



HAL
open science

Well-suited and adaptive post-processing for the visualization of hp simulation results

Matthieu Maunoury, Christophe Besse, Vincent Mouysset, Sébastien Pernet,
Pol-André Haas

► **To cite this version:**

Matthieu Maunoury, Christophe Besse, Vincent Mouysset, Sébastien Pernet, Pol-André Haas. Well-suited and adaptive post-processing for the visualization of hp simulation results. *Journal of Computational Physics*, 2018, 375, pp.1179-1204. 10.1016/j.jcp.2018.09.017 . hal-01743380

HAL Id: hal-01743380

<https://hal.science/hal-01743380>

Submitted on 26 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Well-suited and adaptive post-processing for the visualization of hp simulation results

Matthieu Maunoury^a, Christophe Besse^b, Vincent Mouysset^a, Sébastien Pernet^a, Pol-André Haas^a

^aONERA / DTIS, Université de Toulouse, Toulouse - France

^bInstitut de Mathématiques de Toulouse; UMR 5219, Université de Toulouse; CNRS, UPS IMT, F-31062 Toulouse cedex 9, France

Abstract

While high order methods became very popular as they allow to perform very accurate solutions with low computational time and memory cost, there is a lack of tools to visualize and post-treat the solutions given by these methods. Originally, visualization softwares were developed to post-process results from methods such that finite differences or usual finite elements and therefore process linear primitives. In this paper, we present a methodology to visualize results of high order methods. Our approach is based on the construction of an optimized affine approximation of the high order solution which can therefore be handled by any visualization software. A representation mesh is constructed and the process is guided by an a posteriori estimate which control the error between the numerical solution and its representation pointwise. This point by point control is crucial as under their picture form, data correspond to values mapped on elements where anyone can pick up a pointwise information. A strategy is established to ensure that discontinuities are well represented. These discontinuities come either from the physical problem (material change) or the numerical method (discontinuous Galerkin method) and are pictured accurately. Several numerical examples are presented to demonstrate the potential of the method.

Keywords: Visualization, high-order finite element, discontinuous Galerkin, post-processing

1. Introduction

High order methods represent an important class of numerical methods to solve partial differential equations (PDE). Among these methods, one can cite the Finite Element method [1] and the Discontinuous Galerkin (DG) method [2]. Many domains are concerned by these methods: for solving wave equations (in acoustic, elastodynamic or electromagnetism [3]), in Computational Fluid Dynamic [4] or for Solid and Structural Mechanics. High order methods are built from high order basis functions: high degree of the element (for example curved element) and/or high polynomial degree. These methods became very popular as they allow to perform very accurate solutions with low computational time and memory cost [3]. In isogeometric analysis [5], functions used to describe the geometry of the computational domain are directly employed. Thus, high order functions are widely used [6]. The analysis and development of these high order methods is still an active area of research and is widely used in the industry.

However, while many works have been done on the development of the solvers, there is a lack of tools to visualize and post-treat the high order methods. Historically, PDE were solved with low order method (finite difference, low order finite element). Thus, visualization of linear functions defined on simple elements (e.g. triangles, tetrahedra, quadrilaterals and hexahedra) is common and well-suited. Originally, the standard visualization software were not thought for high order methods. To the best of our knowledge, two main strategies are used to bypass this limitation: pixel-exact rendering and low order remeshing. The first strategy is the creation of a visualization software which will be adapted to high order methods by using a pixel-exact rendering [7, 8]. An example of a high order visualization software is the Element Visualizer (ElVis, [9]). Such a software is based on pixel exact rendering, the costs to interactively manipulate the data are therefore significantly higher than using linear approximations [9]. As the rendering is constructed pointwise, there is no element vision and thus, an additional and expensive work is necessary to post-process the solution. The second strategy is to transform the data given by the hp method in a combination of linear functions defined on simple elements which will be handled by any visualization software. The basic idea is to define a representation mesh and introduce a refinement strategy. A visualization error is also introduced due

to the P^1 approximation of high order functions. Remacle et al. [10] use an isotropic refinement which can result in the generation of a large amount of data. They evaluate the error measures (the approximation of the geometry and the field representation) with a L^2 -norm. The post-processing of the data can be done directly in the visualization software [11]. Note that Schroeder et al. [12] propose a first method based on the subdivision of element edges for some finite element methods but this latter seems to be not optimal in terms of *a posteriori* indicators and doesn't take into account the specificities of DG methods for example.

In this paper, we present a methodology to visualize the solution f_{num} of a *hp* method. We propose to construct an approximation of this solution, noted f_{vis} , which can be handled accurately by most visualization softwares. All the usual post-processing tools can therefore be at our disposal. Our approach is therefore to convert the data into a combination of affine functions with an auto-adaptive method based on a L^∞ guidance used as refinement method. It allows to ensure a good pointwise accuracy and a low memory cost as it focuses only where needed. Our methodology allows to overcome the weakness of the linear strategy (accuracy and memory cost). We define three objectives that f_{vis} must respect to be "well-suited":

(O_1): The representation of f_{num} is obtained by plotting piecewise affine functions on k -simplexes, where k is the (local) dimension of the (local) support of f_{num} ,

(O_2): According to a prescribed tolerance $\varepsilon > 0$, error between f_{num} and f_{vis} is lower than ε in L^∞ -norm,

(O_3): The representation f_{vis} is (locally) continuous if f_{num} is (locally) continuous.

Objective (O_1) represents a compatibility aim on the whole process, including the use of plotting tools. Indeed, piecewise affine functions on k -simplexes can be represented without loss of accuracy with any visualization software. Thus, if f_{vis} verifies (O_1), one is ensured to keep accurateness between its construction and representation. The notion of k -simplex is introduced to say that, for instance, f_{num} stands for a 2D surface in \mathbb{R}^3 and we are going to mesh it using triangles. In the same way, the locality of this notion indicates that f_{num} can be made of elements with various dimensions (e.g. 2D and 3D elements for hybrid Boundary Integral Equation / Finite Element Method computations).

Actually, objective (O_2) is composed of two parts. First, we ask to control the error between f_{vis} and f_{num} . Then, we force the measure to be completed in L^∞ -norm. Motivation of the first sub-objective is quite natural as we are going to approximate f_{num} and thus generate errors. The second one is linked to the specific use of data represented. Indeed, under its picture form, the data corresponds to values mapped on elements where anyone can pick up a pointwise information. Hence, we ask to use the natural norm to control accuracy on pointwise values: the L^∞ -norm.

Objective (O_3) ensures that f_{vis} does not introduce gaps where f_{num} is continuous. Indeed, if not, it will give a false information in the final representation. This has to be avoided. The addition of objectives (O_2) and (O_3) can be expressed as: at the prescribed tolerance, the representation shows gaps if and only if f_{num} has. Capturing gaps of f_{num} should be of prime interest for discontinuous schemes (such as DG: Discontinuous Galerkin). Indeed, gaps can either be physically relevant or coming from numerical errors. In the first case, showing them enhances the quality of the scheme. In the other case, gaps of f_{num} should be related to *a posteriori* estimates of the numerical error and thus qualify the accuracy of the scheme.

In this paper, only surfaces are treated (i.e. the local dimension is $k = 2$). The main result of this paper is the following theorem and the approach given in its sketch of proof.

Theorem 1.1. *Let ε be a prescribed tolerance. If f_{num} is a suitable numerical solution (as described in (H_1) to (H_5) hereafter) and a function f_{vis} is constructed following algorithm 1 given in section 4, then this function f_{vis} verifies objectives (O_1)-(O_2)-(O_3).*

Sketch of proof. The algorithm presented is based on a description of f_{num} from a transformation between a reference cell \widehat{K} and a mesh cell K (as described in (H_1) to (H_5)). This formalism takes into account a wide class of *hp* numerical schemes. Thus, f_{vis} is a piecewise affine function supported on a mesh of K which is the linear mapping of a mesh of \widehat{K} made of simplexes. Then, this f_{vis} satisfies (O_1) by construction. The mesh of \widehat{K} is performed using usual adaptive techniques with a specific *a posteriori* estimate which is an upper-bound of Hausdorff distance, based on L^∞ -norm, between the graphs of f_{num} and f_{vis} . Hence, (O_2) is verified. Now, to fulfil (O_3), the mesh of any cell K is done from a mesh of its boundary ∂K , where ∂K was decomposed in edges being meshed separately. Each edge mesh is built adaptively using an *a posteriori* estimate involving the values of f_{num} coming from all cells sharing this edge.

However, the weak point of this strategy is that to satisfy (O_3) , during the mesh phase of K , we forbid to modify edge meshes, then for fine enough elements close to ∂K , the linear mapping used to define the mesh of K happens to be not injective. The consequence is that we can achieve a mesh such that our *a posteriori* estimate tends to the prescribed tolerance ε whereas it doesn't control Hausdorff distance as the latter is no longer defined. As a consequence, (O_2) is not satisfied. To bypass this problem, we define a specific *a posteriori* estimate involved when meshing edges which introduces an *a priori* control of the 2D mesh process in the vicinity of ∂K . ■

The structure of the paper is as follows. In section 2, we start by defining what a well-suited visualization should be. Section 3 is devoted to the construction of a well-suited visualization: a pseudo algorithm is given and *a posteriori* estimates are defined to lead the construction in an adaptive manner. Section 4 contains the algorithm and explanation on how the algorithm is performed. Section 5 deals with the convergence of the methodology presented in this paper. Section 6 contains numerical results. Finally, conclusions and future works are given in the last section.

2. Definition of a well-suited visualization

In this section, we begin by describing the *hp* framework. Then, we give details of the construction of a well-desired visualization. In the latter part of this section, a quantification of the visualization error is introduced.

2.1. Definition of the *hp* framework

In all the sequel, we consider that the original problem was to compute the approximation of a function f defined as a mapping from a subset $X = \mathbb{R}^d$ ($d = 1, 2$ or 3) of a given space \mathbb{X} into a \mathbb{K} -vector space \mathbb{Y} ($\mathbb{K} = \mathbb{R}$ or \mathbb{C}). These spaces are supposed to be of Banach type and are thus equipped with norms respectively noted $\|\cdot\|_{\mathbb{X}}$ and $\|\cdot\|_{\mathbb{Y}}$. The numerical approximation of f , noted f_{num} , is computed with help of a discretization of X . This mesh of X is designed by $\mathcal{T}(X)$ and is supposed to verify the following properties:

- (H_1) : $\mathcal{T}(X)$ is a conformal meshing of X , that is $\cup_{K \in \mathcal{T}(X)} K = X$, made of bounded closed cells (K),
- (H_2) : there is no overlapping between elements, that is $Int(K) \cap Int(K') = \emptyset$, for any couple $(K, K') \in \mathcal{T}(X)^2$, $K \neq K'$, where $Int(K)$ stands for the interior of K ,
- (H_3) : each cell $K \in \mathcal{T}(X)$ is defined as the image of a given reference cell \widehat{K} , expressed in a reference space $\widehat{\mathbb{X}}_K$, by a bijective transformation g_K . The dimension of $\widehat{\mathbb{X}}_K$, being the dimension of K in \mathbb{X} , is noted n_K .

These hypotheses are motivated by the following observations: hypothesis (H_1) is not crucial but is introduced in a sake of simplicity (if $\cup_{K \in \mathcal{T}(X)} K \neq X$, then replace in all the sequel X by the set $\cup_{K \in \mathcal{T}(X)} K$). Hypothesis (H_2) should allow non-conform meshes in the finite-element sense. Hypothesis (H_3) is introduced here in order to grant a wide enough range of applications for the numerical schemes to be addressed. This property is linked to the description of basis functions explained hereafter. Actually, we will be able to treat schemes with basis functions constructed in X or defined on a reference cell (\widehat{K}) and obtained by mean of a transformation (g_K). In the first case, one simply has to choose the trivial identification $\widehat{K} = K$, $\widehat{\mathbb{X}}_K = \mathbb{X}$ and $g_K = Id$. The second one becomes very interesting especially when X is a subset of \mathbb{X} with smaller dimension (for instance a curved surface in \mathbb{R}^3). In that case, a natural choice of $\widehat{\mathbb{X}}_K$ is a space of same dimension (here \mathbb{R}^2) giving a more natural coordinate system to define the basis functions. Note at last that the notation of the reference cell (\widehat{K}) is linked to the one of the cell (K) so different local descriptions can be adopted in the same set X . This last point allows, for instance, to deal with hybridized schemes.

The basis functions are defined on the reference cells. We note them $\{\varphi_i^K\}_{i=1, N_K}$ with respect to $K \in \mathcal{T}(X)$ and we assume that:

- (H_4) : on any $K \in \mathcal{T}(X)$, $\{\varphi_i^K\}_{i=1, N_K} \in C^0(\widehat{K}, \mathbb{Y})$, where $C^0(\widehat{K}, \mathbb{Y})$ is the set of continuous functions over \widehat{K} ,
- (H_5) : on any $K \in \mathcal{T}(X)$, the definition of f_{num} on K , noted f_{num}^K , is expressed with given coefficients (the degrees of freedom) $f_i^K \in \mathbb{K}$ by

$$\forall x \in K, f_{num}^K(x) = \sum_{i=1}^{N_K} f_i^K \varphi_i^K(g_K^{-1}(x)).$$

Regularity introduced with hypothesis (H_4) is not really restrictive as most of hp methods are built on polynomial basis. Basis functions can be scalar or vectorial depending on \mathbb{Y} . Hypothesis (H_5) describes the usual construction of a numerical approximation from a basis where functions are mappings from reference cells. The notations about the numerical function are plotted on figure 1.

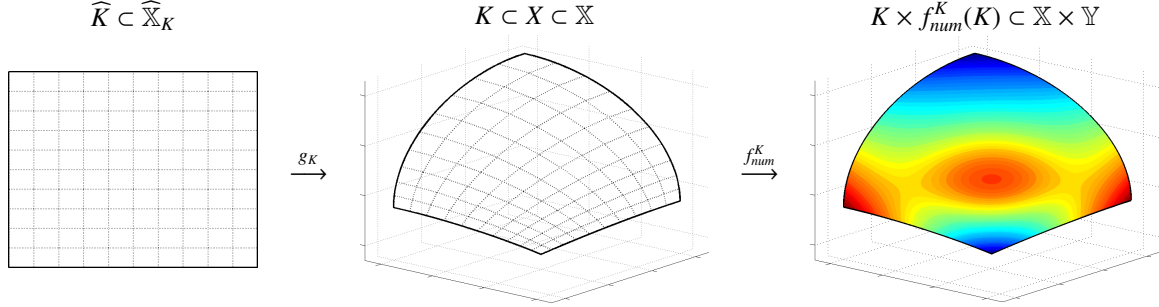


Figure 1: Description of the notations used.

Over the domain X , f_{num} can be explicitated as:

$$\forall x \in X, f_{num}(\{x\}) = \{f_{num}^K(x) / \forall K \in \mathcal{T}(X), x \in K\}. \quad (1)$$

Note that f_{num} is not a function on X . Indeed, the definition of f_{num} given by (1) does not ensure that a $x \in X$ is associated to at most one value. This comes from (H_2) which allows that two cells have all or part of their boundary in common. This definition can thus handle with the representation of discontinuous functions such as those obtained in DG schemes. However, from (H_2) to (H_5), for any $K \in \mathcal{T}(X)$, f_{num}^K is a well-defined and continuous function when restricted to the interior of K . Moreover, this function admits a straightforward continuous extension to K (which is closed by (H_1)).

2.2. Construction of desired visualization and notations

According to the definition of f_{num} given previously, the values of $f_{num}(\{x\})$ are determined by the cell we consider that x belongs to. Indeed, for any $x \in X$, from (H_1) there is at least one cell $K \in \mathcal{T}(X)$ containing x . There are two possibilities: x belongs to the interior of K and then by (H_2) this cell is unique which involves $f_{num}(\{x\}) = \{f_{num}^K(x)\}$. Otherwise, x lays on the boundary of K which can be common to several cells. In this case, f_{num} can present multiple values (up to one per cell containing x). Our aim is to represent accurately f_{num} and to show gaps at the boundary of cells if they exist (objective (O_3)). Hence, following (1), we are going to define f_{vis} through the local expressions $(f_{num}^K)_{K \in \mathcal{T}(X)}$ of f_{num} by constructing local visualizations noted $(f_{vis}^{\bar{K}})_{K \in \mathcal{T}(X)}$. The exponent \bar{K} denotes that the support of this function can differ from K (as we will observe in the sequel).

To ensure a maximum of compatibility between $f_{vis}^{\bar{K}}$ and visualization softwares we are looking for a linear function expressed on simplexes (objective (O_1)). Hence, a meshing of K , noted $\mathcal{T}(K)$, will be constructed and $f_{vis}^{\bar{K}}$ will satisfy:

$$\forall S \in \mathcal{T}(K), f_{vis}^{\bar{K}}|_S \in \mathbb{P}^1(S), \quad (2)$$

where $\mathbb{P}^1(S)$ stands for the space of polynomials with total order lower or equal to 1. Note that K has not been supposed to be divisible into n_K -simplexes and hence it can happen that

$$\bar{K} := \bigcup_{S \in \mathcal{T}(K)} S \neq K. \quad (3)$$

Typically, this can occur when the geometrical transformation g_K is not affine (for instance with curved elements). The union of all these meshed elements is $\bar{X} := \bigcup_{K \in \mathcal{T}(X)} \bar{K}$ and is *a priori* neither included in nor containing X . In the same way, it can also happen that

$$f_{vis}^{\bar{K}}(\bar{K}) \neq f_{num}^K(K). \quad (4)$$

According to the definition of the basis functions $(\varphi_i^K)_i$ which are functions on \widehat{K} with values in \mathbb{Y} (see (H_4)), and according to the link between \widehat{K} and K with g_K (see (H_3)), the meshing of K will thus be performed through a meshing of \widehat{K} . This last mesh, noted $\mathcal{T}(\widehat{K})$, is composed of simplexes and is constructed in the following way. Each element \widehat{S} of $\mathcal{T}(\widehat{K})$ is a n_K -simplex (i.e. the convex hull of $n_K + 1$ points $\widehat{\mathcal{N}} := (\widehat{n}_i)_{i=1, \dots, n_K+1}$). We define Λ which associates $n + 1$ points $\mathcal{P} = (P_i)_{i=1, \dots, n+1}$ to the n -simplex $\Lambda(\mathcal{P})$ and the topology of $\mathcal{T}(\widehat{K})$ by $\widehat{\Theta}_K := \{\widehat{\mathcal{N}} := (\widehat{n}_i)_{i=1, \dots, n_K+1} : \Lambda(\widehat{\mathcal{N}}) \in \mathcal{T}(\widehat{K})\}$. Then, the mesh of \widehat{K} is defined by $\mathcal{T}(\widehat{K}) := \{\Lambda(\widehat{\mathcal{N}}) : \widehat{\mathcal{N}} \in \widehat{\Theta}_K\}$. The construction of $\mathcal{T}(K)$ is performed in the same way. We define the set of nodes of $\mathcal{T}(K)$ as well as its topology by $\Theta_K := \{\mathcal{N} = (N_i)_{i=1, \dots, n_K+1} := g_K(\widehat{\mathcal{N}}) : \widehat{\mathcal{N}} \in \widehat{\Theta}_K\}$ with $g_K(\widehat{\mathcal{N}}) := (g_K(\widehat{n}_i))_{i=1, \dots, n_K+1}$ and $\mathcal{N} := (\widehat{n}_i)_{i=1, \dots, n_K+1}$. The mesh in simplexes of K is defined by $\mathcal{T}(K) := \{\Lambda(\mathcal{N}) : \mathcal{N} \in \Theta_K\}$.

The method proposed is based on a classic plane to surface transformation from \widehat{K} to \widetilde{K} (see [13, 14]). As the L^∞ norm is not monotonic (Hausdorff distance can increase after refinement of the mesh), the advancing front techniques ([15]) can not be used. Furthermore, as two quantities are approximated (the geometry and the numerical function), there is no easy way to use metric methods [16, 17, 18].

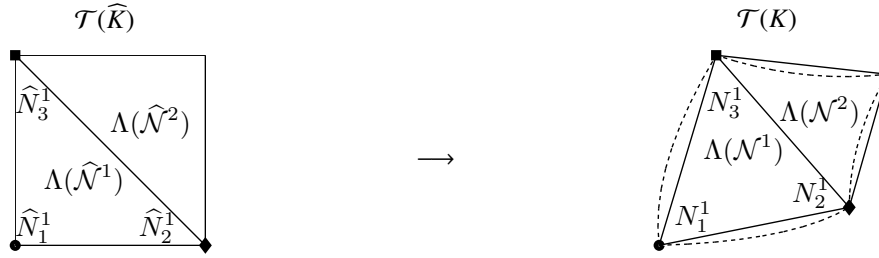


Figure 2: Construction of $\mathcal{T}(\widehat{K})$ and $\mathcal{T}(K)$ for an element $K \in \mathcal{T}(X)$ (in dotted line).

An example of construction of these two meshes $\mathcal{T}(\widehat{K})$ and $\mathcal{T}(K)$ is shown in figure 2. The link between the meshes $\mathcal{T}(\widehat{K})$ and $\mathcal{T}(K)$ is given by the following lemma. A last hypothesis is needed for this lemma:

(H_6) : the reference cells \widehat{K} are polyhedrons. Then, there exists $\mathcal{T}(\widehat{K})$ which satisfies $\widehat{K} = \cup_{\widehat{S} \in \mathcal{T}(\widehat{K})} \widehat{S}$ such that for any couple $(S, S') \in \mathcal{T}(\widehat{K})^2$, $S \neq S'$, $Int(S) \cap Int(S') = \emptyset$.

Lemma 2.1 (Identification of $\mathcal{T}(K)$). For all $K \in \mathcal{T}(X)$, under (H_3) and (H_6) , the \mathbb{P}^1 interpolation of g_K constructed from $\mathcal{T}(\widehat{K})$, noted $P^1 g_{\mathcal{T}(K)}$, is

1. a bijective function between $\mathcal{T}(\widehat{K})$ and $\mathcal{T}(K)$,
2. a bijective function between the sets $\widehat{\Theta}_K$ and Θ_K ,
3. a surjective function from \widehat{K} onto \widetilde{K} .

Proof. Let $\Lambda(g_K(\widehat{\mathcal{N}}))$, $\widehat{\mathcal{N}} \in \widehat{\Theta}_K$ be a n_K -simplex of $\mathcal{T}(K)$. By definition, the \mathbb{P}^1 -approximation of g_K verifies $P^1 g_{\mathcal{T}(K)}(\widehat{n}_i) = g_K(\widehat{n}_i) = N_i$ for any node N_i of $\Lambda(g_K(\widehat{\mathcal{N}}))$. Hence, $P^1 g_{\mathcal{T}(K)}(\Lambda(\widehat{\mathcal{N}})) = P^1 g_{\mathcal{T}(K)}(\{\sum_{i=1}^{n_K+1} x_i \widehat{n}_i, (x_i) \in [0, 1]^{n_K+1}\}) = \{\sum_{i=1}^{n_K+1} x_i N_i, (x_i) \in [0, 1]^{n_K+1}\} = \Lambda(g_K(\widehat{\mathcal{N}}))$. The surjectivity of $P^1 g_{\mathcal{T}(K)}$ from $\mathcal{T}(\widehat{K})$ to $\mathcal{T}(K)$ comes by construction as $\mathcal{T}(K) = \{\Lambda(g_K(\widehat{\mathcal{N}})), \widehat{\mathcal{N}} \in \widehat{\Theta}_K\}$. Injectivity is proven by noting that $\mathcal{T}(\widehat{K})$ and $\mathcal{T}(K)$ have the same cardinality. It proves the first result. The second result stems from the bijectivity of g_K and by noticing that $P^1 g_{\mathcal{T}(K)} = g_K$ on $\widehat{\mathcal{N}}$. The third result comes from the surjectivity of the first one and the hypothesis (H_6) : $\widehat{K} = \cup_{\widehat{\mathcal{N}} \in \widehat{\Theta}_K} \Lambda(\widehat{\mathcal{N}})$. ■

The function $f_{vis}^{\widehat{K}}$ is then constructed with help of these simplexes and the definition of functions $f_{vis}^{\Lambda(\mathcal{N})}$ on each simplex $\Lambda(\mathcal{N})$ of $\mathcal{T}(K)$. $f_{vis}^{\Lambda(\mathcal{N})}$ is an affine function on $\Lambda(\mathcal{N})$ defined by $f_{vis}^{\Lambda(\mathcal{N})}(x) = \sum_{i=1}^{n_K+1} x_i f_{num}^K(N_i) \in \mathbb{Y}$, where $x = \sum_{i=1}^{n_K+1} x_i N_i \in \Lambda(\mathcal{N}) \subset \widetilde{K}$, $(N_i)_{i=1, \dots, n_K+1}$ are the nodes of $\Lambda(\mathcal{N})$ and $(x_i)_{i=1, \dots, n_K+1} \in [0, 1]^{n_K+1}$ is unique as $\Lambda(\mathcal{N})$ is

the closed hull of $\{N_i : i = 1, \dots, n_K\}$. Note that for each node $N_i \in \mathcal{N}$, $f_{vis}^{\Lambda(N)}(N_i) := f_{num}^K(N_i)$. Then, $f_{vis}^{\bar{K}}$ is given by:

$$\forall x \in \bar{K}, f_{vis}^{\bar{K}}(\{x\}) = \left\{ f_{vis}^{\Lambda(N)}(x) / \forall N \in \Theta_K, x \in \Lambda(N) \right\}. \quad (5)$$

Figure 3 gives an illustration of both notations and the construction process.

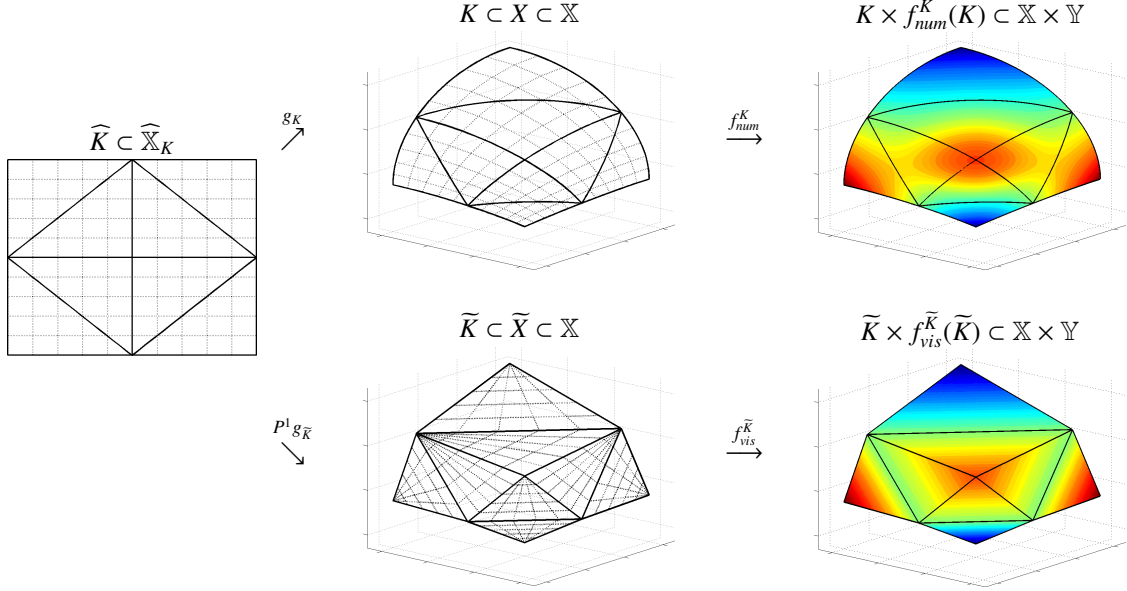


Figure 3: Description of the notations used.

Remark 2.2. Unfortunately, lemma 2.1 does not ensure the bijectivity from \widehat{K} onto \bar{K} . Indeed, the definition of $\mathcal{T}(K)$ does not forbid the creation of k -simplexes with a non-empty intersection of their interior. The consequence of this loss of bijectivity is that $f_{vis}^{\bar{K}}$ constructed following (5) is not a function on \bar{K} (see the following proposition).

Proposition 2.3. Let $K \in \mathcal{T}(X)$ and $\mathcal{T}(K)$ be its mesh. Under (H_1) to (H_6) and with notations given previously, $f_{vis}^{\bar{K}}$ is a function on \bar{K} if and only if $P^1 g_{\mathcal{T}(K)}$ is an injective function from \widehat{K} onto \bar{K} .

Proof. Let $x \in \bar{K}$ and $P^1 g_{\mathcal{T}(K)}$ be an injective function. By lemma 2.1, $P^1 g_{\mathcal{T}(K)}$ is a bijective function, there is at least one simplex containing x . There are two possibilities. First, if x belongs to the interior of a simplex $\Lambda(N)$, then thanks to the bijectivity of $P^1 g_{\mathcal{T}(K)}$ and (H_6) , this simplex $\Lambda(N)$ is unique and $f_{vis}^{\bar{K}}(x) = f_{vis}^{\Lambda(N)}(x)$. In the other case, x belongs to the boundary of one or more simplexes. If x is a node of a simplex, $x = N$, then $f_{vis}^{\bar{K}}(x) = f_{num}^K(N)$. Otherwise, x can be common to two simplexes $\Lambda(N^1)$ and $\Lambda(N^2)$ such that $x = x_1 N_1 + x_2 N_2$ where $(N_i)_{i=1,2}$ and $(x_i)_{i=1,2}$ are unique by bijectivity of $P^1 g_{\mathcal{T}(K)}$. Thus, $f_{vis}^{\bar{K}}(x) = \sum_{i=1}^2 x_i f_{num}^K(N_i)$. As f_{num}^K is a function on K , this proves that $f_{vis}^{\bar{K}}$ is a function on \bar{K} .

Conversely, if $P^1 g_{\mathcal{T}(K)}$ is not an injective function, an element x of \bar{K} can be the image of several \hat{x}_j where $j > 1$, $j \in \mathbb{N}$. Hence, x can belong to several simplexes $\Lambda(N^j)$: $\{x\} = \left\{ \sum_{i=1}^{n_K+1} x_i^j N_i^j / x \in \Lambda(N^j), N^j = (N_i^j)_{i=1, \dots, n_K+1}, (x_i^j)_j \in [0, 1]^{n_K+1} \right\}$. Thus, $f_{vis}^{\bar{K}}(\{x\}) = \left\{ f_{vis}^{\Lambda(N)}(x) / \forall N \in \Theta_K, x \in \Lambda(N) \right\} = \left\{ \sum_{i=1}^{n_K+1} x_i^j f_{num}^K(N_i^j) \right\}$. It involves that $f_{vis}^{\bar{K}}$ is not a function on \bar{K} . ■

The construction of the mesh $\mathcal{T}(\widehat{K})$ is guided by the control of the errors made when the geometry is approximated by simplexes and the functions are approximated by affine functions.

2.3. Quantification of the visualization error

We introduce a quantification of the error between f_{num} and f_{vis} . This error will be used to lead the construction of f_{vis} in an adaptive manner. Actually, definition of both f_{num} and f_{vis} being local to any $K \in \mathcal{T}(X)$, this error is locally estimated for any given $K \in \mathcal{T}(X)$. The main difficulty is the possible non-matching between the supports (see (3)) or the range (see (4)) of these functions. To this purpose, we introduce Hausdorff distance on $\mathbb{X} \times \mathbb{Y}$ allowing to measure the gap between the graph of two functions:

$$d_H^{\mathbb{X} \times \mathbb{Y}} : (f, \tilde{f}) \in (C_c^0(\mathbb{X}, \mathbb{Y}))^2 \mapsto \max \left(\sup_{x \in \text{Supp} f} \inf_{\tilde{x} \in \text{Supp} \tilde{f}} d((x, f(x)), (\tilde{x}, \tilde{f}(\tilde{x}))), \sup_{\tilde{x} \in \text{Supp} \tilde{f}} \inf_{x \in \text{Supp} f} d((x, f(x)), (\tilde{x}, \tilde{f}(\tilde{x}))) \right),$$

where d is a distance over $\mathbb{X} \times \mathbb{Y}$, with positive parameters α and β :

$$d : ((x, y), (\tilde{x}, \tilde{y})) \in (\mathbb{X} \times \mathbb{Y})^2 \mapsto \max(\alpha \|x - \tilde{x}\|_{\mathbb{X}}, \beta \|y - \tilde{y}\|_{\mathbb{Y}}).$$

The two constants α and β are scaling parameters to allow pertinent comparisons between values on \mathbb{X} and \mathbb{Y} . The evaluation of this distance is very expansive. Thus, in section 3, *a posteriori* estimates are defined to evaluate Hausdorff distance.

3. Construction of a well-suited visualization

This section is devoted to the construction of the visualization of f_{num} which verifies the properties summarized in objectives (O_1) - (O_2) - (O_3) . First, a decomposition of all elements is explained which gives a sketch of algorithm. Then, *a posteriori* estimates, which will be used in place of Hausdorff distance, are defined.

3.1. Fulfilment of objective (O_3)

In order to guarantee objective (O_3) , we propose an approach based on the decomposition of all boundaries elements in lower dimension and a construction of a mesh for each element. First, elements are collected in sets $\mathcal{T}_{iD}(X) \subset \mathcal{T}(X)$ with $i = 1, 2$ with respect to their dimension:

$$\mathcal{T}_{iD}(X) := \{K \in \mathcal{T}(X), n_K = i\}.$$

To decompose each element, we define the set of all edges

$$\mathcal{E}(X) = \{E \subset \partial X \cap \partial K, K \in \mathcal{T}_{2D}(X)\} \cup \{\Sigma, \exists!(K, K') \in \mathcal{T}_{2D}(X)^2, K \neq K', \Sigma = \partial K \cap \partial K'\},$$

where ∂X (resp. ∂K) denotes the boundary of the set X (resp. of K). Each element $K \in \mathcal{T}(X)$ admits a boundary $\partial K \subset K$ which can be decomposed in edges $\mathcal{E}(K)$ such that $\mathcal{E}(K) = \mathcal{E}(X) \cap K$. This decomposition satisfies $\partial K = \cup_{E \in \mathcal{E}(K)} E$ and the following property: for any couple $(E, E') \in \mathcal{E}(K)^2$, $E \neq E'$, $\text{Int}(E) \cap \text{Int}(E') = \emptyset$. The bijection g_K thus involves a decomposition of the boundary of \widehat{K} in the same way: $\mathcal{E}(\widehat{K}) := \{\widehat{E} \in \widehat{\partial K}, g_K(\widehat{E}) \in \mathcal{E}(K)\}$. Finally, we populate the following sets $\widetilde{\mathcal{T}}_{iD}(X)$ for $i = 1, 2$:

$$\widetilde{\mathcal{T}}_{2D}(X) = \mathcal{T}_{2D}(X), \quad \widetilde{\mathcal{T}}_{1D}(X) = \mathcal{T}_{1D}(X) \cup \mathcal{E}(X).$$

This specific decomposition in edges is used to fulfil objective (O_3) :

Proposition 3.1 (fulfilment of O_3). *Under hypotheses (H_1) to (H_6) , let $(\mathcal{T}(\Sigma))_{\Sigma \in \widetilde{\mathcal{T}}_{iD}(X), i=1,2}$ be a set of meshes in simplices satisfying*

$$\forall K \in \mathcal{T}_{2D}(X), \mathcal{T}(K)|_{\partial \widehat{K}} = \mathcal{T}(K) \cap \widehat{\partial K} = \bigcup_{\Sigma \in \mathcal{E}(K)} \mathcal{T}(\Sigma), \quad (6)$$

with $\widehat{K} = \cup_{S \in \mathcal{T}(K)} S$. Then, the representation f_{vis} of f_{num} constructed from $\mathcal{T}(K)$ for all $K \in \mathcal{T}_{2D}(X)$ fulfils objective (O_3) .

Proof. Let $\Sigma \in \widetilde{\mathcal{T}}_{1D}(X)$ such that $\Sigma \notin \partial X$. There exists two elements K and K' of $\mathcal{T}(X)$, such that $K \neq K'$ and $\Sigma = \partial K \cap \partial K'$. Let assume that $f_{num|\Sigma}^K = f_{num|\Sigma}^{K'}$ on Σ where $f_{num|\Sigma}^K$ denotes the restriction of f_{num}^K on Σ . By hypothesis, there exists a mesh $\mathcal{T}(\Sigma)$ of Σ . Let S be an edge of $\mathcal{T}(\Sigma)$ whose extremities are noted a and b . According to section 2.2, $f_{vis|S}^{\widetilde{K}}$ and $f_{vis|S}^{\widetilde{K}'}$ are affine functions defined from $f_{num}^K(a)$ to $f_{num}^K(b)$, by continuity of f_{num} on Σ . Hence, $f_{vis|S}^{\widetilde{K}} = f_{vis|S}^{\widetilde{K}'}$ for any edge S of $\mathcal{T}(\Sigma)$. Finally, $f_{vis|\Sigma}^{\widetilde{K}} = f_{vis|\Sigma}^{\widetilde{K}'}$ and no artificial discontinuities are added. ■

Proposition 3.1 thus provides a "simple" algorithm to ensure objective (O_3):

1. First, construct meshes of elements of $\widetilde{\mathcal{T}}_{1D}(X)$.
2. Then, create meshes of elements of $\widetilde{\mathcal{T}}_{2D}(X)$ from those of $\widetilde{\mathcal{T}}_{1D}(X)$ and satisfying proposition 3.1.

3.2. Fulfilment of objective (O_2): elements of $\widetilde{\mathcal{T}}_{2D}(X)$

As Hausdorff distance involves two embedded optimizations, it should be quite expensive to employ. Thus, we introduce the following application for any given $K \in \mathcal{T}(X)$ and its associated mesh $\mathcal{T}(K)$

$$\Delta_{\mathcal{T}(K)} : (f, \widetilde{f}) \in C^0(K, \mathbb{Y}) \times C^0(\widetilde{K}, \mathbb{Y}) \mapsto \sup_{\hat{x} \in \widetilde{K}} \delta_{\mathcal{T}(K)}(f, \widetilde{f}; \hat{x}),$$

where

$$\delta_{\mathcal{T}(K)}(f, \widetilde{f}; \hat{x}) = \max \left(\alpha \left\| g_K(\hat{x}) - P^1 g_{\mathcal{T}(K)}(\hat{x}) \right\|_{\mathbb{X}}, \beta \left\| (f \circ g_K)(\hat{x}) - (\widetilde{f} \circ P^1 g_{\mathcal{T}(K)})(\hat{x}) \right\|_{\mathbb{Y}} \right),$$

and we prove next result

Lemma 3.2. *Let $K \in \mathcal{T}(X)$, under (H_3), (H_6) and the notations given previously, the estimate $\Delta_{\mathcal{T}(K)}$ verifies:*

$$\forall (f, \widetilde{f}) \in C^0(K, \mathbb{Y}) \times C^0(\widetilde{K}, \mathbb{Y}), d_H^{\mathbb{X} \times \mathbb{Y}}(f, \widetilde{f}) \leq \Delta_{\mathcal{T}(K)}(f, \widetilde{f}).$$

Proof. First, let $x \in K$ be any given point. By (H_3), we can note $\hat{x} = g_K^{-1}(x)$. As $P^1 g_{\mathcal{T}(K)}(\hat{x}) \in \widetilde{K}$ it comes $\inf_{\widetilde{x} \in \widetilde{K}} d((x, f(x)), (\widetilde{x}, \widetilde{f}(\widetilde{x}))) \leq d((g_K(\hat{x}), f \circ g_K(\hat{x})), (P^1 g_{\mathcal{T}(K)}(\hat{x}), \widetilde{f} \circ P^1 g_{\mathcal{T}(K)}(\hat{x})))$, and then it infers that the supremum over all $x \in K$ of the left hand term is lower than $\Delta_{\mathcal{T}(K)}(f, \widetilde{f})$.

Conversely, let $\widetilde{x} \in \widetilde{K}$ be arbitrarily given. From lemma 2.1 and the surjectivity of $P^1 g_{\mathcal{T}(K)}$, there exists $\hat{x} \in \widetilde{K}$ such that $\widetilde{x} = P^1 g_{\mathcal{T}(K)}(\hat{x})$. As $g_K(\hat{x}) \in K$, it comes $\inf_{x \in K} d((x, f(x)), (\widetilde{x}, \widetilde{f}(\widetilde{x}))) \leq d((g_K(\hat{x}), f \circ g_K(\hat{x})), (\widetilde{x}, \widetilde{f}(\widetilde{x})))$. Then, the supremum over all $\widetilde{x} \in \widetilde{K}$ of the left hand term is lower than $\Delta_{\mathcal{T}(K)}(f, \widetilde{f})$.

Finally, the two parts of d_H under the max are bounded from above by $\Delta_{\mathcal{T}(K)}(f, \widetilde{f})$ giving the result. ■

For any given $K \in \mathcal{T}(X)$ we identify f_{num}^K and $f_{vis}^{\widetilde{K}}$ with their extensions by zero off K and finally we can formulate the *a posteriori* estimate

Proposition 3.3 (A posteriori estimate). *With notations introduced in section 2, under (H_1) to (H_6), if $P^1 g_{\mathcal{T}(K)}$ is an injective function for any element $K \in \mathcal{T}(X)$, then we have*

$$d_H^{\mathbb{X} \times \mathbb{Y}}(f_{num}, f_{vis}) = \max_{K \in \mathcal{T}(X)} d_H^{\mathbb{X} \times \mathbb{Y}}(f_{num}^K, f_{vis}^{\widetilde{K}}) \leq \max_{K \in \mathcal{T}(X)} \Delta_{\mathcal{T}(K)}(f_{num}^K, f_{vis}^{\widetilde{K}}).$$

Proof. Let $K \in \mathcal{T}(X)$ be an arbitrary cell as defined in (H_1) to (H_3). We have to check that f_{num}^K and $f_{vis}^{\widetilde{K}}$ are continuous and then result comes by applying the lemma 3.2 on K (H_6 is satisfied). Actually, if (H_4) and (H_5) are verified then f_{num}^K is continuous on K . By hypothesis, $P^1 g_{\mathcal{T}(K)}$ is an injective function, then from proposition 2.3, $f_{vis}^{\widetilde{K}}$ is continuous on \widetilde{K} . ■

When for any $K \in \mathcal{T}(X)$, $P^1 g_{\mathcal{T}(K)}$ is an injective function, the estimate $\Delta_{\mathcal{T}(K)}$ control Hausdorff distance which is the criterion to evaluate the error between f_{num} and f_{vis} (see section 2.3). As $\Delta_{\mathcal{T}(K)}$ is localized in K , it provides a tool to measure the quality of the representation $f_{vis}^{\widetilde{K}}$ at each step of its construction. Then, the convergence of this estimate (under the tolerance ε) ensures that objective (O_2) is satisfied.

3.3. Fulfilment of objective (O_2): specific treatment for elements of $\widetilde{\mathcal{T}}_{1D}(X)$

Construction of $f_{vis}^{\widetilde{K}}$ will be based on an auto-adaptive method using Bowyer-Watson algorithm where we create a refined enough mesh $\mathcal{T}(K)$, for each cell K , to reach a prescribed tolerance. Proposition 3.3 shows that the construction of $f_{vis}^{\widetilde{K}}$ can be led directly over \widetilde{K} by controlling the function $\Delta_{\mathcal{T}(K)}(f_{num}^K, f_{vis}^{\widetilde{K}})$. Then, if this last quantity tends to zero, the convergence between the visualization f_{vis} and the numerical solution f_{num} is thus ensured. Note that this convergence includes both the difference in values of these functions and the matching of their supports. At last, the subscript $\mathcal{T}(K)$ introduced in the notation of $\Delta_{\mathcal{T}(K)}$ shows the dependence on the mesh of K .

Now proposition 3.3 introduced a new hypothesis assuming that $P^1 g_{\mathcal{T}(K)}$ is injective. Actually, it can be transgressed when refining $\widehat{S} \in \mathcal{T}(\widehat{K})$ by addition of a new vertex $\hat{x} \in \widehat{S}$ but such that $x = g_K(\hat{x})$ does not lie in $P^1 g_{\mathcal{T}(K)}(\widehat{S})$. Figure 4 shows an example where $P^1 g_{\mathcal{T}(K)}$ fails to satisfy this condition. One sees an area in \widetilde{K} where each point got several values of $f_{vis}^{\widetilde{K}}$. As a consequence, there is an overlapping area where $f_{vis}^{\widetilde{K}}$ is not well-defined.

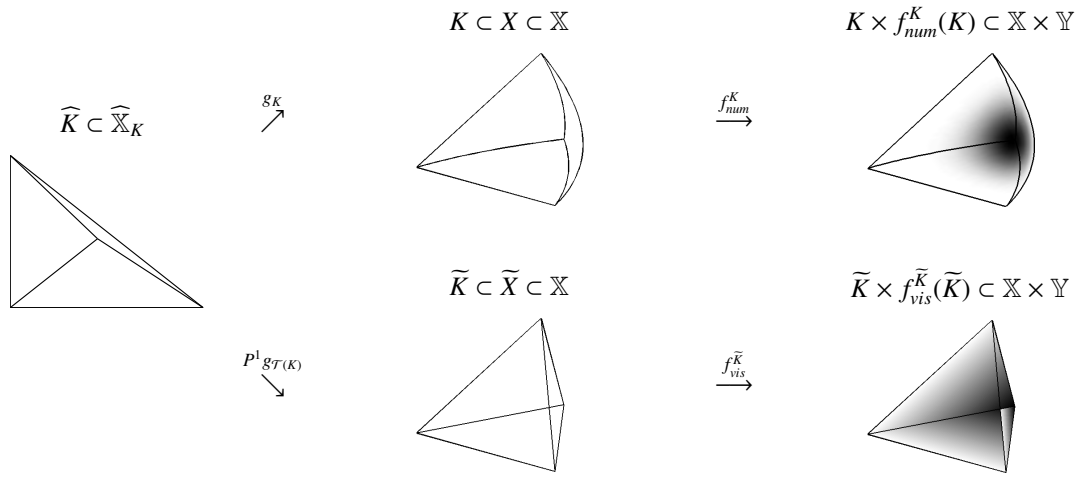


Figure 4: Visualization of a function f_{num}^K defined on an element K and its representation $f_{vis}^{\widetilde{K}}$ constructed on an element \widetilde{K} : $P^1 g_{\mathcal{T}(K)}$ is not bijective from \widetilde{K} to \widehat{K} .

For $\mathbb{X} = \mathbb{R}^2$, this phenomenon can be decomposed into two categories:

1. *In-domain vertex insertion*: a point is added inside the former \widetilde{K} (the last tessellation before addition of this point).
2. *Out-domain vertex insertion*: a point is added outside the former \widetilde{K} .

First case can be usually solved using swap techniques and sometimes additional vertices inside the domain. But, if the second case happens, the effective boundary of \widetilde{K} is no longer $P^1 g_{\mathcal{T}(K)}(\partial\widetilde{K})$ (see figure 4). As long as x stays outside the domain delimited by $P^1 g_{\mathcal{T}(K)}(\partial\widetilde{K})$, regardless to swap that can be committed, $\partial\widetilde{K}$ differs from $P^1 g_{\mathcal{T}(K)}(\partial\widetilde{K})$ and the problem remains. Thus, the only solution is to add points on $\partial\widetilde{K}$ to recover $\partial\widetilde{K} = P^1 g_{\mathcal{T}(K)}(\partial\widetilde{K})$. If this operation is done careless of the mesh performed for any cell of $\mathcal{T}(X)$ sharing this piece of $g_K(\partial\widetilde{K})$, then hypothesis of proposition 3.1 is transgressed and by the way, (O_3) is lost. To bypass this problem, we should update the boundary mesh of all these adjacent cells at the risk of an expensive and potentially diverging algorithm. Our point is therefore to propose a fine enough mesh of $\partial\widetilde{K}$ such that the out-domain vertex insertion never occurs. This will be done by *a priori* identifying the location of those bad points, called Rollover Areas, to ensure the preservation of the boundary mesh ($\partial\widetilde{K} = P^1 g_{\mathcal{T}(K)}(\partial\widetilde{K})$) during the full meshing process. Hence, proposition 3.1 applies and thus (O_3) is verified. Finally, both vertex insertion problems can be avoided letting $P^1 g_{\mathcal{T}(K)}$ be injective.

Note that for $\mathbb{X} = \mathbb{R}^3$, to the best of our knowledge, the question of guarantying the injectivity of $P^1 g_{\mathcal{T}(K)}$ when meshing a curved surface defined on a reference plane is not issued. However, the process that will be proposed as a

consequence of the management of possible Rollover Areas in $\mathbb{X} = \mathbb{R}^2$ is straightforwardly extended in $\mathbb{X} = \mathbb{R}^3$ and gives an efficient tool to handle for specific visualization artefact in the curved surface cases.

This section is thus devoted to first identify the Rollover Areas and then introduce an accuracy control of those areas for $\mathbb{X} = \mathbb{R}^2$.

3.3.1. Definition of the Rollover Areas

Let $K \in \mathcal{T}(X)$, $\mathcal{T}(K)$ be a mesh associated to this element such that $P^1 g_{\mathcal{T}(K)}$ is injective from \widehat{K} to $\widetilde{K} = \cup_{S \in \mathcal{T}(K)} S$. The Rollover Area of K is defined as:

$$\mathcal{R}_{\mathcal{T}(\partial K)}^K = K \setminus \text{Int}(\widetilde{K}). \quad (7)$$

Its inverse image in the reference element is $\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K = g_K^{-1} (K \setminus \text{Int}(\widetilde{K}))$. The subscript indicates that this area depends only on the mesh of the boundary.

Proposition 3.4. *Let $K \in \mathcal{T}(X)$ and assume that hypotheses (H_1) to (H_6) are satisfied. Let $\mathcal{T}(K)$ be a given mesh of K with $P^1 g_{\mathcal{T}(K)}$ being injective from \widehat{K} to \widetilde{K} . Finally, let $\hat{x} \in \widehat{K}$ be a given point and note $\mathcal{T}_{\hat{x}}(\widehat{K})$ the mesh refined by the insertion of the new vertex \hat{x} in $\mathcal{T}(\widehat{K})$. If $x = g_K(\hat{x})$ is an out-domain vertex (ie $x \in \mathcal{R}_{\mathcal{T}(\partial K)}^K$), then $P^1 g_{\mathcal{T}_{\hat{x}}(K)}$, the function associated to $\mathcal{T}_{\hat{x}}(K)$, is not injective.*

Proof. Let $\hat{x} \in \widehat{S}$ such that $\hat{x} \in \widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$ where $\widehat{S} \in \mathcal{T}(\widehat{K})$. Following definition of $\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$ given in (7), $x = g_K(\hat{x}) \notin \text{Int}(\widetilde{K})$, then $x \in \partial \widetilde{K}$ or $x \notin P^1 g_{\mathcal{T}(K)}(\widetilde{S})$. In the first case, as $x \in \partial \widetilde{K}$, two elements are flat and then $P^1 g_{\mathcal{T}_{\hat{x}}(K)}$ is not injective. In the second case, three configurations are possible (see [19]): \hat{x} belongs to the interior of a simplex \widehat{S} which is unique by (H_6) , \hat{x} belongs to the common edge of two simplexes or \hat{x} belongs to the boundary of \widehat{K} . In the first (resp. second) case, three (resp. four) simplexes are created instead of the simplex \widehat{S} (resp. two simplexes). We consider the first case, the proof is similar for the second configuration. Let denote $\widehat{S}_1, \widehat{S}_2$ and \widehat{S}_3 the simplexes created from \hat{x} and the three nodes of \widehat{S} and S_1, S_2 and S_3 their images by $P^1 g_{\mathcal{T}_{\hat{x}}(K)}$. Let us prove there exist i and $k, i \neq k$, such that $\text{Int}(S_i) \cap \text{Int}(S_k) \neq \emptyset$. We prove it by contradiction. We assume that $\text{Int}(S_i) \cap \text{Int}(S_k) = \emptyset$ for each $i \neq k$. In that case, $\cup_{i=1}^3 S_i = S$ and $x = g_K(\hat{x}) \in S$ which is absurd. Thus, there exists $y \in \text{Int}(S_i) \cap \text{Int}(S_k), i \neq k$, such that $y = P^1 g_{\mathcal{T}_{\hat{x}}(K)}(\widehat{y}_i) = P^1 g_{\mathcal{T}_{\hat{x}}(K)}(\widehat{y}_k)$ with $\widehat{y}_i \in \widehat{S}_i$ and $\widehat{y}_k \in \widehat{S}_k$. Finally, $P^1 g_{\mathcal{T}_{\hat{x}}(K)}$ is not injective as there is an overlapping area. ■

The definition of the Rollover Area given by equation (7) is pretty uneasy from a practical point of view as it is implicit, so we will introduce new sets noted \mathcal{V}_S^K , to be used instead of $\mathcal{R}_{\mathcal{T}(\partial K)}^K$, such that: $\mathcal{R}_{\mathcal{T}(\partial K)}^K \subset \cup_{S \in \mathcal{T}(\partial K)} \mathcal{V}_S^K$.

Definition 3.5. *Let $K \in \mathcal{T}(X)$, $\mathcal{T}(K)$ be its mesh and $\widehat{S} = [\hat{a}\hat{b}] \in \mathcal{T}(\partial \widehat{K})$ a boundary segment, where the brackets mean that a segment is considered. We note $\widehat{S}' = g_K^{-1}(P^1 g_{\mathcal{T}(K)}(\widehat{S}) \cap K)$ and \hat{n} the inward normal of \widehat{S} pointed toward \widehat{K} . We define the two following quantities.*

1. We note \hat{n}_a and \hat{n}_b , the vectors associated to vertices \hat{a} and \hat{b} ,

$$\hat{n}_a = \arg \min_{\hat{u} \in \left\{ \hat{n}, \frac{\hat{a}\hat{y}}{\|\hat{a}\hat{y}\|} \text{ where } \hat{y} \in \widehat{S}' \setminus \{\hat{a}\} \right\}} \left(\frac{\hat{u} \cdot \hat{a}\hat{b}}{\|\hat{a}\hat{b}\|} \right) \quad \text{and} \quad \hat{n}_b = \arg \min_{\hat{u} \in \left\{ \hat{n}, \frac{\hat{b}\hat{y}}{\|\hat{b}\hat{y}\|} \text{ where } \hat{y} \in \widehat{S}' \setminus \{\hat{b}\} \right\}} \left(\frac{\hat{u} \cdot \hat{b}\hat{a}}{\|\hat{b}\hat{a}\|} \right)$$

2. Let $\hat{x} = (1 - \alpha)\hat{a} + \alpha\hat{b} \in \widehat{S}$ and $\hat{n}_x = (1 - \alpha)\hat{n}_a + \alpha\hat{n}_b$, where $\alpha \in [0, 1]$. Let $\hat{y}^* \in \widehat{S} \setminus g_K^{-1}(\widetilde{K})$ be arbitrary,

$$\begin{cases} t_{\hat{x}} = \max \left\{ t, \hat{x} + t\hat{n}_x \in \partial \widehat{K} \right\}, & \text{if } (ab \wedge ag_K(\hat{y}^*)) \cdot (ab \wedge ag_K(\hat{x} + t\hat{n}_x)) > 0, \\ t_{\hat{x}} = \max \left\{ t, \hat{x} + t\hat{n}_x \in \widehat{K}, ab \wedge ag_K(\hat{x} + t\hat{n}_x) = 0 \right\} \cup \{0\}, & \text{else.} \end{cases}$$

Finally, with those notations, we set

$$\widehat{\mathcal{V}}_S^K = \left\{ \hat{x} + t\hat{n}_x, \hat{x} = (1 - \alpha)\hat{a} + \alpha\hat{b} \in \widehat{S}, \hat{n}_x = (1 - \alpha)\hat{n}_a + \alpha\hat{n}_b, \alpha \in [0, 1], 0 \leq t \leq t_{\hat{x}} \right\} \quad (8)$$

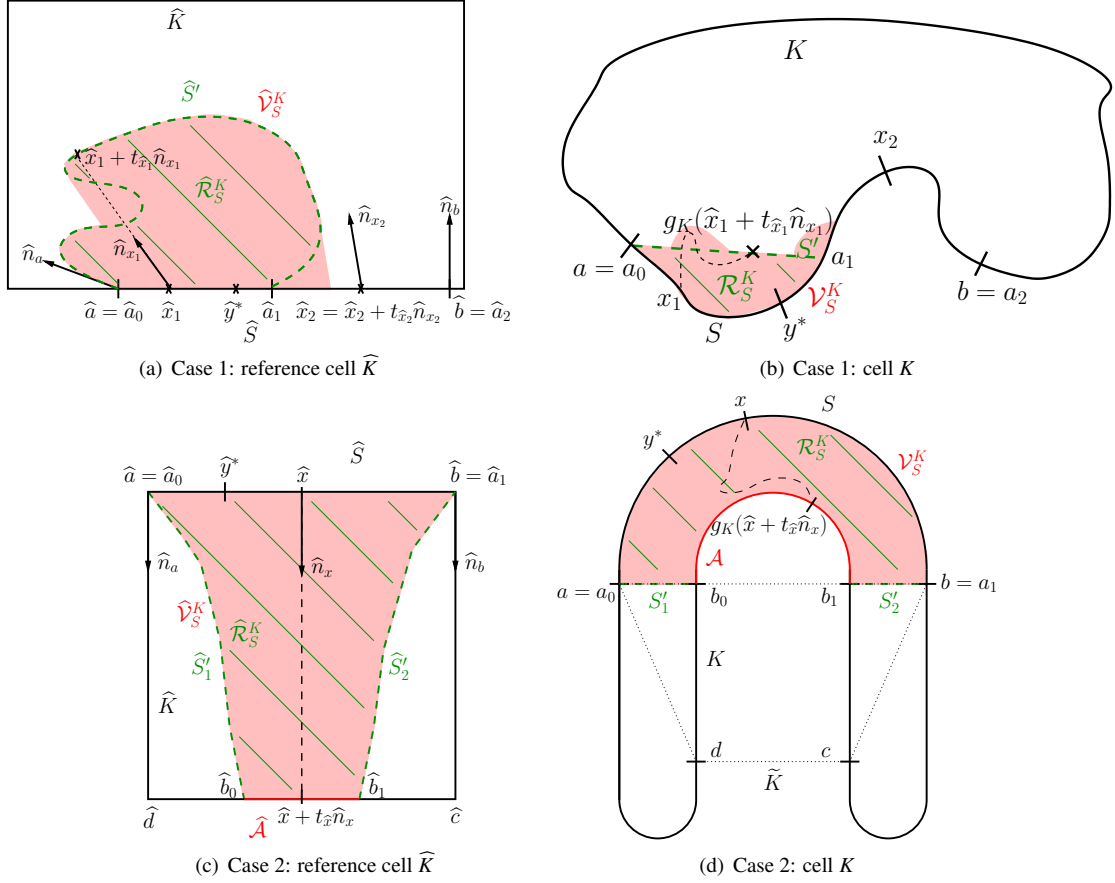


Figure 5: Illustration of the rollover areas and all the notations in two cases.

Note that the first case of $t_{\widehat{x}}$ is well-defined as \widehat{K} is convex and thus there are at most two points on $\partial\widehat{K}$ intersecting the line issued from \widehat{x} with direction \widehat{n}_x (they can be equal, thus corresponding $t = 0$, see for instance when \widehat{x} is the vertex of \widehat{K} being a triangle). This case is illustrated in figures 5.c and 5.d. Furthermore, the set $\widehat{S} \setminus g_K^{-1}(\widehat{K})$ used to choose \widehat{y}^* can be empty, nevertheless, it means that $\widehat{\mathcal{V}}_S^K = \widehat{S}$ and naturally $t_{\widehat{x}} = 0$. The second case of $t_{\widehat{x}}$ is shown in figures 5.a and 5.b.

Lemma 3.6. Let $K \in \mathcal{T}(X)$, $\mathcal{T}(K)$ be a mesh of K , $\mathcal{R}_{\mathcal{T}(\partial K)}^K$ and $\mathcal{V}_S^K = g_K(\widehat{\mathcal{V}}_S^K)$ given respectively by (7) and (8), then

$$\mathcal{R}_{\mathcal{T}(\partial K)}^K \subset \cup_{S \in \mathcal{T}(\partial K)} \mathcal{V}_S^K$$

Proof. To prove the result of this lemma, we introduce for each edge S a set \mathcal{R}_S^K , which is used to identify the connected components of $\mathcal{R}_{\mathcal{T}(\partial K)}^K$ and have a decomposition following the edges. Then, we prove that these sets \mathcal{R}_S^K verify $\mathcal{R}_{\mathcal{T}(\partial K)}^K \subset \cup_{S \in \mathcal{T}(\partial K)} \mathcal{R}_S^K$. Finally, to prove the result, we show that for each S , $\mathcal{R}_S^K \subset \mathcal{V}_S^K$.

1. Let $\widehat{S} = [\widehat{a}\widehat{b}] \in \mathcal{T}(\partial\widehat{K})$ be a boundary segment and $S = g_K(\widehat{S})$. We define the points $\{a_i\}_{i=0,N}$ as the extremities of the connected components of $g_K(\widehat{S}) \cap P^1 g_{\mathcal{T}(K)}(\widehat{S})$. The number of such points is therefore limited and defined such that $a_0 = a$, $a_N = b$, $\exists \beta_i \in [0, 1]$, $a_i = (1 - \beta_i)a + \beta_i b = g_K(\widehat{a}_i)$ where the reals β_i are sorted in ascending order. Now, we define $\widehat{\Sigma}_S^K = \cup_{i=0}^{N-1} \widehat{\Sigma}_{S,i}^K$, where

$$\widehat{\Sigma}_{S,i}^K = \begin{cases} g_K^{-1}([a_i a_{i+1}]) \cup [\widehat{a}_i \widehat{a}_{i+1}], & \text{if } [a_i a_{i+1}] \subset K, \\ g_K^{-1}([a_i a_{i+1}] \cap K) \cup \widehat{\mathcal{A}} \cup [\widehat{a}_i \widehat{a}_{i+1}], & \text{else,} \end{cases}$$

where $\widehat{\mathcal{A}}$ is an arc or an union of arcs such that $\widehat{\mathcal{A}} \subset \partial\widehat{K}$ (see figure 5). Each arc of $\widehat{\mathcal{A}}$ is composed of arcs or segments $[\widehat{b}_j\widehat{b}_{j+1}]$ such that $b_j = g_K(\widehat{b}_j) \in (\partial K \setminus S) \cap P^1 g_{\mathcal{T}(K)}(\widehat{S})$, where the points $\{\widehat{b}_j\}$ are sorted in ascending order (in the same way than $\{\widehat{a}_i\}$). To simplify the following, we will note $[\widehat{b}_j\widehat{b}_{j+1}]$ even if it is not a segment. Hence, each such $\widehat{\Sigma}_{S,i}^K$ is closed.

We note $\mathcal{B}(C)$ the bounded connected components of the complement of C in X and we introduce

$$\widehat{\mathcal{R}}_S^K = (\mathcal{B}(\widehat{\Sigma}_S^K) \cup \widehat{\Sigma}_S^K) \setminus g_K^{-1}(\text{Int}(\widehat{K})) \quad (9)$$

2. We want to show that $\mathcal{R}_{\mathcal{T}(\partial K)}^K \subset \cup_{S \in \mathcal{T}(\partial K)} \mathcal{R}_S^K$. First, let notice that the boundary of the Rollover Area, $\partial\mathcal{R}_{\mathcal{T}(\partial K)}^K = \partial(K \setminus \text{Int}(\widehat{K}))$ can be decomposed as

$$\partial\mathcal{R}_{\mathcal{T}(\partial K)}^K = \partial(K \cap \text{Int}(\widehat{K})^c) = (\partial K \cap \text{Int}(\widehat{K})^c) \cup (K \cap \partial\widehat{K}) = \bigcup_{S \in \mathcal{T}(\partial\widehat{K})} ((g_K(\widehat{S}) \setminus \text{Int}(\widehat{K})) \cup (P^1 g_{\mathcal{T}(K)}(\widehat{S}) \cap K)). \quad (10)$$

Then, by construction of each $\Sigma_S^K = g_K(\widehat{\Sigma}_S^K)$ and equation (10), $\partial\mathcal{R}_{\mathcal{T}(\partial K)}^K \subset \cup_{S \in \mathcal{T}(\partial K)} \Sigma_S^K$. Let now prove that $\text{Int}(\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K) \subset \cup_{S \in \mathcal{T}(\partial K)} \mathcal{R}_S^K$. We decompose $\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$ as $\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K = \cup_{\widehat{C}} \widehat{C}$ where \widehat{C} are the connected components of $\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$. We now consider any of this connected component \widehat{C} such that $\text{Int}(\widehat{C}) \neq \emptyset$ (otherwise $\widehat{C} \subset \partial\widehat{K}$ and the result is immediate). Then, $\exists \widehat{S}_i = [\widehat{a}_i\widehat{a}_{i+1}] \subset \widehat{S} \subset \partial\widehat{K}$, $\text{Int}_{1D}(\widehat{S}_i) \cap \widehat{C} \neq \emptyset$, where $\text{Int}_{1D}(\widehat{S}_i)$ denotes the 1D interior of the segment \widehat{S}_i , and such that $g_K(\widehat{S}_i) \neq P^1 g_{\mathcal{T}(K)}(\widehat{S}_i)$. Three cases could *a priori* occur.

- If $\widehat{C} \subset \mathcal{B}(\widehat{\Sigma}_{S,i}^K) \cup \widehat{\Sigma}_{S,i}^K$, then $\widehat{C} \subset \cup_{S \in \mathcal{T}(\partial K)} \widehat{\mathcal{R}}_S^K$.
- The second case would be that $\text{Int}(\widehat{C} \cap (\mathcal{B}(\widehat{\Sigma}_{S,i}^K) \cup \widehat{\Sigma}_{S,i}^K)) = \emptyset$, then $\widehat{C} \cap \text{Int}(\widehat{K}) = \emptyset$ or $\widehat{\Sigma}_{S,i}^K \cap \text{Int}(\widehat{K}) = \emptyset$. However, as $\text{Int}(\widehat{C}) \neq \emptyset$, $\widehat{C} \cap \text{Int}(\widehat{K}) = \emptyset$ is impossible and as $g_K(\widehat{S}_i) \neq P^1 g_{\mathcal{T}(K)}(\widehat{S}_i)$, $\widehat{\Sigma}_{S,i}^K \cap \text{Int}(\widehat{K}) = \emptyset$ is impossible too. Hence, $\text{Int}(\widehat{C} \cap (\mathcal{B}(\widehat{\Sigma}_{S,i}^K) \cup \widehat{\Sigma}_{S,i}^K)) \neq \emptyset$.
- The third case would be that $\text{Int}(\widehat{C} \cap (\mathcal{B}(\widehat{\Sigma}_{S,i}^K) \cup \widehat{\Sigma}_{S,i}^K)) \neq \emptyset$ but such that $\widehat{C} \not\subset \mathcal{B}(\widehat{\Sigma}_{S,i}^K)$. In that case, there exists a connected component $\widehat{C}' \subset (\widehat{C} \cap \mathcal{B}(\widehat{\Sigma}_{S,i}^K))$ such that its boundary $\partial\widehat{C}'$ satisfies $\partial\widehat{C}' \subset \partial\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$. Thus, \widehat{C}' is a connected component of $\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$ strictly included in \widehat{C} : this is impossible.

Hence, any connected component $\widehat{C} \subset \widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$ satisfies $\widehat{C} \subset \cup_{S \in \mathcal{T}(\partial K)} \widehat{\mathcal{R}}_S^K$ and therefore $\mathcal{R}_{\mathcal{T}(\partial K)}^K \subset \cup_{S \in \mathcal{T}(\partial K)} \mathcal{R}_S^K$.

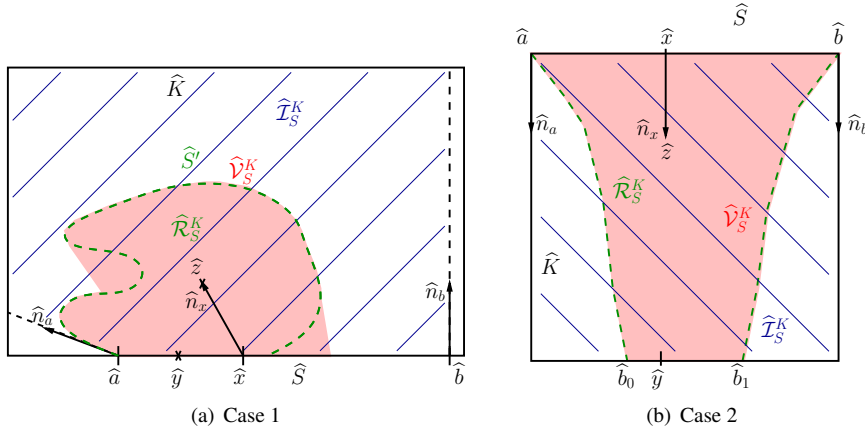


Figure 6: Illustration of the area $\widehat{\mathcal{I}}_S^K$ in two cases

3. The last step of the proof is to show that for any $S \subset \partial K$, $\mathcal{R}_S^K \subset \mathcal{V}_S^K$. Let $\widehat{z} \in \widehat{\mathcal{R}}_S^K$, to prove that $\widehat{z} \in \widehat{\mathcal{V}}_S^K$, we show first that \widehat{z} can be written as $\widehat{z} = \widehat{x} + \widehat{t}\widehat{n}_x$ where $\widehat{t} \geq 0$, $\widehat{x} \in \widehat{S} = [\widehat{a}\widehat{b}]$, then we show that $\widehat{t} \leq \widehat{t}_x$. We

introduce $\varphi(\alpha, \hat{t}) := (1 - \alpha)(\hat{a} + \hat{t}\hat{n}_a) + \alpha(\hat{b} + \hat{t}\hat{n}_b)$ and the set $\widehat{\mathcal{I}}_S^K = \varphi([0, 1] \times \mathbb{R}^+) \cap \widehat{K}$ which is a convex set. Let prove that $\hat{z} \in \widehat{\mathcal{I}}_S^K$. It is obvious that $\hat{t} \geq 0$, otherwise for any $\hat{x} \in \widehat{S}$, $\hat{x} + \hat{t}\hat{n}_x \notin \widehat{K}$. Let prove that $\hat{n}_a = \arg \min_{\hat{u} \in \{\hat{n}_a, \frac{\hat{a}\hat{y}}{\|\hat{a}\hat{y}\|} \text{ where } \hat{y} \in \partial\widehat{\mathcal{R}}_S^K\}} \left(\frac{\hat{u}\hat{a}\hat{b}}{\|\hat{a}\hat{b}\|} \right)$. Let consider $\hat{y} \in \partial\widehat{\mathcal{R}}_S^K$, then there are three possibilities, $\hat{y} \in \widehat{S}'$, $\hat{y} \in \widehat{S}$ or $\hat{y} \in \widehat{\mathcal{A}}$ (see definition of $\widehat{\Sigma}_{S,i}^K$).

If $\hat{y} \in \widehat{S}$, $\hat{a}\hat{y}\hat{a}\hat{b} \geq 0$, then $\left(\frac{\hat{a}\hat{y}\hat{a}\hat{b}}{\|\hat{a}\hat{y}\|\|\hat{a}\hat{b}\|} \right) \geq 0 \geq \min_{\hat{u} \in \{\hat{n}_a, \frac{\hat{a}\hat{y}}{\|\hat{a}\hat{y}\|} \text{ where } \hat{y} \in \widehat{S}' \setminus \{\hat{a}\}\}} \left(\frac{\hat{u}\hat{a}\hat{b}}{\|\hat{a}\hat{b}\|} \right)$ as $\hat{n}_a\hat{a}\hat{b} = 0$. If $\hat{y} \in \widehat{\mathcal{A}}$, there exists two points \hat{b}_j, \hat{b}_{j+1} which are the extremities of $\widehat{\mathcal{A}}$, and \hat{y} lies on $\partial\widehat{K}$ between \hat{b}_j and \hat{b}_{j+1} , such that $\left| \widehat{b}_j\hat{a}\hat{y} \right| \leq \left| \widehat{b}_j\hat{a}\hat{b}_{j+1} \right|$ and $\left| \widehat{y}\hat{a}\hat{b}_{j+1} \right| \leq \left| \widehat{b}_j\hat{a}\hat{b}_{j+1} \right|$ (see figure 6) where $\left| \widehat{bac} \right|$ denotes the absolute value of the angle from the point a . These two inequalities show that $\arg \min_{\hat{y} \in \widehat{\mathcal{A}}} \left(\frac{\hat{a}\hat{y}\hat{a}\hat{b}}{\|\hat{a}\hat{y}\|\|\hat{a}\hat{b}\|} \right) = \arg \min_{\hat{y} \in \{\hat{b}_j, \hat{b}_{j+1}\}} \left(\frac{\hat{a}\hat{y}\hat{a}\hat{b}}{\|\hat{a}\hat{y}\|\|\hat{a}\hat{b}\|} \right)$. However, as \hat{b}_j and \hat{b}_{j+1} lie in \widehat{S}' , $\arg \min_{\hat{u} \in \{\hat{n}_a, \frac{\hat{a}\hat{y}}{\|\hat{a}\hat{y}\|} \text{ where } \hat{y} \in \partial\widehat{\mathcal{R}}_S^K\}} \left(\frac{\hat{u}\hat{a}\hat{b}}{\|\hat{a}\hat{b}\|} \right) = \hat{n}_a$. The same result holds for \hat{n}_b . Then, the definition of \hat{n}_a as an argmin on \widehat{S}' given in definition 3.5 is the same than the one evaluated in the boundary $\partial\widehat{\mathcal{R}}_S^K$. Furthermore, by continuity of the function minimized in a closed set, the argmin of this function on $\widehat{\mathcal{R}}_S^K$ lies in its boundary too. Thus, $\hat{z} \in \widehat{\mathcal{I}}_S^K$ and \hat{z} can be expressed as $\hat{z} = \hat{x} + t_{\hat{z}}\hat{n}_x$ where $\hat{x} \in \widehat{S}$ and $t_{\hat{z}} \geq 0$.

To end the proof, we need to show that $t_{\hat{z}} \leq t_{\hat{x}}$. Two cases have to be considered: $\hat{z} \in \partial\widehat{\mathcal{R}}_S^K$ and $\hat{z} \in \text{Int}(\widehat{\mathcal{R}}_S^K)$. If $\hat{z} \in \partial\widehat{\mathcal{R}}_S^K$, then there are three possibilities: $\hat{z} \in \widehat{S}$, $\hat{z} \in \widehat{\mathcal{A}}$ or $\hat{z} \in \widehat{S}'$. If $\hat{z} \in \widehat{S}$, then $t_{\hat{z}} = 0$ and $t_{\hat{z}} \leq t_{\hat{x}}$. If $\hat{z} \in \widehat{\mathcal{A}}$, $t_{\hat{z}} = t_{\hat{x}}$ as $t_{\hat{x}}$ is defined by its first case in definition 3.5, and $t_{\hat{x}}$ is the upper bound (otherwise the point is outside \widehat{K}). If $\hat{z} \in g_K^{-1}(P^1 g_{\mathcal{T}(K)} \cap K)$, then $ab \wedge ag_K(\hat{x} + t_{\hat{z}}\hat{n}_x) = 0$ and $t_{\hat{z}} \in \{t, \hat{x} + t\hat{n}_x \in \widehat{K}, ab \wedge ag_K(\hat{x} + t\hat{n}_x) = 0\}$ and $t_{\hat{z}} \leq t_{\hat{x}}$. Finally, in all cases if $\hat{z} \in \partial\widehat{\mathcal{R}}_S^K$, $t_{\hat{z}} \leq t_{\hat{x}}$. Let now consider the case $\hat{z} \in \text{Int}(\widehat{\mathcal{R}}_S^K)$. In this case, following the line from \hat{x} and guided by \hat{n}_x , there exists $\hat{y} = \hat{x} + t_{\hat{y}}\hat{n}_x$ such that $\hat{y} \in \partial\widehat{\mathcal{R}}_S^K$ and $t_{\hat{y}} \geq t_{\hat{z}}$. Thanks to the previous case, we know that $t_{\hat{y}} \leq t_{\hat{x}}$, then $t_{\hat{z}} \leq t_{\hat{x}}$. Finally, $\hat{z} \in \widehat{\mathcal{V}}_S^K$.

■

3.3.2. Definition of a specific estimate

As we now have a characterization of the Rollover Area, we want to control Hausdorff distance not only on the edge considered but on $\widehat{\mathcal{R}}_S^K$. Then, for any $K \in \mathcal{T}(X)$, $\widehat{S} \subset \widehat{E}$, $\widehat{E} \in \mathcal{T}(\partial\widehat{K})$, we define the following application:

$$\Gamma_S^K : (f_{num}^K, f_{vis}^{\overline{S}}) \in C^0(K, \mathbb{Y}) \times C^0(\widehat{S}, \mathbb{Y}) \mapsto \sup_{\hat{x} \in \widehat{S}, \hat{t} \in [0, t_{\hat{x}}]} \max \left(\alpha \left\| g_K(\hat{x} + \hat{t}\hat{n}_x) - G(f_{num}^K, f_{vis}^{\overline{S}}; \hat{x}, \hat{t}) \right\|_{\mathbb{X}}, \beta \left\| (f_{num}^K \circ g_K)(\hat{x} + \hat{t}\hat{n}_x) - F(f_{num}^K, f_{vis}^{\overline{S}}; \hat{x}, \hat{t}) \right\|_{\mathbb{Y}} \right), \quad (11)$$

where each component of G noted G_i , ($i = 1, \dots, 3$) is defined as:

$$G_i(f_{num}^K, f_{vis}^{\overline{S}}; \hat{x}, \hat{t}) = \arg \max_{G_i^l, l=1, \dots, 4} \left(\left| (g_K)_i(\hat{x} + \hat{t}\hat{n}_x) - G_i^l(f_{num}^K, f_{vis}^{\overline{S}}; \hat{x}, \hat{t}) \right| \right), \quad i = 1, \dots, 3.$$

Each function G_i^l is given by

$$G_i^l(f_{num}^K, f_{vis}^{\overline{S}}; \hat{x}, \hat{t}) = P^1 g_{\mathcal{T}(E)}(\hat{x}) + \hat{t} J_{g_K}^l \hat{n}_x, \quad l = 1, \dots, 4.$$

and

$$J_{g_K}^1 = \begin{pmatrix} \partial_1 g_1^{\max} & \partial_2 g_1^{\max} \\ \partial_1 g_2^{\max} & \partial_2 g_2^{\max} \\ \partial_1 g_3^{\max} & \partial_2 g_3^{\max} \end{pmatrix}, J_{g_K}^2 = \begin{pmatrix} \partial_1 g_1^{\min} & \partial_2 g_1^{\min} \\ \partial_1 g_2^{\min} & \partial_2 g_2^{\min} \\ \partial_1 g_3^{\min} & \partial_2 g_3^{\min} \end{pmatrix}, J_{g_K}^3 = \begin{pmatrix} \partial_1 g_1^{\min} & \partial_2 g_1^{\max} \\ \partial_1 g_2^{\min} & \partial_2 g_2^{\max} \\ \partial_1 g_3^{\min} & \partial_2 g_3^{\max} \end{pmatrix}, J_{g_K}^4 = \begin{pmatrix} \partial_1 g_1^{\min} & \partial_2 g_1^{\min} \\ \partial_1 g_2^{\min} & \partial_2 g_2^{\min} \\ \partial_1 g_3^{\min} & \partial_2 g_3^{\min} \end{pmatrix},$$

where $\partial_j g_i^{\max}$ (resp. $\partial_j g_i^{\min}$) denotes the maximum (resp. minimum) of the partial derivative of the i -th component of g_K in the j -th direction. Similarly, we define the functions F and F^l ($l = 1, \dots, 4$) by substituting g_K by $f_{num}^K \circ g_K$ and $P^1 g_{\mathcal{T}(E)}$ by $f_{vis}^{\overline{S}} \circ P^1 g_{\mathcal{T}(E)}$ and each $J_{f_{num}^K \circ g_K}^l$ is defined in the same way than $J_{g_K}^l$. Then, we prove next result.

Proposition 3.7. *With previous notations, let $K \in \mathcal{T}(X)$ and $\widehat{S} \in \mathcal{T}(\partial\widehat{K})$. Then,*

$$\forall \hat{z} \in \widehat{\mathcal{V}}_S^K, \quad \delta_{\mathcal{T}(K)} \left(f_{num}^K, f_{vis}^{\widehat{K}}; \hat{z} \right) \leq \Gamma_S^K \left(f_{num}^K, f_{vis}^{\widehat{S}} \right),$$

Proof. We just need to prove that for any mesh $\mathcal{T}(K)$ of K , such that $\mathcal{T}(K)|_{\widehat{S}} = \widehat{S}$, G satisfies for all $\hat{z} = \hat{x} + \hat{t} \hat{n}_x \in \widehat{\mathcal{V}}_S^K$, $\|g_K(\hat{z}) - P^1 g_{\mathcal{T}(K)}(\hat{z})\|_{\mathbb{X}} \leq \|g_K(\hat{z}) - G(f_{num}^K, f_{vis}^{\widehat{S}}; \hat{x}, \hat{t})\|_{\mathbb{X}}$. Then, with similar arguments the following result will be also true for any mesh, for all $\hat{z} = \hat{x} + \hat{t} \hat{n}_x \in \widehat{\mathcal{V}}_S^K$, $\|(f_{num}^K \circ g_K)(\hat{z}) - (f_{vis}^{\widehat{K}} \circ P^1 g_{\mathcal{T}(K)})(\hat{z})\|_{\mathbb{Y}} \leq \|(f_{num}^K \circ g_K)(\hat{z}) - F(f_{num}^K, f_{vis}^{\widehat{S}}; \hat{x}, \hat{t})\|_{\mathbb{Y}}$.

For $\hat{z} = \hat{x} + \hat{t} \hat{n}_x \in \widehat{\mathcal{V}}_S^K$, $P^1 g_{\mathcal{T}(K)}(\hat{z})$ can be expressed as:

$$P^1 g_{\mathcal{T}(K)}(\hat{z}) = P^1 g_{\mathcal{T}(K)}(\hat{x} + \hat{t} \hat{n}_x) = P^1 g_{\mathcal{T}(E)}(\hat{x}) + \hat{t} J_{P^1 g_{\mathcal{T}(K)}}(\hat{x}) \hat{n}_x,$$

where $J_{P^1 g_{\mathcal{T}(K)}}$ denotes the jacobian associated to $P^1 g_{\mathcal{T}(K)}$. Note that $P^1 g_{\mathcal{T}(E)}(\hat{x})$ depends only on the mesh of the edge E . Furthermore, for each component:

$$\min_{l=1,\dots,4} \left(J_{g_K}^l \hat{n}_x \right)_i \leq \left(J_{P^1 g_{\mathcal{T}(K)}}(\hat{x}) \hat{n}_x \right)_i \leq \max_{l=1,\dots,4} \left(J_{g_K}^l \hat{n}_x \right)_i, \quad i = 1, \dots, 3,$$

and

$$\left| (g_K)_i(\hat{z}) - (P^1 g_{\mathcal{T}(K)})_i(\hat{z}) \right| \leq \left| (g_K)_i(\hat{z}) - G(f_{num}^K, f_{vis}^{\widehat{S}}; \hat{x}, \hat{t}) \right|, \quad i = 1, \dots, 3,$$

which gives the result. ■

From proposition 3.7, for any $K \in \mathcal{T}(X)$, $E \in \mathcal{E}(K)$, its mesh $\mathcal{T}(E)$ and $\widehat{S} \in \mathcal{T}(\widehat{E})$, an *a posteriori* estimate Γ_S^K is constructed such that for any mesh $\mathcal{T}(K)$ satisfying $\mathcal{T}(K)|_{\widehat{S}} = \widehat{S}$:

$$\forall \hat{z} \in \widehat{\mathcal{V}}_S^K, \quad \delta_{\mathcal{T}(K)} \left(f_{num}^K, f_{vis}^{\widehat{K}}; \hat{z} \right) \leq \Gamma_S^K \left(f_{num}^K, f_{vis}^{\widehat{S}} \right).$$

From lemma 3.6, we know that the Rollover Area $\mathcal{R}_{\mathcal{T}(\partial\widehat{K})}^K$ is included in the union of all \mathcal{V}_S^K . Thus, using above estimate, we can control, in this area, the accuracy of any further 2D mesh that will be generated, whereas it is not constructed yet, by 1D evaluations over each element of the boundary mesh.

4. Algorithm

This section is devoted to the presentation of the algorithm which is summed up in algorithm 1. The organization of the section follows the steps of the construction of f_{vis} .

4.1. Computation of the extrema

Several extrema have to be computed: for the function f_{num} , its derivatives, the geometrical transformation g_K associated to each mesh cell K , its jacobian and its hessian. All these quantities are known analytically. These extrema are used either to compute the weights α and β used in the function d in Hausdorff distance or to evaluate $t_{\hat{x}}$ defined for the Rollover Area. To find these extrema, a modification of a global optimization algorithm known as DIRECT is used. DIRECT, was initially developed by Jones et al. (see [20]) to solve global optimization problems. The algorithm is deterministic and samples the domain by adding points and decides where to search depending on the information given by tested points. DIRECT, acronym of *dividing rectangles*, is developed for hyperrectangles. For our problem, simplexes are used instead of hyperrectangles which means a kind modification of the original algorithm (see [21]). The algorithm is an usual branch and bound one. For surfaces, a triangle is subdivided into four triangles by inserting the middle of edges (see [22]). The value of each function on any triangle is evaluated at its barycentre. Two strategies are used simultaneously to select triangles to subdivide. First, candidates are taken among triangles with the biggest error : it ensures a local search. Secondly, triangles with the lowest levels of subdivision (which have the biggest areas) are subdivided : it ensures a global search (see figure 7). In our algorithm, only one DIRECT is used to find all the extrema needed at once: all the constrains of search are cumulated which allow to be computationally less expansive.

Algorithm 1: Construction of f_{vis}

Input: $\mathcal{T}(X)$, f_{num} , ε

Output: f_{vis}

1. Computation of the extrema of the functions, the geometry, their gradients and the weights α and β .
 2. Decomposition of the boundary: 2D \rightarrow 1D.
 3. Mesh of the elements of $\widetilde{\mathcal{T}}_{1D}$:


```

for each edge  $E \in \widetilde{\mathcal{T}}_{1D}(X)$  do
  Computation of  $\Gamma_{\mathcal{T}(E)}$ .
  while  $\Gamma_{\mathcal{T}(E)} > \varepsilon$  do
    Refinement of  $\mathcal{T}(\widehat{E})$  and update of  $\Gamma_{\mathcal{T}(E)}$ .
  end
end

```
 4. Mesh of the elements of $\widetilde{\mathcal{T}}_{2D}$:


```

for each element  $K \in \widetilde{\mathcal{T}}_{2D}(X)$  do
  Construction of an initial mesh  $\mathcal{T}(\widehat{K})$  from meshes of the edges  $\widehat{E} \in \mathcal{E}(\widehat{K})$ .
  Computation of  $\Delta_{\mathcal{T}(K)}$ .
  while  $\Delta_{\mathcal{T}(K)} > \varepsilon$  do
    Refinement of  $\mathcal{T}(\widehat{K})$  and update of  $\Delta_{\mathcal{T}(K)}$ .
  end
end

```
 5. Generation of the output.
-

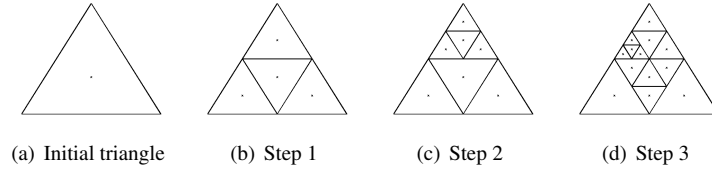


Figure 7: Step by step DIRECT method

4.2. Decomposition of the boundary

The decomposition of the boundary of the elements in edges is primordial because it permits to fulfil (O_3) as explained in proposition 3.1. Then, the mesh of all the elements of $\widetilde{\mathcal{T}}_{1D}$ has to be done once for all. This is the third step of algorithm 1. However, as any edge E can belong to multiple 2D cells, several accuracy constraints are formulated. Indeed, for each element K whose boundary contains S , we ask to verify $\Gamma_S^K \leq \varepsilon$. Note that this definition of Γ_S^K depends on the trace of f_{num}^K . The latter being the local definition of f_{num} on K , these traces can thus be different from a cell to an adjacent one. Then, to take into account all these contributions, we introduce an estimate $\Gamma_{\mathcal{T}(E)}$ defined as

$$\Gamma_{\mathcal{T}(E)} = \max_{S \in \mathcal{T}(E)} \Gamma_S = \max_{S \in \mathcal{T}(E)} \max_{\{K, S \in \mathcal{T}(\partial K)\}} \Gamma_S^K.$$

This results in the creation of a list of connected functions on E including any pure 1D function supported on E (when $E \in \mathcal{T}_{1D}$) and traces of 2D functions whose boundary of its support contains E (when $E \in \widetilde{\mathcal{T}}_{1D}$). When we consider pure 1D function or when we *a priori* know that no Rollover Area will occur (in the case of straight edges), Γ_S^K is replaced by Δ_S^K defined as:

$$\begin{aligned} \Delta_S^K : (f, \widetilde{f}) &\in C^0(\mathcal{S}, \mathbb{Y}) \times C^0(\widetilde{\mathcal{S}}, \mathbb{Y}) \\ \mapsto \sup_{\hat{x} \in \widetilde{\mathcal{S}}} \max &\left(\alpha \left\| g_E(\hat{x}) - P^1 g_{\mathcal{T}(E)}(\hat{x}) \right\|_{\mathbb{X}}, \beta \left\| (f \circ g_E)(\hat{x}) - (\widetilde{f} \circ P^1 g_{\mathcal{T}(E)})(\hat{x}) \right\|_{\mathbb{Y}} \right), \end{aligned} \quad (12)$$

When equation (12) is used with f (resp. \tilde{f}) being the trace of a given 2D function f_{num}^K (resp. f_{vis}^K) on E , g_E (resp. $P^1 g_{\mathcal{T}(E)}$) thus stands for the restriction of g_K (resp. $P^1 g_{\mathcal{T}(K)}$) to E .

4.3. Mesh of the elements of $\tilde{\mathcal{T}}_{1D}$

The mesh of the elements of $\tilde{\mathcal{T}}_{1D}$ follows a Bowyer-Watson algorithm (see [23, 24, 25]). It is an adaptive remeshing method guided by an *a posteriori* estimate. As presented in the previous section, in order to obtain a unique mesh of an element E of $\tilde{\mathcal{T}}_{1D}$, the estimate considered must be the maximum of all the accuracy constraints on E . Each constraint comes from a connected function on E for which either Δ_S^K or Γ_S^K has to be evaluated.

In the case of pure 1D function or when we know that no Rollover Area will appear, we compute Δ_S^K which is only supported on S . Thus, this optimization problem (equation (12)) is straightforwardly solved by DIRECT method 1D.

In the other cases, each function to be treated is the trace of a function f_{num}^K defined on an element $K \in \mathcal{T}_{2D}$. Thus, the estimate Γ_S^K given by equation (11) is used. In this formulation, we need to perform an optimization on a subset $\widehat{\mathcal{V}}_S^K$ of \widehat{K} . The difficulty lies in the fact that $\widehat{\mathcal{V}}_S^K$ itself is implicitly defined by another optimization problem. Then, the use of Γ_S^K should be complicated and expansive. To bypass this difficulty, we start replacing, with help of a mapping given hereafter, this not explicit domain $\widehat{\mathcal{V}}_S^K$ by an arbitrary shape $\widehat{\mathcal{V}}^0$ more suitable for optimization. We choose $\widehat{\mathcal{V}}^0$ as an isosceles triangle whose vertices are $A(0, 0), B(1, 0), C(0.5, 0.5)$. Hence, a point is referred in the isosceles triangle by its coordinates (α, t) with:

$$\alpha \in [0, 1], \quad t \in \begin{cases} [0, \alpha] & \text{if } \alpha \leq 0.5, \\ [0, 1 - \alpha] & \text{else.} \end{cases}$$

A point $\hat{z} \in \widehat{\mathcal{V}}_S^K$ can be written as $\hat{z} = \hat{x} + \hat{t}\hat{n}_x$ where $\hat{x} \in \widehat{S}$ for some $\widehat{S} \in \mathcal{T}(\widehat{E})$ whose extremities are noted \hat{a} and \hat{b} . Then, the bijection ψ from $\widehat{\mathcal{V}}^0$ onto $\widehat{\mathcal{V}}_S^K$ is defined by:

$$\psi : (\alpha, t) \mapsto \hat{z} = \hat{a} + \alpha\hat{a}\hat{b} + \hat{t}\hat{n} \quad \text{where} \quad \hat{t} = \begin{cases} \frac{t}{\alpha}t_{\hat{x}} & \text{if } 0 \leq \alpha \leq 0.5, \\ \frac{t}{1-\alpha}t_{\hat{x}} & \text{else.} \end{cases}$$

With this new parameterization, DIRECT algorithm can be used as $\widehat{\mathcal{V}}^0$ is a simplex. When evaluating the estimate Γ_S^K , we obtain the test point $\hat{z} \in \widehat{\mathcal{V}}_S^K$ by mean of ψ . Thus, for the point $\hat{x} = (1 - \alpha)\hat{a} + \alpha\hat{b} \in \widehat{S}$, we have to know the value of $t_{\hat{x}}$ which is obtained by solving an optimization problem with help of a projected gradient method. However, the combination of DIRECT and the specific shape of $\widehat{\mathcal{V}}^0$ allow to drastically limit the number of different values of $t_{\hat{x}}$ evaluated as shown hereafter.

As DIRECT is a subdivision algorithm, we introduce a numbering of the vertices in the following algorithm.

Algorithm 2: Numbering of the vertices in DIRECT

Input: m the maximum level of DIRECT

1. Initialization of the indices associated to the vertices of the triangle $\widehat{\mathcal{V}}^0$: $A(i_A = 0), B(i_B = 2^{m+1})$ and $C(i_C = 2^m)$.
 2. When a triangle of vertices $V_1(i_{V_1}), V_2(i_{V_2})$ and $V_3(i_{V_3})$ is refined, we note V_4, V_5 and V_6 the center of the segments $[V_1 V_2], [V_1 V_3]$ and $[V_2 V_3]$, thus generating next level triangles $V_1 V_4 V_5, V_2 V_4 V_6, V_3 V_5 V_6$ and $V_4 V_5 V_6$. The indices associated to the new vertices are: $i_{V_4} = \frac{i_{V_1} + i_{V_2}}{2}, i_{V_5} = \frac{i_{V_1} + i_{V_3}}{2}$ and $i_{V_6} = \frac{i_{V_2} + i_{V_3}}{2}$.
-

Algorithm 2 gives an easy way to have the abscissa of a center of gravity of a triangle, as presented in next lemma, which is the point where each estimate Γ_S^K is evaluated by DIRECT method.

Lemma 4.1. *Let m be the maximum level of DIRECT method and $\alpha_k^m = k/2^{m+1}$ for $k \in \llbracket 0, 2^{m+1} \rrbracket$. Let T be a triangle of arbitrary level i whose vertices are noted $V_1(i_{V_1}), V_2(i_{V_2})$ and $V_3(i_{V_3})$. We note G the center of gravity of this triangle T and α_G its abscissa. Then,*

1. For each vertex $V(i_V)$, its index i_V is an integer and its abscissa α_V is given by $\alpha_V = \alpha_{i_V}^m$,
2. The abscissa of G is given by $\alpha_G = \alpha_{i_G}^m$ where $i_G = (i_{V_1} + i_{V_2} + i_{V_3})/3$.

Proof. Let $V(i_V)$ be a vertex of a triangle at a level i of DIRECT. To prove that its index i_V is an integer, we see by recurrence that $i_V = 2^{m-i}k$, with $k \in \llbracket 0, 2^{i+1} \rrbracket$. The key point is to notice that if V is created at level i then it is the isobarycentre of two vertices $V_1(i_{V_1})$ and $V_2(i_{V_2})$ of level $i-1$. Thus, we have $i_V = \frac{i_{V_1} + i_{V_2}}{2} = 2^{m-i}(k_1 + k_2)$ with $k_1 + k_2 \in \llbracket 0, 2^{i+1} \rrbracket$. In a very similar way, we observe that $\alpha_V = \frac{\alpha_{V_1} + \alpha_{V_2}}{2}$ and $\alpha_{i_V}^m = \frac{\alpha_{i_{V_1}}^m + \alpha_{i_{V_2}}^m}{2}$ which involves $\alpha_V = \alpha_{i_V}^m$.

To prove the second result, we notice that an isosceles triangle whose base is parallel to (AB) is subdivided into 4 isosceles triangles whose each base is also parallel to (AB). Then, the abscissa of G is the same as the abscissa of the opposite vertex (for instance V_3) to the base (V_1V_2). Thus, $\alpha_{V_3} = \frac{\alpha_{V_1} + \alpha_{V_2}}{2}$, and from the first result, $i_{V_3} = 2^{m+1}\alpha_{V_3} = 2^{m+1}\left(\frac{\alpha_{V_1} + \alpha_{V_2}}{2}\right) = \frac{i_{V_1} + i_{V_2}}{2}$. Taking $i_G = \frac{i_{V_1} + i_{V_2} + i_{V_3}}{3}$, we have $i_G = \frac{i_{V_1} + i_{V_2}}{2} = i_{V_3}$ which ends the proof. ■

To summarize, the computation of the estimate Γ_S^K is performed in the following way. First, we initiate DIRECT method on $\widehat{\mathcal{V}}^0$. Each step of DIRECT needs to evaluate $\Gamma_S^K \circ \psi$ at the center of gravity with coordinates (α, t) of a triangle of $\widehat{\mathcal{V}}^0$. However, to calculate $\psi(\alpha, t)$, we need to know $t_{\hat{x}}$ with $\hat{x} = (1-\alpha)\hat{a} + \alpha\hat{b}$. But, with help of algorithm 2, according to lemma 4.1, we have seen that the centers of gravity are sharing a small group of \hat{x} (identified by their abscissas α_k^m). As a consequence, the maximum number of different $t_{\hat{x}}$ to evaluate is $2^{m+1} - 1$, with respect to 4^m centers of gravity generated by m level DIRECT method on a triangle. Thus, the complexity goes from $\mathcal{O}(N)$ to $\mathcal{O}(\sqrt{N})$.

Note that the vector \hat{n}_a (and \hat{n}_b) introduced to define the Rollover Area in section 3 *a priori* requires to know the inverse transformation g_K^{-1} . When it is difficult to evaluate, we simplify the expression of the term $\hat{a}g_K^{-1}(y)$ by introducing a limited development of g_K^{-1} around a and using the identity $J_{g_K^{-1}}(a) = \left(J_{g_K}(\hat{a})\right)^{-1}$, we obtain the following approximation: $\frac{\hat{a}\hat{b} \cdot \hat{a}\hat{y}}{\|\hat{a}\hat{b}\| \cdot \|\hat{a}\hat{y}\|} \approx \frac{\hat{a}\hat{b} \cdot J_{g_K}^{-1}(\hat{a})(ab)}{\|\hat{a}\hat{b}\| \cdot \|J_{g_K}^{-1}(\hat{a})(ab)\|}$ which is accurate enough when ab is small and does not need to know g_K^{-1} .

The refinement of an element $\widehat{S} \in \mathcal{T}(\widehat{E})$ is done by splitting \widehat{S} around a point realizing the maximum of the estimate. The abscissa of this point can be directly straightforwardly given by algorithm 2.

4.4. Mesh of the elements of $\widetilde{\mathcal{T}}_{2D}$

The mesh of an element of $\widetilde{\mathcal{T}}_{2D}$ starts with the recovery of all the points on the boundary (from the meshes of the boundary). Then, a first mesh $\mathcal{T}(\widehat{K})$ of an element \widehat{K} is created with help of all these points. The strategy is similar than for elements of $\widetilde{\mathcal{T}}_{1D}$. A Bowyer-Watson algorithm is used to refine this mesh. The estimate $\Delta_{\mathcal{T}(K)}$ is evaluated with DIRECT method on each simplex. The choice of the point added is given by the criterion (C):

(C) : The point \hat{x} added to the representation mesh $\mathcal{T}(\widehat{K})$ is one satisfying

$$\hat{x} = \arg \max_{\hat{y} \in \widehat{K}} \delta_{\mathcal{T}(K)} \left(f_{num}^K, f_{vis}^{\widehat{K}}; \hat{y} \right).$$

4.5. Generation of the output

The last step of the algorithm is the generation of the output at the specific format of a visualization software (for instance Gmsh [10], Paraview [26], VisIt [27]). For each element $K \in \mathcal{T}(X)$, a mesh $\mathcal{T}(\widehat{K})$ has been built. Then, following lemma 2.1, a mesh $\mathcal{T}(K)$ is constructed based on a linear mapping. Finally, the values of f_{num}^K are associated with each point of $\mathcal{T}(K)$. On a common edge to two elements, two nodes are created (one for each element) and their values can be different (if f_{num} is not continuous at this interface). It allows to observe numerical jumps which is pertinent for Discontinuous Galerkin method for instance (this duplication of the nodes was also used in [28]).

5. Convergence

We summarize in the following theorem all the hypotheses and properties that led to the algorithm proposed in section 4.

Theorem 5.1. *Let ε be a prescribed tolerance, $\mathbb{X} = \mathbb{R}^2$ and f_{num} be a suitable numerical solution (following hypotheses (H_1) to (H_5)). If algorithm 1 described in section 4 has converged then f_{vis} satisfies objectives (O_1) - (O_2) - (O_3) .*

Proof. Objective (O_1) is satisfied by construction as simplexes are used: each element $K \in \mathcal{T}(X)$ is approximated by $\tilde{K} = \cup_{S \in \mathcal{T}(K)} S$ and affine functions are defined on these simplexes.

Several ingredients are necessary to verify (O_2) : for each element $K \in \mathcal{T}(X)$, $f_{vis}^{\tilde{K}}$ has to be a function on \tilde{K} (otherwise Hausdorff distance is not well-defined) and the threshold ε must be an upper-bound of the estimate $\Delta_{\mathcal{T}(K)}$. Proposition 2.3 shows that $f_{vis}^{\tilde{K}}$ is a function on \tilde{K} if and only if $P^1 g_{\mathcal{T}(K)}$ is a bijective function. To ensure that $P^1 g_{\mathcal{T}(K)}$ remains bijective at each step of the algorithm, an estimate Γ_S^K is used. Proposition 3.7 shows that this estimate in the Rollover Area is an upper-bound of any estimate $\Delta_{\mathcal{T}(K)}$ whose mesh is constructed from the mesh of the boundary. As the point which is selected and added in the representation mesh is the one where the error is maximum (criterion (C)), if the estimate Γ_S^K is bounded by the tolerance, no point will be added in the Rollover Area. Then, proposition 3.4 shows that this ensures that no exterior loss of bijectivity is possible. Finally, Hausdorff distance is well-defined at each step of the algorithm and thus can be controlled by the estimate $\Delta_{\mathcal{T}(K)}$. Proposition 3.3 proves that (O_2) is satisfied if the tolerance is an upper-bound of the estimate $\Delta_{\mathcal{T}(K)}$ for each element K .

Objective (O_3) is fulfilled by proposition 3.1. ■

6. Numerical examples

Several numerical examples are presented to demonstrate the potential of the method. The first example deals with the rendering of basis functions. Despite this case should be easy, it must be analyzed cautiously. The second numerical example is more realistic and is about the propagation of an acoustic wave. Vectorial finite elements are used to solve the problem on a hybrid mesh. The last example focus on an elasticity problem solved by discontinuous Galerkin method.

6.1. Rendering tests with Gauss-Lobatto basis functions

Basis functions are the keystone of high order methods as the numerical solutions are constructed from basis functions and degrees of freedom. In this paragraph, the rendering of some Gauss-Lobatto basis functions is done. These functions are classic in high order finite elements and discontinuous Galerkin methods, for example in spectral finite elements methods for wave propagation [3]. First, we want to compare the rendering given when the process of construction of the representation is based on L^∞ -norm and L^2 -norm. Then, in the case of curved elements, we show why the use of the specific estimate introduced in section 3.3 is crucial. Finally, the costs of the adaptive method in comparison with a uniform refinement are presented.

Figure 8 compares the representation given by the adaptive method when the the construction is guided following L^2 -norm (figures 8.b and 8.e) and L^∞ -norm (figures 8.c and 8.f) of two Gauss-Lobatto basis functions of order $r = 3$ defined on a quadrangle. In this example, we choose the unit square, the representation of its geometry is therefore trivial. Gauss Lobatto basis functions for this cell are Lagrange interpolating polynomial, φ_{ij}^{GLr} , $i, j = 1, \dots, r + 1$ (r denotes the polynomial order), defined from the Gauss Lobatto points $\xi_{ij} = (\xi_i, \xi_j)$ where ξ_i are the Gauss Lobatto points defined on the unit segment [29]. Figure 8.a is the rendering reference of the function φ_{22}^{GL3} (the representation mesh is very fine and constructed as a regular subdivision of the element). Figure 8.b is the representation given by the adaptive strategy when the construction is guided following the L^2 -norm under the tolerance 1% (the representation mesh is shown in white). One sees that the rendering is very far from the reference even if the tolerance is quite fine. Figure 8.c is obtained by our approach (with L^∞ -norm) with the same tolerance of 1%. The difference in comparison to the previous one is the pointwise control. In the second case (figures 8.d, 8.e and 8.f), the function is the sum of two Gauss Lobatto basis functions: $\varphi_{22}^{GL3} + 0.05 \varphi_{33}^{GL3}$. This case is close to the previous one but with the addition of a second basis function whose amplitude is 5% of the first one. This function is shown in figure 8.d and the rendering given by the L^2 -norm (resp. L^∞ -norm) is done in figure 8.e (resp. 8.f) at the same tolerance (1 %). When L^2 -norm is used, the additional basis function is not "detected" and the rendering is almost similar to the first case. This additional basis function should be detected as the tolerance is lower than the rate of amplitude (5%) which means than it should not be neglected. A significant part of the information is therefore lost. The control by the L^∞ -norm enables to avoid

this kind of visualization artifact. Finally, a plot over one diagonal of the quadrangle is performed in figure 8.g for the six representations. The extremities of the line are two vertices of the square: (0,0) and (1,1). This extraction confirms that the rendering given by the L^2 -norm does not detect the second basis function. This is particularly annoying as the interpolating point of this basis function lies in this line.

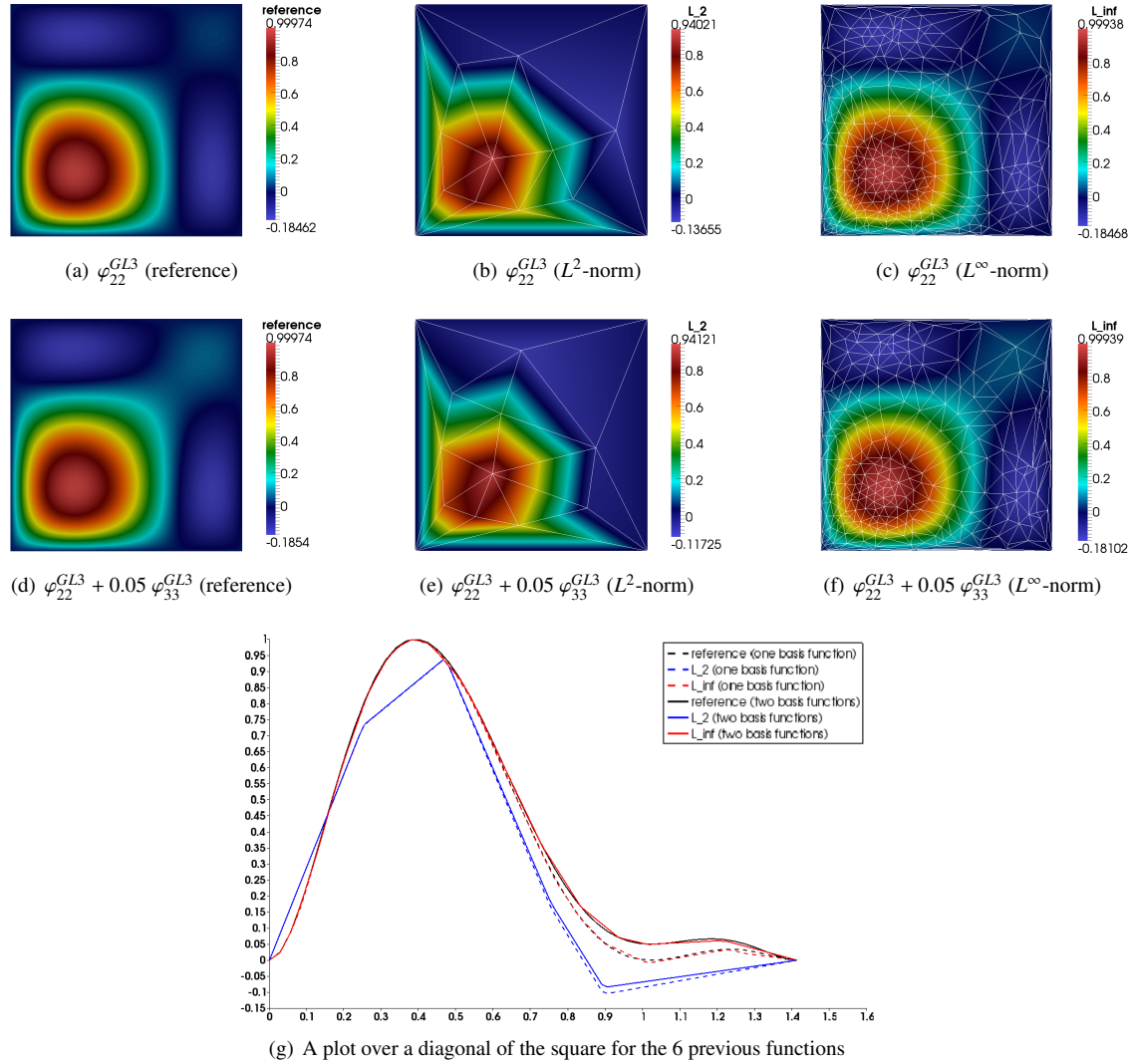


Figure 8: Representation of Gauss Lobatto basis functions of order 3 on a quadrangle (accuracy 1%)

The second example is a Gauss-Lobatto basis function of order 6 defined on a quadrangle Q^2 (of order 2, defined from 9 nodes): φ_{32}^{GL6} . In figure 9, the two estimates Δ_S^K (figures 9.a and 9.c) and Γ_S^K (figures 9.b and 9.d) given in section 3.3 are compared. The representation mesh is shown and the mesh of the boundary (thick black line). As expected, when the estimate Δ_S^K is used, the construction of the boundary mesh takes into consideration only the trace of the function. The consequence is that points can be added in the Rollover area (see section 3.3.1) and there is an overlapping. In figure 9.c, one can see that some points are outside the mesh boundary (thick line). However, when Γ_S^K is used, the mesh of the boundary is finer than previously as the vicinity of the boundary is taking into consideration. Thus, there is no overlapping and no loss of bijectivity. Note that the number of elements is similar in both cases and even lower with Γ_S^K (216 triangles) than with Δ_S^K (221 triangles).

Figure 10 compares the performance for adaptive and uniform methods in the two cases presented previously. As

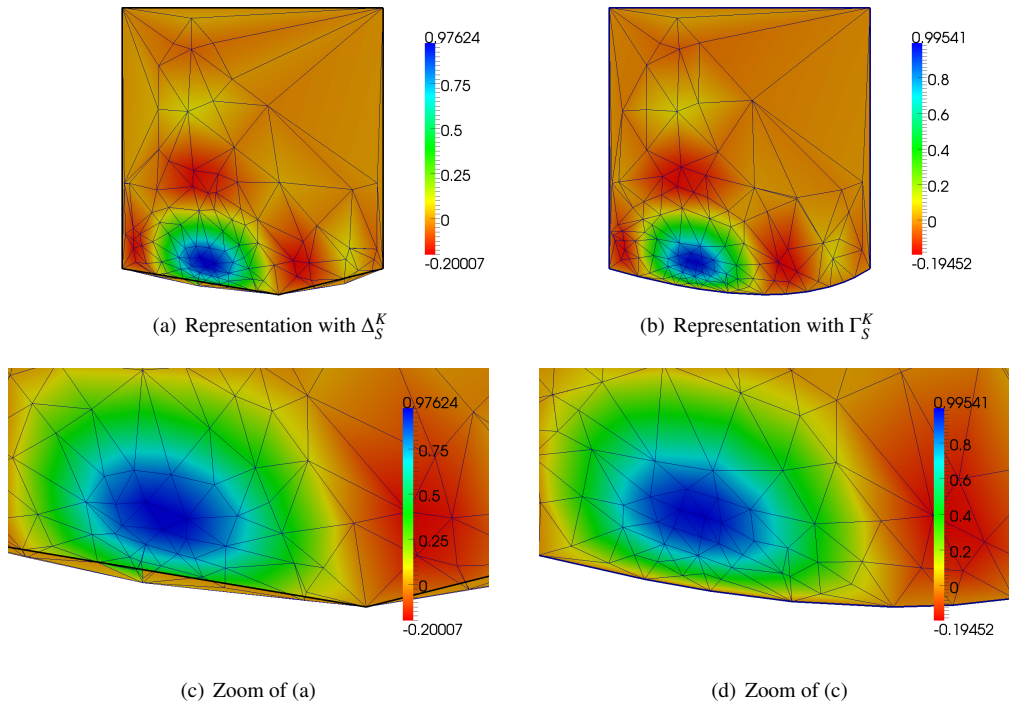


Figure 9: Representation at 3.5% with the two estimates Δ_S^K (a and c) and Γ_S^K (b and d) of a Gauss-Lobatto basis function Q^6 on quadrangle Q^2

expected, the use of the adaptive method allows, for a given tolerance, to have much less elements than with a regular refinement. Indeed, a uniform remeshing generally implies many useless subdivisions due to the different scales of the function variations. This is particularly true when the order of the function (or the element) increases. Hence, in the second case, the differences between adaptive and regular are even greater.

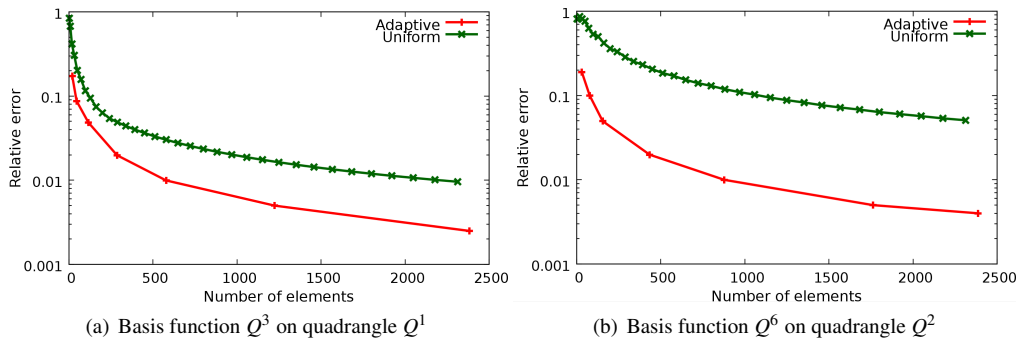


Figure 10: Comparison of the accuracy given from adaptive and uniform methods for two Gauss-Lobatto basis function

6.2. Acoustic wave equation

The second numerical example focuses on the propagation of an acoustic wave. We are treating a displacement-based transient acoustic equation. Thus, the system we are going to solve is the following:

$$\begin{cases} \rho(x) \frac{\partial^2 u}{\partial t^2}(x, t) - \nabla \cdot (\mu(x) \nabla \cdot u(x, t)) = f(x, t), & \text{in } \Omega \times [0, T] \\ u(x, t) \cdot n = 0, & \text{on } \Gamma \times [0, T] \\ u(x, 0) = 0, & \text{in } \Omega \\ \frac{\partial u}{\partial t}(x, 0) = 0, & \text{in } \Omega \end{cases} \quad (13)$$

where Ω is the bounded domain, Γ its boundary, n the outward normal and T the final time. Classical notations are introduced: ∇ denotes the gradient and $\nabla \cdot$ the divergence. The displacement vector is noted u , ρ and μ are parameters which depend on the medium they belong to. The problem is solved by finite elements method. Raviart-Thomas elements are used (see hereafter) on an hybrid mesh: curved triangles, of order 2 (ie. defined from 6 points), are used to have a good approximation of the geometry, and quadrangles are used where a Cartesian mesh is possible. The displacement field is approximated in the following finite element space

$$V_h^k = \{v_h \in H(\text{div}, \Omega) : \forall K \in \mathcal{T}(X), \mathcal{P}_K^{-1} v_{h_K} \circ g_K \in RT^k(\widehat{K})\}$$

where k is the order of the Raviart-Thomas elements used, \mathcal{P}_K denotes the Piola transformation [30], $\mathcal{P}_K = \frac{J_{g_K}}{|J_{g_K}|}$ with J_{g_K} the jacobian matrix of the geometrical transformation and $|J_{g_K}|$ its determinant. Depending on the nature of the elements, two cases have to be considered. For the triangles, $g_K \in P^2(\widehat{K})^2$, the triangles are thus defined from 6 nodes and $RT^k(\widehat{K}) = P^k(\widehat{K})^2 + xP^k(\widehat{K})$ as in [30]. For the quadrangles, the reference cell \widehat{K} is the unit square, $g_K \in Q^1(\widehat{K})^2$, $RT^k(\widehat{K}) = Q^{k+1,k}(\widehat{K}) \times Q^{k,k+1}(\widehat{K})$ as in [3] where $Q^{l,m}(\widehat{K}) = P^l([0, 1]) \otimes P^m([0, 1])$.

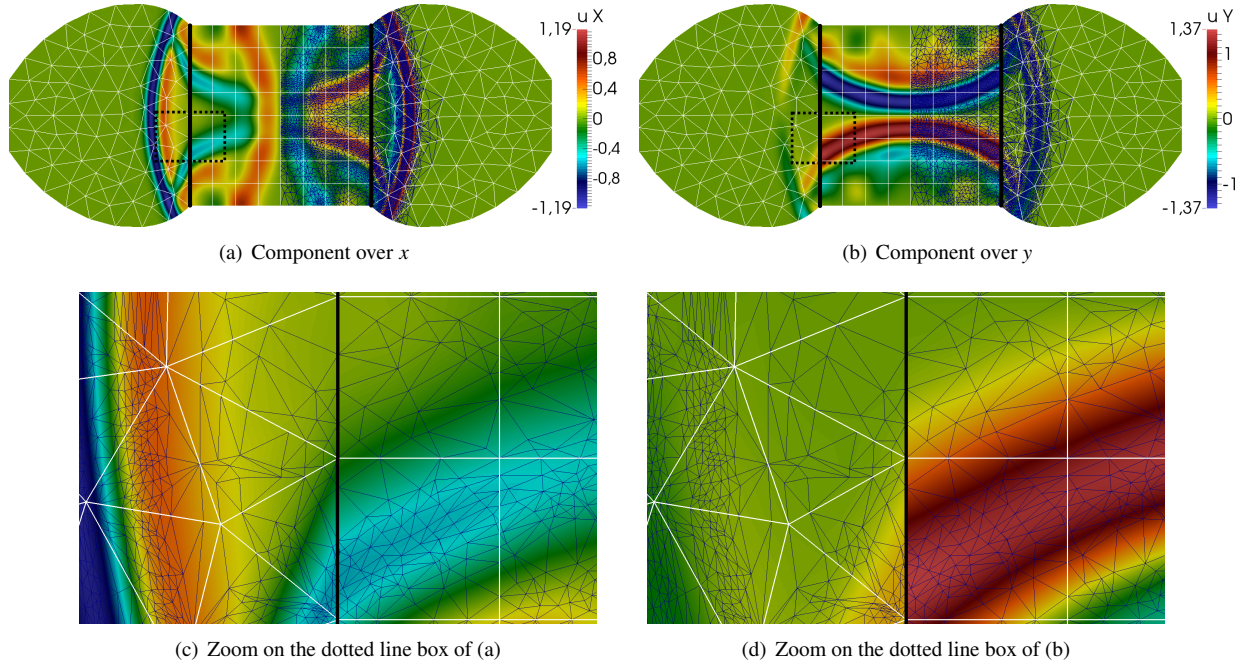


Figure 11: Representation of the solution of the propagation of an acoustic wave at a final time $T=5$

The computational mesh used for figure 11 is composed of 64 quadrangles and 306 triangles, the order chosen is $k = 5$. The domain is constituted of two medium: one medium is composed of triangles, the second of quadrangles. For the triangles, the parameters are constant: $\rho_1 = 4$ and $\mu_1 = 1$. For the quadrangles, $\rho_2 = 1$ and $\mu_2 = 1$. As $\rho_1 \neq \rho_2$, there is a physical discontinuity due to a material change at the interface between the triangles and the quadrangles.

Two black straight lines show this interface in figure 11. As this interface is parallel to y -axis, the x -component of the solution should be continuous whereas the y -component of the solution should be discontinuous. Note that a unique representation mesh is built: each component of the solution has therefore the same representation mesh. Figures 11.c and 11.d are zooms of an area of interest. Each edge (between two elements) has a unique representation mesh. It allows to represent currently the physical gaps (y -component) and the continuities (x -component), as asked by objective (O_3).

6.3. Elasticity problem solved by discontinuous Galerkin method

The last example is devoted to the propagation of an elastic wave in a 3 layered medium computed by a Discontinuous Galerkin method [31] constructed similarly than in part 6.1 but by using Gauss points instead of Gauss Lobatto points. The order of the basis functions used is 7. The computational mesh is a Cartesian mesh of 24×24 cells. Figure 12 shows some representations of the numerical solution at the final time. Figure 12.a shows the representation given by the adaptive method under the tolerance 0.5 %. The representation mesh (white) is plotted, 52108 elements constitute the representation, which involves an average of 90 elements by cell. A zoom on the dotted line box is done in figure 12.c. As DG method was used to solve the problem, one sees gaps between each elements of the computational mesh (see figure 12.c). As asked by objective (O_3), numerical jumps are well-represented. Figure 12.b is a similar zoom but an uniform remeshing of 7×7 (corresponding to 98 triangles) of each computational cell is done to obtain this representation. This subdivision corresponds to an usual remeshing where the number of representation points is equal to the number of degrees of freedom (ie. 64). The rendering obtained is not accurate, the relative error is about 10.5 % in L^∞ -norm. Furthermore, any post-processing (isolines, cuts, streamlines,...) done from such a "bad" representation will obviously lead to an incorrect result. For instance, ten isovalues are plotted from the two representations (see figures 12.b and 12.c), we clearly see that the isovalues coming from the uniform refinement are inaccurate.

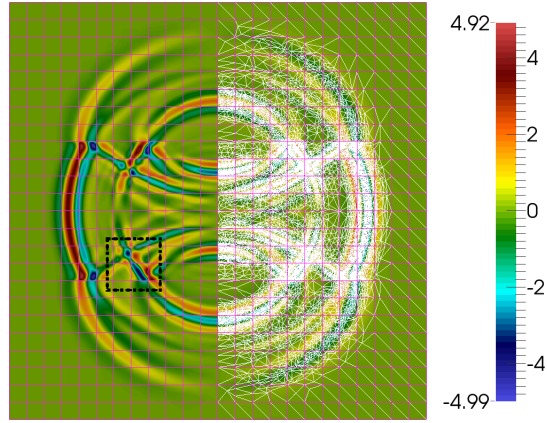
We now focus on one cell of the computational mesh, noted C , which lies in the middle of the dotted line box and is particularly interesting as it contains many information, due to a change of the characteristics of the elastic medium. Figure 12.d is a zoom on this cell given by the uniform 7×7 approach, figure 12.e (resp. 12.f) gives a comparison with the adaptive method for the same accuracy of 10 % (resp. the same global number of elements). Table 1 sums up some of these information. It gives a comparison of the number of elements of the representation mesh created for several cases. The global number of elements to have 10% (resp. 1 %) accuracy is 25 (resp. 33) times bigger with the uniform subdivision than the adaptive method.

Method	Local (cell C)		Global (576 cells)	
	Number of elements	Accuracy	Number of elements	Accuracy
Uniform 7×7	98	10,4 %	56448 (98/cell)	10,5 %
Adaptive (10)	23	9.93 %	2194 (\approx 4/cell)	9.93 %
Uniform 27×27	1458	1.03 %	839808 (1458/cell)	1.03 %
Adaptive (1)	416	1.00 %	24856 (\approx 43/cell)	1.00 %

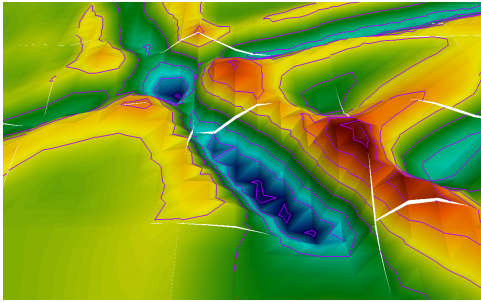
Table 1: Comparison of the number of elements of the representation mesh with the uniform subdivision and the adaptive methods for several accuracies on one cell of the computational mesh (local, cell C) and on the whole computation mesh (global)

7. Conclusion

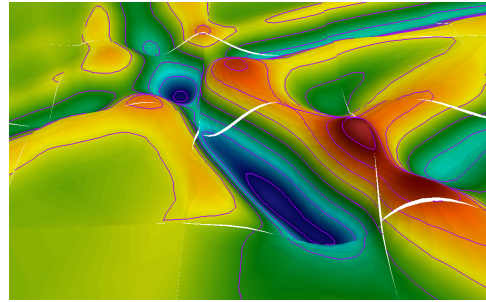
An auto-adaptive method for the visualization of high-order methods is presented. Our approach is based on the construction of an optimized P_1 approximation of the hp solution. A key point is that we want to control the error between the solution and its representation in L^∞ -norm. The rendering of discontinuities is another point of interest. These discontinuities can either comes from the physical problem (material change) or the numerical method (DG). A decomposition of the boundary of the elements in edges is introduced to handle these discontinuities. A representation mesh is constructed element by element from the mesh of the boundary. The refinement of this mesh is guided by an *a posteriori* estimate which control Hausdorff distance, the latter being introduced to evaluate pointwise the error



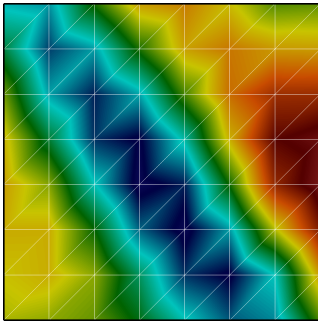
(a) Representation of the numerical solution with the adaptive method (0.5 %)



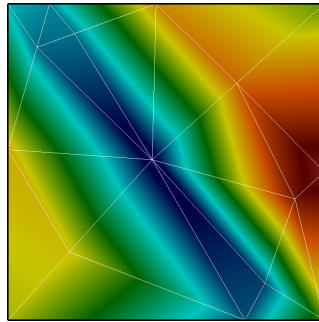
(b) Zoom on the dotted line box, uniform 7×7 , 56448 elements



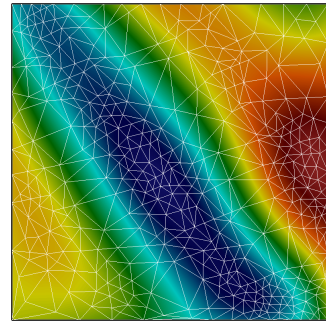
(c) Zoom on the dotted line box of (a), adaptive (0.5 %), 52108 elements



(d) Focus on the central cell of the dotted line box, noted C , uniform 7×7 , 98 elements



(e) Focus on cell C , adaptive 10 %, 23 elements



(f) Focus on cell C , adaptive 0.5 %, 818 elements

Figure 12: Propagation of an elastic wave in a 3 layered medium computed by Discontinuous Galerkin (DG) method Q^7 on a 24×24 uniform mesh

between the numerical solution and its representation. In the case of curved elements, a specific *a posteriori* estimate is defined to ensure the convergence of the algorithm. Finally, a theorem proves that the representation constructed following this algorithm is "well-suited". The framework is general for any standard visualization software as the data are transformed into linear functions. Several numerical examples show the relevance of the methodology proposed. This non-uniform refinement process allows to limit the amount of data generated. The pointwise control avoid to lose information/phenomenon. This paper focused on the rendering of surfaces but the volume rendering could be straightforwardly treated by the algorithm we propose.

References

- [1] O. C. Zienkiewicz, R. L. Taylor, The finite element method, Vol. 3, McGraw-hill London, 1977.
- [2] B. Cockburn, G. E. Karniadakis, C.-W. Shu, The development of discontinuous galerkin methods, in: *Discontinuous Galerkin Methods*, Springer, 2000, pp. 3–50.
- [3] G. Cohen, S. Pernet, *Finite Elements and Discontinuous Galerkin Methods for Transient Wave Equations*, Springer, 2017.
- [4] G. Karniadakis, S. Sherwin, *Spectral/hp element methods for computational fluid dynamics*, Oxford University Press, 2013.
- [5] T. J. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer methods in applied mechanics and engineering* 194 (39) (2005) 4135–4195.
- [6] A. Bartezzaghi, L. Dedè, A. Quarteroni, Isogeometric analysis of high order partial differential equations on surfaces, *Computer Methods in Applied Mechanics and Engineering* 295 (2015) 446–469.
- [7] Y. Zhou, M. Garland, R. Haber, Pixel-exact rendering of spacetime finite element solutions, in: *Proceedings of the Conference on Visualization'04*, IEEE Computer Society, 2004, pp. 425–432.
- [8] A. Stahl, T. Kvamsdal, C. Schellewald, Post-processing and visualization techniques for isogeometric analysis results, *Computer Methods in Applied Mechanics and Engineering* 316 (2017) 880–943.
- [9] B. Nelson, E. Liu, R. M. Kirby, R. Haimes, ElVis: A system for the accurate and interactive visualization of high-order finite element solutions, *IEEE transactions on visualization and computer graphics* 18 (12) (2012) 2325–2334.
- [10] J.-F. Remacle, N. Chevaugeon, E. Marchandise, C. Geuzaine, Efficient visualization of high-order finite elements, *International Journal for Numerical Methods in Engineering* 69 (4) (2007) 750–771.
- [11] J.-F. Remacle, C. Geuzaine, G. Compère, B. Helenbrook, Adaptive mesh generation and visualization, *Encyclopedia of Aerospace Engineering*.
- [12] W. J. Schroeder, F. Bertel, M. Malaterre, D. Thompson, P. P. Pebay, R. O'Bara, S. Tendulkar, Methods and framework for visualizing higher-order finite elements, *IEEE Transactions on Visualization and Computer Graphics* 12 (4) (2006) 446–460.
- [13] T. Lan, S. Lo, Finite element mesh generation over analytical curved surfaces, *Computers & Structures* 59 (2) (1996) 301–309.
- [14] H. Chen, J. Bishop, Delaunay triangulation for curved surfaces, *Meshing Roundtable* (1997) 115–127.
- [15] R. Löhner, P. Parikh, Generation of three-dimensional unstructured grids by the advancing-front method, *International Journal for Numerical Methods in Fluids* 8 (10) (1988) 1135–1149.
- [16] R. B. Simpson, Anisotropic mesh transformations and optimal error control, *Applied Numerical Mathematics* 14 (1) (1994) 183–198.
- [17] L. Chen, P. Sun, J. Xu, Optimal anisotropic meshes for minimizing interpolation errors in L^p -norm, *Mathematics of Computation* 76 (257) (2007) 179–204.
- [18] A. Loseille, F. Alauzet, Continuous mesh framework part I: well-posed continuous interpolation error, *SIAM Journal on Numerical Analysis* 49 (1) (2011) 38–60.
- [19] D. Walfisch, Visualization for high-order discontinuous Galerkin CFD results, Master's thesis, Massachusetts Institute of Technology. Dept. of Aeronautics and Astronautics. (2007).
- [20] D. R. Jones, C. D. Perttunen, B. E. Stuckman, Lipschitzian optimization without the Lipschitz constant, *Journal of optimization Theory and Applications* 79 (1) (1993) 157–181.
- [21] J. Clausen, A. Žilinskas, Subdivision, sampling, and initialization strategies for simplicial branch and bound in global optimization, *Computers & Mathematics with Applications* 44 (7) (2002) 943–955.
- [22] A. Žilinskas, J. Žilinskas, Global optimization based on a statistical model and simplicial partitioning, *Computers & Mathematics with Applications* 44 (7) (2002) 957–967.
- [23] A. Bowyer, Computing Dirichlet tessellations, *The computer journal* 24 (2) (1981) 162–166.
- [24] D. F. Watson, Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes, *The computer journal* 24 (2) (1981) 167–172.
- [25] S. Rebay, Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm, *Journal of computational physics* 106 (1) (1993) 125–138.
- [26] U. Ayachit, The paraview guide: a parallel visualization application.
- [27] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, D. Camp, O. Rübel, M. Durant, J. M. Favre, P. Navrátil, VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data, in: *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, 2012, pp. 357–372.
- [28] A. O. Leone, P. Marzano, E. Gobetti, R. Scateni, S. Pedinotti, Discontinuous finite element visualization, in: *Proceedings 8th International Symposium on Flow Visualization*, 1998.
- [29] M. Abramowitz, I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, Vol. 55, Courier Corporation, 1964.
- [30] V. Ervin, Computational bases for RT_k and BDM_k on triangles, *Computers & Mathematics with Applications* 64 (8) (2012) 2765–2774.

- [31] Y. Dudouit, Spatio-temporal refinement using a discontinuous Galerkin approach for elastodynamic in a high performance computing framework, Ph.D. thesis, Bordeaux (2014).