclude that, with the same metric data and hence, the same inputs, the r-adaption mesh post-processing improves the quality of the meshes generated with the MMG algorithm. In addition, for the straight-edged case, we have presented a global method to improve the stretching and alignment prescribed by the metric after applying an h-adaption approach.

4.4 Distortion minimization: curved boundaries

We following illustrate that our approach is compatible with curved boundaries. To this end, we consider the holed domain $\Omega = \frac{1}{2}[-1,1]^2 \backslash C$ where C is the circle with radius equal to $\frac{3}{16}$ and centered at the origin. The domain Ω has two boundaries, the one of the square $\frac{1}{2}[-1,1]^2$ and the one of the circle C. We equip it with the target metric presented in Equation (12) with $h_{\min} = 0.01$. Then, we generate with MMG a linear isotropic triangular mesh of input size $h_{\min}/2 = 0.005$ over $\frac{1}{2}[-1,1]^2$. As before, we couple such mesh with the target metric evaluated at the mesh vertices and normalized according to size $h = 0.2$. Finally, we apply the MMG algorithm to obtain a straight-sided anisotropic mesh of polynomial degree 2 composed by 672 nodes and 290 triangles, see Figures 10 and 11.

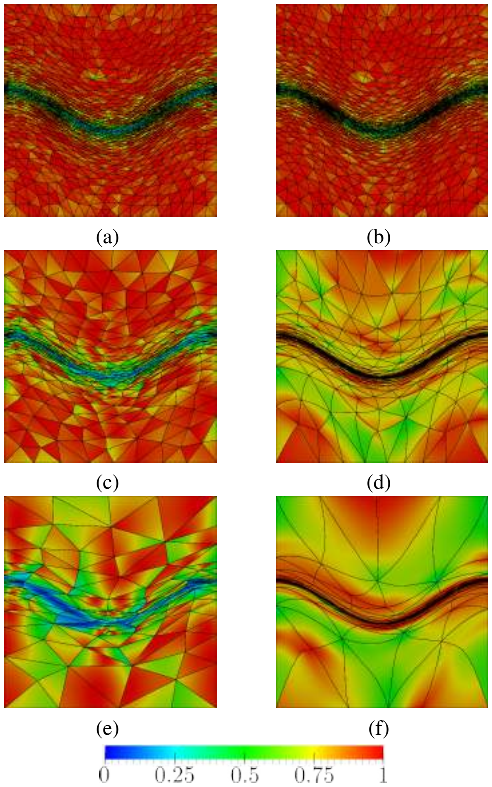

(a) (b) (c) (d) (e) (f)

Figure 9: Point-wise distortion for triangular meshes of polynomial degree 1, 2 and 4 in rows. Initial straight-sided anisotropic meshes and optimized meshes from initial meshes in columns.

To accommodate the curved boundaries we include, to the presented functional, a boundary term that takes into account the mesh approximation to the boundaries of the domain (both the square and the circle). In addition, to approximate the metric stretching, we optimize the mesh using the metric interpolation approach presented in this work. Finally, when optimizing the mesh functional all mesh nodes coordinates are free that is, each mesh node moves in $\mathbb{R}^2$.

In Figures 10 and 11 we illustrate the optimized mesh. We observe that the elements lying in the anisotropic region are

Table 4: Quality Statistics for the initial MMG and optimized meshes with interpolated 2D metric.

Mesh	Minimum		Maximum		Mean		Standard deviation	
degree	Initial	Final	Initial	Final	Initial	Final	Initial	Final
1	0.0365	0.1794	0.9988	0.9989	0.7806	0.7961	0.2273	0.2040
2	0.0624	0.6300	0.9982	0.9913	0.6966	0.8692	0.2558	0.0788
4	0.0424	0.6063	0.9774	0.9965	0.5677	0.9137	0.2681	0.0886

Table 5: Quality Statistics for the initial MMG and optimized mesh with interpolated 2D metric over the holed domain.

Mesh	Minimum	Maximum	Mean	Standard deviation
Initial	0.0489	0.9877	0.6058	0.2512
Optimized	0.3042	0.9927	0.7397	0.1821

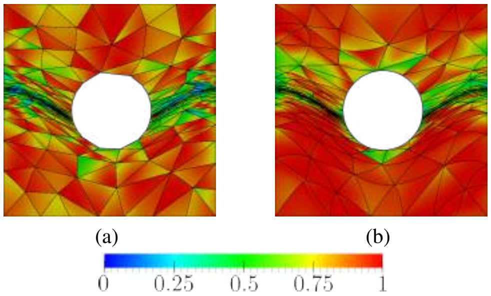


Figure 10: Point-wise distortion for triangular meshes of polynomial degree 2. Initial straight-sided anisotropic mesh (a) and optimized mesh (b).

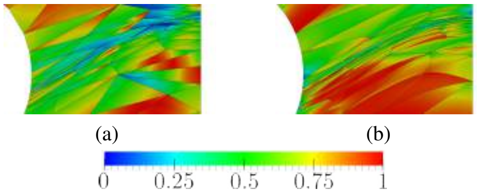


Figure 11: Zoom of the right region for the initial (a) and optimized mesh (b).

compressed to attain the stretching and alignment prescribed by the metric. Note that the boundary elements are curved to match both the metric and the curved domain boundaries. In Table 5 we show the quality statistics of both the initial and optimized mesh. In the optimized mesh the minimum is improved and the standard deviation of the element qualities is reduced when compared with the initial configuration.

5. CONCLUDING REMARKS

In conclusion, we have obtained unique results in curved r-adaption to a discrete high-order metric. We have represented the discrete metric in a curved background mesh as a high-order log-Euclidean metric interpolation. For this metric interpolation, we have detailed the first and second derivatives in terms of the physical coordinates. These derivatives have allowed minimizing with Newton's method a mesh distortion accounting for the discrete high-order metric. The discrete metric results compare well with the analytic metric results. In both cases, the method exploits the non-constant Jacobian of curved high-order elements to match curved anisotropic features properly.

In perspective, this capability to match curved anisotropic features might be an attractive ingredient for curved high-order goal-oriented or indicator-based adaption. In these adaptive processes, one would have a high-order metric field in the current curved mesh. This background field would drive curved r-adaption to globally (locally) relocate the current curved mesh (re-meshed cavity) according to the curved anisotropic features of the solution.

6. ACKNOWLEDGEMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 715546. This work has also received funding from the Generalitat de Catalunya under grant number 2017 SGR 1731. The work of X. Roca has been partially supported by the Spanish Ministerio de Economía y Competitividad under the personal grant agreement RYC-2015-01633.

7. APPENDIX: DERIVATIVES OF THE EIGENVALUE DECOMPOSITION

In this Appendix, we detail the first and second-order spatial derivatives of the eigenvalue decomposition (eigenvalues and eigenvectors), first presented in [36] and rewritten herein using our notation.

Let us consider, for $\ell = 1,...,d$, the eigenvalue equation for the eigenvector $\mathbf{u}_\ell$ with eigenvalue λ_ℓ

$$\mathbf{L}_\ell \mathbf{u}_\ell := (\mathbf{L} - \lambda_\ell \mathbf{I})\,\mathbf{u}_\ell = 0,$$

where $\mathbf{L}$ is a symmetric matrix and $\mathbf{I}$ is the identity matrix. Then, by taking its first-order and second-order derivatives we respectively obtain

$$0 = \partial_j (\mathbf{L}_\ell \mathbf{u}_\ell) = \left(\partial_j \mathbf{L}_\ell\right) \cdot \mathbf{u}_\ell + \mathbf{L}_\ell \cdot \partial_j \mathbf{u}_\ell, \tag{15}$$

$$\begin{aligned} 0 = \partial_{jk} (\mathbf{L}_\ell \mathbf{u}_\ell) = \left(\partial_{jk} \mathbf{L}_\ell\right) \cdot \mathbf{u}_\ell + \mathbf{L}_\ell \cdot \partial_{jk} \mathbf{u}_\ell + \\ \left(\partial_j \mathbf{L}_\ell\right) \cdot \partial_k \mathbf{u}_\ell + \left(\partial_k \mathbf{L}_\ell\right) \cdot \partial_j \mathbf{u}_\ell. \end{aligned} \tag{16}$$

For each ℓ one first computes the first-order derivative of the eigenvalue λ_ℓ by left-multiplying by $\mathbf{u}_\ell$ to Equation (15). Then, by solving the remaining unknown term of Equation (15) one obtains the first-order derivatives of the eigenvector $\mathbf{u}_\ell$. In particular, the first-order derivatives of the eigenvalues and the eigenvectors are given by

$$\partial_j \lambda_\ell = \mathbf{u}_\ell^{\mathrm{T}} \cdot \partial_j \mathbf{L} \cdot \mathbf{u}_\ell, \;\; \partial_j \mathbf{u}_\ell = -\mathbf{L}_\ell^{+} \cdot \partial_j \mathbf{L}_\ell \cdot \mathbf{u}_\ell,$$

where the operation $\mathbf{L}_\ell^{+}$ is the Moore-Penrose pseudo-inverse matrix for the matrix $\mathbf{L}_\ell$. We use the Moore-Penrose pseudo-inverse matrix instead of the inverse matrix because the matrix $\mathbf{L}_\ell$ is singular. In addition, the redundant equations are satisfied automatically.

The second-order derivatives are obtained by applying a similar procedure. For each ℓ one first computes the second-order derivative of the eigenvalue λ_ℓ by left-multiplying by $\mathbf{u}_\ell$ to Equation (16). Then, by solving the remaining unknown term of Equation (16) one obtains the second-order derivatives of the eigenvector $\mathbf{u}_\ell$. In particular, the second-order derivatives of the eigenvalues are given by

$$\partial_{jk} \lambda_\ell = \mathbf{u}_\ell^{\mathrm{T}} \cdot \left(\partial_k \mathbf{L}_\ell \cdot \partial_j \mathbf{u}_\ell + \partial_j \mathbf{L}_\ell \cdot \partial_k \mathbf{u}_\ell + \partial_{jk} \mathbf{L} \cdot \mathbf{u}_\ell\right),$$

$$\begin{aligned} \partial_{jk} \mathbf{u}_\ell = -\mathbf{L}_\ell^{+} \cdot \left(\partial_k \mathbf{L}_\ell \cdot \partial_j \mathbf{u}_\ell + \partial_j \mathbf{L}_\ell \cdot \partial_k \mathbf{u}_\ell + \partial_{jk} \mathbf{L}_\ell \cdot \mathbf{u}_\ell\right) - \\ \left(\partial_j \mathbf{u}_\ell \cdot \partial_k \mathbf{u}_\ell\right) \mathbf{u}_\ell, \end{aligned}$$

where the last term of the second-order derivative of the eigenvector is obtained by imposing the second-order derivative of the imposed normalization condition $\mathbf{u}_\ell^{\mathrm{T}} \cdot \mathbf{u}_\ell = 1$

$$0 = \partial_{jk} \left(\mathbf{u}_\ell^{\mathrm{T}} \cdot \mathbf{u}_\ell\right) = 2\partial_{jk} \mathbf{u}_\ell^{\mathrm{T}} \cdot \mathbf{u}_\ell + 2\partial_j \mathbf{u}_\ell^{\mathrm{T}} \cdot \partial_k \mathbf{u}_\ell$$

Note that, for each differentiation order, the computation of the eigenvectors derivatives requires the values of the eigenvalues derivatives.

References

[1] Yano M., Darmofal D.L. "An optimization-based framework for anisotropic simplex mesh adaptation." *Journal of Computational Physics*, vol. 231, no. 22, 7626–7649, 2012

[2] Loseille A., Alauzet F. "Continuous mesh framework part I: well-posed continuous interpolation error." *SIAM Journal on Numerical Analysis*, vol. 49, no. 1, 38–60, 2011

[3] Coupez T., Silva L., Hachem E. "Implicit boundary and adaptive anisotropic meshing." *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, pp. 1–18. Springer, 2015

[4] Huang W., Russell R.D. *Adaptive Moving Mesh Methods*, vol. 174 of *Applied Mathematical Sciences*. Springer, 2011

[5] Knupp P.M. "Algebraic mesh quality metrics." *SIAM J. Numer. Anal.*, vol. 23, no. 1, 193–218, 2001

[6] Alauzet F., Loseille A. "A Decade of Progress on Anisotropic Mesh Adaptation for Computational Fluid Dynamics." *Computer-Aided Design, Elsevier*, vol. 72, no. 1, 13–39, 2016

[7] Gruau C., Coupez T. "3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric." *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 48-49, 4951–4976, 2005

[8] Frey P., Alauzet F. "Anisotropic mesh adaptation for CFD computations." *Computer methods in applied mechanics and engineering*, vol. 194, no. 48-49, 5068–5082, 2005

[9] Hecht F. "BAMG: bidimensional anisotropic mesh generator." *User Guide. INRIA, Rocquencourt*, 1998

[10] Coupez T. "Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing." *Journal of computational physics*, vol. 230, no. 7, 2391–2405, 2011

[11] Fidkowski K.J., Darmofal D.L. "Review of output-based error estimation and mesh adaptation in computational fluid dynamics." *AIAA journal*, vol. 49, no. 4, 673–694, 2011

[12] Coupez T. "On a Basis Framework for High Order Anisotropic Mesh Adaptation." vol. 203, 141–153, 2017. Research Note 26th International Meshing Roundtable

[13] Johnen A., Geuzaine C., Toulorge T., Remacle J.F. "Quality Measures for Curvilinear Finite Elements." *TILDA: Towards Industrial LES/DNS in Aeronautics*, p. 221, 2021

[14] Zhang R., Johnen A., Remacle J.F. "Curvilinear mesh adaptation." *International Meshing Roundtable*, pp. 57–69. Springer, 2018

[15] Zahr M.J., Persson P.O. "An optimization based discontinuous Galerkin approach for high-order accurate shock tracking." *2018 AIAA Aerospace Sciences Meeting*, p. 0063. 2018

[16] Zahr M.J., Shi A., Persson P.O. "Implicit shock tracking using an optimization-based high-order discontinuous Galerkin method." *Journal of Computational Physics*, vol. 410, 109385, 2020

[17] Zahr M.J., Persson P.O. "An r-adaptive, high-order discontinuous Galerkin method for flows with attached shocks." *AIAA Scitech 2020 Forum*, p. 0537. 2020

[18] Marcon J., Turner M., Moxey D., Sherwin S.J., Peiró J. "A variational approach to high-order r-adaptation." *IMR26*, 2017

[19] Marcon J., Castiglioni G., Moxey D., Sherwin S.J., Peiró J. "rp-adaptation for compressible flows." *International Journal for Numerical Methods in Engineering*, vol. 121, no. 23, 5405–5425, 2020

[20] Dobrev V., Knupp P., Kolev T., Mittal K., Tomov V. "The target-matrix optimization paradigm for high-order meshes." *SIAM Journal on Scientific Computing*, vol. 41, no. 1, B50–B68, 2019

[21] Aparicio-Estrems G., Gargallo-Peiró A., Roca X. "Defining a Stretching and Alignment Aware Quality Measure for Linear and Curved 2D Meshes." *International Meshing Roundtable*, pp. 37–55. Springer, 2018

[22] Dobrev V., Knupp P., Kolev T., Tomov V. "Towards simulation-driven optimization of high-order meshes by the Target-Matrix Optimization Paradigm." *International Meshing Roundtable*, pp. 285–302. Springer, 2018

[23] Dobrev V., Knupp P., Kolev T., Mittal K., Rieben R., Tomov V. "Simulation-driven optimization of high-order meshes in ALE hydrodynamics." *Computers & Fluids*, vol. 208, 104602, 2020

[24] Aparicio-Estrems G., Gargallo-Peiró A., Roca X. "Anisotropic Optimization of curved meshes: specific-purpose line-search and trust-region globalizations for Newton's method." *International Meshing Roundtable*. 2019

[25] Aparicio-Estrems G., Gargallo-Peiró A., Roca X. "Stretching and aligning piece-wise polynomial meshes to match curved anisotropic features." *International Conference on Spectral and High-Order Methods*. 2021

[26] Arsigny V., Fillard P., Pennec X., Ayache N. "Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors." *Magnetic Resonance in Medicine*, vol. 56, 411–421, 2006

[27] Rochery L., Loseille A. "P2 cavity operator and Riemannian curved edge length optimization: a path to high-order mesh adaptation." *AIAA Scitech 2021 Forum*, p. 1781. 2021

[28] Branets L.V., Garanzha V.A. "Distortion measure of trilinear mapping. Application to 3-D grid generation." *Numerical linear algebra with applications*, vol. 9, no. 6-7, 511–526, 2002

[29] López E.J., Nigro N.M., Storti M.A. "Simultaneous untangling and smoothing of moving grids." *Int. J. Numer. Meth. Eng.*, vol. 76, no. 7, 994–1019, 2008

[30] Escobar J.M., Rodríguez E., Montenegro R., Montero G., González-Yuste J.M. "Simultaneous untangling and smoothing of tetrahedral meshes." *Comput. Meth. Appl. Mech. Eng.*, vol. 192, no. 25, 2775–2787, 2003

[31] Gargallo-Peiró A., Roca X., Peraire J., Sarrate J. "Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes." *Int. J. Numer. Meth. Eng.*, vol. 103, 342–363, 2015

[32] Gargallo-Peiró A. *Validation and generation of curved meshes for high-order unstructured methods*. Ph.D. thesis, Universitat Politècnica de Catalunya, 2014

[33] Ekelschot D., Ceze M., Murman S.M., Garai A. "Parallel high-order anisotropic meshing using discrete metric tensors." *AIAA Scitech 2019 Forum*, p. 1993. 2019

[34] Mittal K., Dutta S., Fischer P. "Nonconforming Schwarz-spectral element methods for incompressible flow." *Computers & Fluids*, vol. 191, 104237, 2019

[35] Sitaraman J., Floros M., Wissink A., Potsdam M. "Parallel domain connectivity algorithm for unsteady flow computations using overlapping and adaptive grids." *Journal of Computational Physics*, vol. 229, no. 12, 4703–4723, 2010

[36] Andrew A.L., Chu K.W.E., Lancaster P. "Derivatives of eigenvalues and eigenvectors of matrix functions." *SIAM journal on matrix analysis and applications*, vol. 14, no. 4, 903–926, 1993

[37] Bezanson J., Edelman A., Karpinski S., Shah V.B. "Julia: A fresh approach to numerical computing." *SIAM review*, vol. 59, no. 1, 65–98, 2017

[38] MATLAB. *version 9.3.0.713579 (R2017b)*. The MathWorks Inc., Natick, Massachusetts, 2017

[39] Dobrzynski C. *MMG3D: User guide*. Ph.D. thesis, INRIA, 2012

[40] Loseille A., Löhner R. "Anisotropic Adaptive Simulations in Aerodynamics." *AIAA 2010-169*, 2010. 49th AIAA Aerospace Sciences Meeting, Fairfax, Va, USA