Adaptive Post-Processing Method to Represent High-Order Numerical Solutions

Vincent Mouysset, Sébastien Pernet

9-11 March 2016

ONERA

THE FRENCH AEROSPACE LAB

hp methods

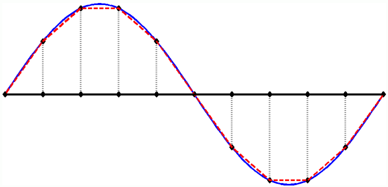


Figure : P^1 interpolation for $\Delta x = \lambda/10$.

hp methods

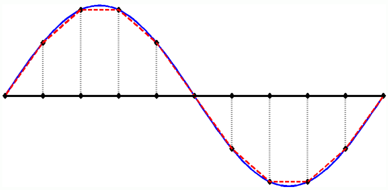


Figure : P^1 interpolation for $\Delta x = \lambda/10$.

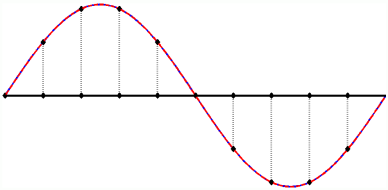


Figure : P^2 interpolation for $\Delta x = \lambda/10$.

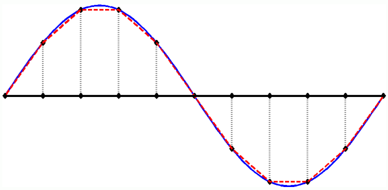


Figure : P^1 interpolation for $\Delta x = \lambda/10$.

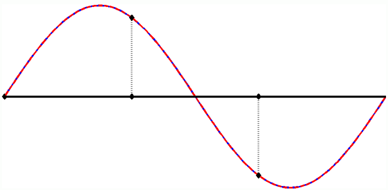


Figure : P^2 interpolation for $\Delta x = \lambda/3$.

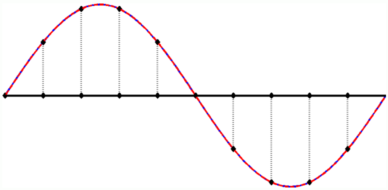


Figure : P^2 interpolation for $\Delta x = \lambda/10$.

hp methods

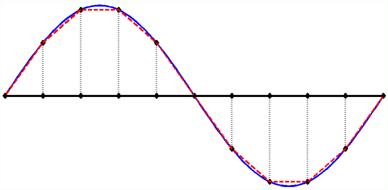


Figure : P^1 interpolation for $\Delta x = \lambda/10$.

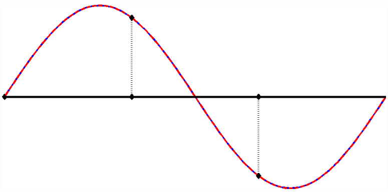


Figure : P^2 interpolation for $\Delta x = \lambda/3$.

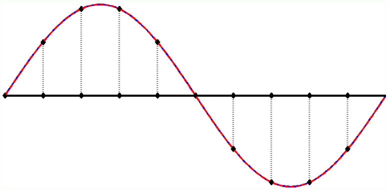


Figure : P^2 interpolation for $\Delta x = \lambda/10$.

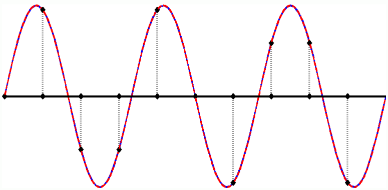


Figure : P^2 interpolation on a 3λ domain.

What do we call *hp* methods?

- **These methods are characterized by:** (one or both)
  - sophisticated functions: high order polynomials, eigen or special functions, approaches of collocation or modal type,

    $\Rightarrow$ the value at each point is given by a function which is not linear.
  - mesh cells different from simplexes: quadrangle, hexahedra, isogeometric elements...

    $\Rightarrow$ representation of the element is not linear.

- ***hp* numerical methods have a great potential:**
  - Very accurate solutions which can contain lots of physical informations in each cell.
  - They allow to decrease the computational costs.

- Numerous declinations in literature and industrial codes:
  - FEM, DG, high-order FV and FD,
  - isoparametric methods,
  - high-order BEM,...

Issue on the exploitation of *hp* solutions

- What do we want to do with a *hp* solution?
  - Depict it.
  - Extract some informations (pointwise values, isolines, slices, gradient,...).

- How?
  - With dedicated subroutines in the computational code,

    $\Rightarrow$ consumes expensive execution time on servers (cpu efforts and hdd accesses).
  - By mean of a given visualization software (GMSH, PARAVIEW, TECPLOT,...),

    $\Rightarrow$ post-treatment is led apart from calculus (on different computers).

  However
    - input formats are not necessarily suited to the considered *hp* element,
    - common format is "low precision": nodal values on simplexes or other cells,
    - there are some "open formats" (GMSH...), transformation to display informations is automated.

- To summarize, two main options are:
  - adapt to what exists $\rightarrow$ we make interpolations/projections to write the *hp*-solution in the chosen format,
  - (when available) use an "open format" allowing to describe the *hp* solution $\rightarrow$ the visualization software controls itseft the interpolations/projections.

Classical approach to represent a *hp* solution

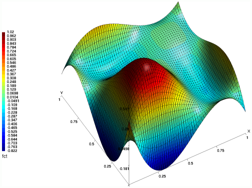


Figure : function to depict: a Gauss-Legendre basis function

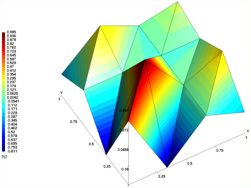


Figure : Usual depiction: 1 point by dof

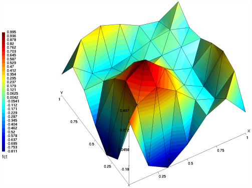
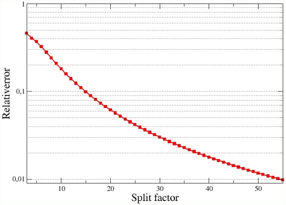


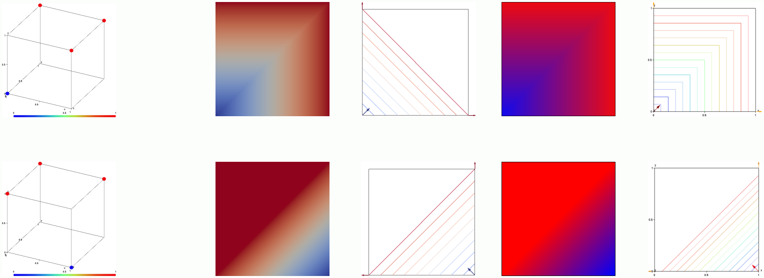
Figure : Refined depiction: 2 points by dof (4 times more elements)



Some practical questions:

- How many subdivisions to perform for a given accuracy?
- How many data will be generated?
- Is this representation giving correct analysis tools to interpret (physically) the *hp* results?

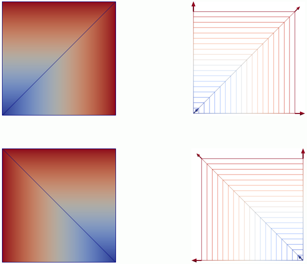
Configurations | Software P. | Software G.



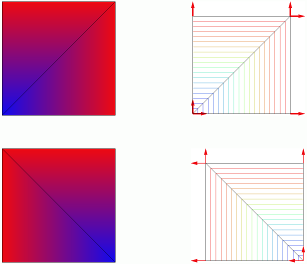
Summary:

- Same function but representations not identical (up to a rotation by 90°)!
- Representations seem affine on simplexes (but data is given on a quadrangle).
- Split of the cell into two triangles independent of the function to represent?
- Isolines and gradients follow or not the depiction.

Software P.



Software G.



Input format = linear on simplexes ⇒ **consistent representations and extractions!**

Formalization

Linear representation suited to the *hp* solutions

Construction of the visualization

Examples

- Example 1: what happens on basis functions?
- Example 2: what happens when combining basis functions?
- Example 3: what happens on more realistic simulations?

Perspectives

Aims

- **Our point of view:** usual exploitation of *hp* solutions are not optimal as
  - the balance accuracy-cost (number of data) required to have a good rendering,
  - the difficulty to give some *a priori* (number of subdivisions, target error) which ensure the quality of rendering.

- **Our aims:** better exploitation (by a classical visualization software) of data produced by *hp* numerical simulations

  $\leftarrow$ provide a reliable information through the representations,

  $\leftarrow$ developments compatible with different *hp* codes,

  $\leftarrow$ reliable extraction of quantities of interest straightforwardly with the visualization software.

Formalisation of objectives

Our approach to define a well-suited visualization for *hp* solutions is summarized into 4 objectives:

(O_1): The representation f_{vis} 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): Error between f_{num} and its representation is controlled in L^∞-norm,

(O_3): At the prescribed tolerance, the representation shows gaps if and only if f_{num} has.

$+(O_4)$: Specific control to be defined according to extractions realized from f_{vis}.

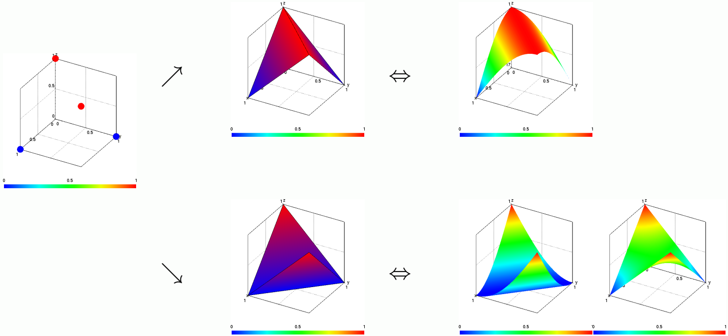


Figure : Interpretation with respect to Q^1-consistent splitting.

O_1 Representation by piecewise linear functions on k-simplexes,

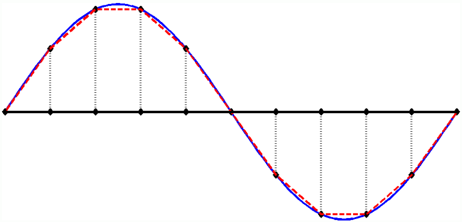


Figure : P^1-approximation, $\Delta x = \lambda/10$.

$\rightarrow$ fine L2 candidate but associated "colorbar" not matching on expected values.

O_1 Representation by piecewise linear functions on k-simplexes,

O_2 Error control in L^∞ norm,

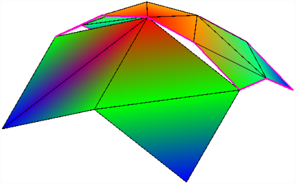


Figure : Representation of a continuous function on non-coincident meshes.

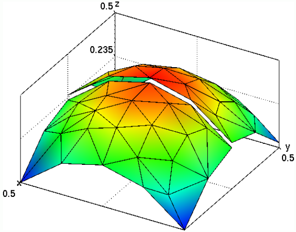
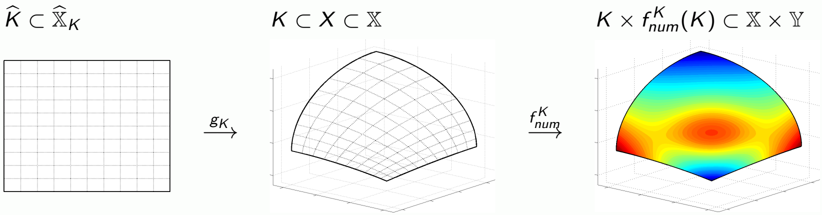


Figure : Representation of a function with jump on coincident meshes.

- O_1 Representation by piecewise linear functions on k-simplexes,
- O_2 Error control in L^∞ norm,
- O_3 Representation of jump if and only if it exists.

Definition of a *hp* solution



Ingredients of the numerical solution f_{num}:

→ **mesh:** $\mathcal{T}(X)$ is a mesh of X,

→ **reference cell:** $\forall K \in \mathcal{T}(X)$, $g_K : \widehat{K} \to K$ bijection,

→ **basis functions:** $\forall K \in \mathcal{T}(X), \forall i = 1 \dots N_K$, $\varphi_i^K : \widehat{K} \to Y$ (at least continuous).

Then, for each $K \in \mathcal{T}(X)$, the definition of f_{num} on K, noted f_{num}^K, is expressed by means of coefficients (degrees of freedom) f_i^K via the following decomposition

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

Formalization

Linear representation suited to the *hp* solutions

Construction of the visualization

Examples

- Example 1: what happens on basis functions?
- Example 2: what happens when combining basis functions?
- Example 3: what happens on more realistic simulations?

Perspectives

Ingredients of the representation function f_{vis}:

- f_{num} has a local definition on $\mathcal{T}(X)$ so f_{vis} will be, (f_{vis}^K denotes its local representation on $K \in \mathcal{T}(X)$),
- Objective O_1: f_{vis}^K is a linear function on simplexes (when f_{num}^K is more sophisticated) so we introduce
  - $\rightarrow$ **(sub)-mesh:** a mesh made of simplexes of K, noted $\mathcal{T}(K)$,
  - $\rightarrow$ **basis functions:** for any $S \in \mathcal{T}(K)$ let $\mathbb{P}^1(S)$ be the space of polynomial of total degree less than or equal to 1.

  Thus, f_{vis}^K will verify

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

Remark: f_{vis}^K is defined on the physical cell K (the one to be plotted) when f_{num}^K is on the reference one $\widehat{K}$.

Definition of f_{vis} : construction of the representation mesh

f^K_{num} is evaluated from reference cell $\widehat{K}$ so $\mathcal{T}(K)$ will be constructed from a mesh composed of simplexes $\mathcal{T}(\widehat{K})$ of $\widehat{K}$.

The construction of $\mathcal{T}(K)$ is then performed in the following way:

1. For $K \in \mathcal{T}(X)$, we define the topology of $\mathcal{T}(\widehat{K})$ by

$$\widehat{\mathcal{N}} := \{\widehat{N} := (\widehat{N}_i)_{i=1,\dots,n_K+1} \; : \; \Lambda(\widehat{N}) \in \mathcal{T}(\widehat{K})\}$$

   where Λ associates $n+1$ points $P := (P_i)_{i=1,\cdots,n+1}$ to the n-simplex $\Lambda(P)$.

2. We define the set of nodes of $\mathcal{T}(K)$ as well as its topology by

$$\mathcal{N} := \{N = (N_i)_{i=1,\dots,n_K+1} := g_K(\widehat{N}) \; : \; \widehat{N} \in \widehat{\mathcal{N}}\}$$

   with $g_K(\widehat{N}) := (g_K(\widehat{N}_i))_{i=1,\dots,n_K+1}$ où $\widehat{N} := (\widehat{N}_i)_{i=1,\dots,n_K+1}$.

3. The mesh in simplexes of K is defined by

$$\mathcal{T}(K) := \{\Lambda(N) \; : \; N \in \mathcal{N}\}.$$

Remark: one can have $\widetilde{K} := \cup_{S \in \mathcal{T}(K)} S \quad \neq K$ and $f^K_{vis}(\widetilde{K}) \neq f^K_{num}(K)$

Lemma (identification of $\mathcal{T}(K)$)

For all $K \in \mathcal{T}(X)$, the $\mathbb{P}^1$ interpolation of g_K constructed from $\mathcal{T}(\widehat{K})$, noted $P^1 g_K$, is

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

Remark : This construction does not ensure the injectivity of $P^1 g_K$ from $\widehat{K}$ onto $\widetilde{K}$.

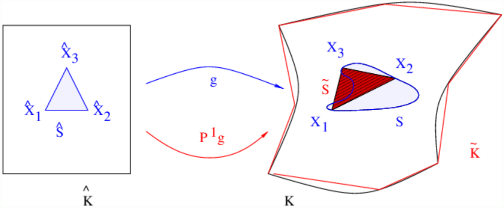


Figure : possible loss of injectivity in the construction of $\mathcal{T}(\widehat{K})$.

The function f_{vis}^K is then defined on each simplex $\Lambda(N)$ of $\mathcal{T}(K)$ by:

1. for each node $N_i \in N$, $f_{vis}^K(N_i) := f_{num}^K(N_i)$,
2. f_{vis}^K is an affine function on $\Lambda(N)$ and is defined by

$$x = \sum_{i=1}^{n_K+1} x_i N_i \in \Lambda(N) \longmapsto f_{vis}^K(x) = \sum_{i=1}^{n_K+1} x_i f_{num}^K(N_i),$$

where $(x_i)_{i=1,\dots,n_K+1} \in [0,1]^{n_K+1}$.

$\Rightarrow$ f_{vis} **fulfils the objective** O_1**.**

Definition of f_{vis}

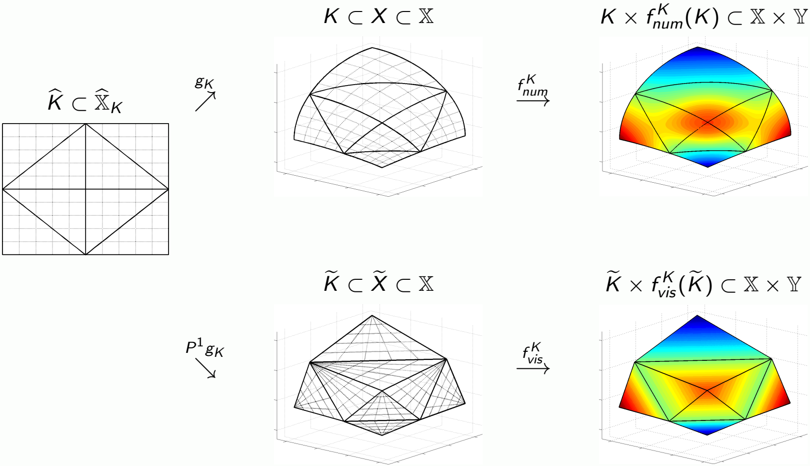


Figure : Construction of the representation f_{vis} of the *hp* solution f_{num}.

Formalization

Linear representation suited to the *hp* solutions

Construction of the visualization

Examples
- Example 1: what happens on basis functions?
- Example 2: what happens when combining basis functions?
- Example 3: what happens on more realistic simulations?

Perspectives

Estimate of the visualization error

- f_{num} and f_{vis} have a local definition on each $K \in \mathcal{T}(X) \Rightarrow$ local estimate
- Difficulty : supports are not coinciding in the physical space X $(K \neq \widetilde{K})$.
- Solution : the Hausdorff distance (convergence of graphs)

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

where d is a distance on $X \times Y$, with α and β be positive parameters:

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

- But, it is too expensive to be calculated.
- Idea: "localization" (inside K)!

Proposition (local *a posteriori* estimate)

Let $K \in \mathcal{T}(X)$. The following estimate holds:

$$d_H\left(f^K_{num}, f^K_{vis}\right) \leq \delta_{\mathcal{T}(K)}\left(f^K_{num}, f^K_{vis}\right),$$

where d_H is the Hausdorff distance and

$$\begin{aligned}\delta_{\mathcal{T}(K)} :& (f, \widetilde{f}) \in C^0(K, \mathbb{Y}) \times C^0(\widetilde{K}, \mathbb{Y}) \\ & \mapsto \sup_{\widehat{x} \in \widehat{K}} \max\left(\alpha \left\|g(\widehat{x}) - P^1 g(\widehat{x})\right\|_{\mathbb{X}}, \beta \left\|(f \circ g)(\widehat{x}) - \left(\widetilde{f} \circ P^1 g\right)(\widehat{x})\right\|_{\mathbb{Y}}\right),\end{aligned}$$

with α and β be dimensioning constants.

Lemma (Generic fulfilment of objectives)

If for all cell $K \in \mathcal{T}(X)$, one has

- $\delta_{\mathcal{T}(K)}\left(f^K_{num}, f^K_{vis}\right) \leq \varepsilon$,
- $\forall K' \in \mathcal{T}(X) \,:\, F := \partial K \cap \partial K' \neq \emptyset$, *meshes* $\mathcal{T}(K)$ *and* $\mathcal{T}(K')$ *lead to*

$$\mathcal{T}(F) = \mathcal{T}(F')$$

  where $\mathcal{T}(F)$ *and* $\mathcal{T}'(F)$ *are the meshes in simplexes of* F *obtained by restriction of those of* K *and* K', *respectively.*

Then the representation f_{vis} *of* f_{num} *fulfils the objectives* (O_1), (O_2) *and* (O_3).

$\rightarrow$ we need to grant $\mathcal{T}(F) = \mathcal{T}(F')$!

Fulfilment of the objective O_3: decomposition in lower dimensions

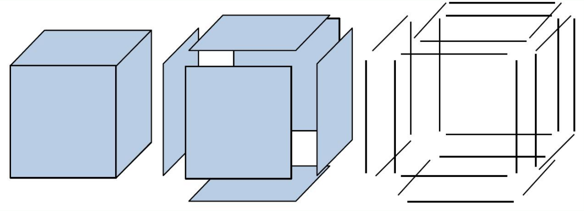


Figure : $\widetilde{\mathcal{T}}_{3\,D}(X)$ (left), $\widetilde{\mathcal{T}}_{2\,D}(X)$ (center) et $\widetilde{\mathcal{T}}_{1\,D}(X)$ (right)

One decomposes $\mathcal{T}(X)$ into sets of elements of lower dimension $\widetilde{\mathcal{T}}_{i\,D}(X)$ for $i = 1, \dots, 3$:

$$\widetilde{\mathcal{T}}_{3\,D}(X) := \mathcal{T}_{3\,D}(X)$$

$$\widetilde{\mathcal{T}}_{2\,D}(X) := \mathcal{T}_{2\,D}(X) \cup \left(\bigcup_{K \in \mathcal{T}(X)\,:\,dim(K)=3} \mathcal{F}(K) \right)$$

$$\widetilde{\mathcal{T}}_{1\,D}(X) := \mathcal{T}_{1\,D}(X) \cup \left(\bigcup_{K \in \mathcal{T}(X)\,:\,dim(K)=2} \mathcal{E}(K) \right) \cup \left(\bigcup_{K \in \mathcal{T}(X)\,:\,dim(K)=3} \mathcal{E}(K) \right)$$

where $\mathcal{F}(K)$ and $\mathcal{E}(K)$ are the faces and the edges of K respectively.

Fulfilment of the objective O_3: decomposition in lower dimensions

Proposition (fulfilment of O_3)

Let $(\mathcal{T}(\Sigma))_{\Sigma \in \widetilde{\mathcal{T}}_{k\,D}(X), k=1,2,3}$ be a set of meshes in simplexes satisfying: for $k = 2, 3$, $\forall \Sigma \in \widetilde{\mathcal{T}}_{k\,D}(X)$,

$$\mathcal{T}(\Sigma)_{|\partial\widetilde{\Sigma}} = \bigcup_{F \in \mathcal{F}(\Sigma)} \mathcal{T}(F) \text{ avec } \widetilde{\Sigma} = \bigcup_{S \in \mathcal{T}(\Sigma)} S \tag{2}$$

then the representation f_{vis} of f_{num} constructed from $\mathcal{T}(K)$ for $K \in \mathcal{T}_{kD}(X)$ fulfils the objective 0_3

This proposition provides a "simple" algorithm to ensure the objective O_3:

1. construction of representation meshes of elements of $\widetilde{\mathcal{T}}_{1\,D}(X)$,
2. construction of representation meshes of elements of $\widetilde{\mathcal{T}}_{2\,D}(X)$ from those of $\widetilde{\mathcal{T}}_{1\,D}(X)$ and satisfying (2),
3. construction of representation meshes of elements of $\widetilde{\mathcal{T}}_{3\,D}(X)$ from those of $\widetilde{\mathcal{T}}_{2\,D}(X)$ and satisfying (2).

First convergence result

Rough description of the algorithm:

1. decomposition in lower dimensions $\rightarrow$ functions and traces are connected to corresponding element,
2. meshing of all 1D cells in $\widetilde{\mathcal{T}}_{1\,D}(X)$ such that
$$\max_{f_{1D}^{num} \text{ connected to the element}} \delta(f_{num}^{1D}, f_{vis}^{1D}) \leq \varepsilon,$$
3. interior meshing of all 2D cells in $\widetilde{\mathcal{T}}_{2\,D}(X)$ - boundaries meshed before - such that
$$\max_{f_{2D}^{num} \text{ connected to the element}} \delta(f_{num}^{2D}, f_{vis}^{2D}) \leq \varepsilon,$$
4. same process for 3D cells.

Proposition (Convergence via meshing with respect to dimensions)

If the algorithm of construction implies:

1. *For all $K \in \mathcal{T}(X)$, the P^1 interpolation of g_K constructed from $\mathcal{T}(\widehat{K})$ is an injective function from $\widehat{K}$ onto $\widetilde{K}$,*
2. *The convergence toward the target error ε is achieved.*

Then the representation f_{vis} of f_{num} fulfils the objectives (O_1), (O_2) and (O_3).

Formalization

Linear representation suited to the *hp* solutions

Construction of the visualization

Examples

- Example 1: what happens on basis functions?
- Example 2: what happens when combining basis functions?
- Example 3: what happens on more realistic simulations?

Perspectives

Q^3 Gauss-Legendre basis function

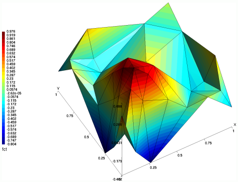


Figure : Adaptive remeshing $\varepsilon = 10\%$

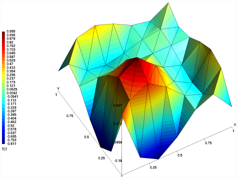


Figure : Uniform 6×6 (same number of elements than adaptive 10%)

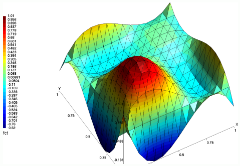


Figure : Uniform 14×14 (same accuracy than adaptive 10%)

Example 1: what happens on basis functions?

Q^3 Gauss-Legendre basis function

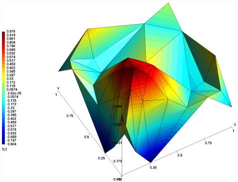


Figure : Adaptive remeshing $\varepsilon = 10\%$

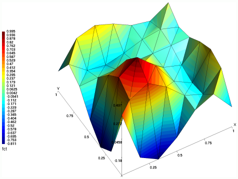


Figure : Uniform 6×6 (same number of elements than adaptive 10%)

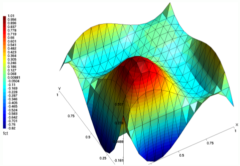


Figure : Uniform 14×14 (same accuracy than adaptive 10%)

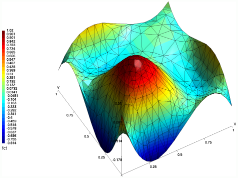


Figure : Adaptive remeshing $\varepsilon = 1\%$

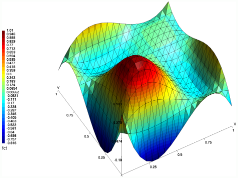


Figure : Uniform 18×18 (same number of elements than adaptive 1%)

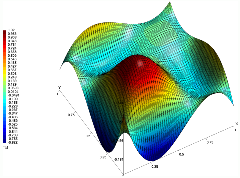
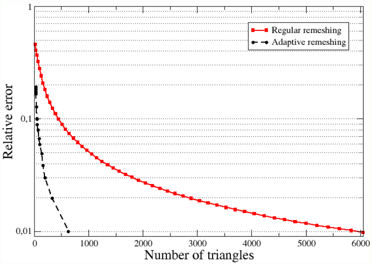


Figure : Uniform 54×54 (same accuracy than adaptive 1%)

Example 1: what happens on basis functions?

Q^3 Gauss-Legendre basis function

Meshing	number triangles	relative error
Adaptive 10%	56	9.91%
Uniform 6 × 6	72	32.27%
Uniform 14 × 14	392	11.0%
Adaptive 5%	138	4.88%
Uniform 9 × 9	162	20.77%
Uniform 22 × 22	968	5.22%
Adaptive 1%	630	1.00%
Uniform 18 × 18	648	7.34%
Uniform 54 × 54	5832	1.01%



Formalization

Linear representation suited to the *hp* solutions

Construction of the visualization

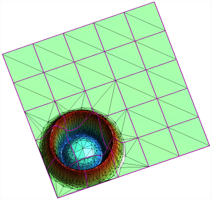
Examples

- Example 1: what happens on basis functions?
- Example 2: what happens when combining basis functions?
- Example 3: what happens on more realistic simulations?

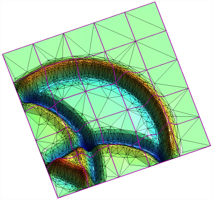
Perspectives

Example 2: what happens when combining basis functions?

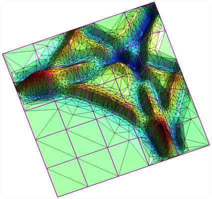
Simulation of the propagation of a source point by a DG method



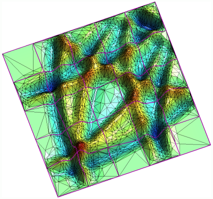
(a) time 1



(b) time 2



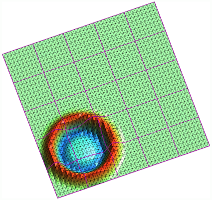
(c) time 3



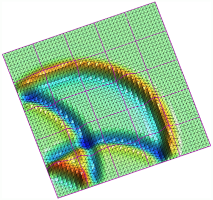
(d) time 4

Figure : Propagation of a source point in a cavity by DG Q^{10} on 5×5 uniform mesh : adaptive remeshing 1%

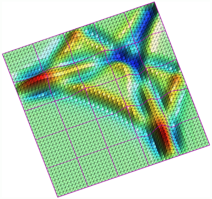
Simulation of the propagation of a source point by a DG method



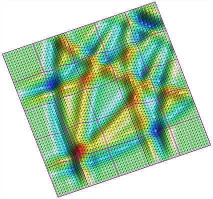
(a) Uniform 8 × 8: time 1



(b) Uniform 9 × 9: time 2



(c) Uniform 9 × 9: time 3

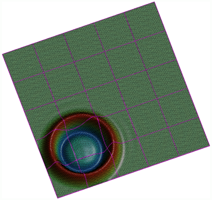


(d) Uniform 11 × 11: time 4

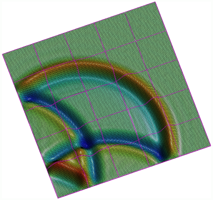
Figure : Propagation of a source point in a cavity by DG Q^{10} on 5 × 5 uniform mesh : uniform remeshing with same number of elements as adaptive 1%

Example 2: what happens when combining basis functions?

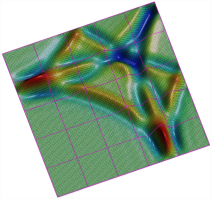
Simulation of the propagation of a source point by a DG method



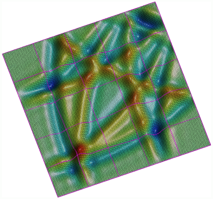
(a) Régulier 35 $\times$ 35: time 1



(b) Régulier 25 $\times$ 25: time 2



(c) Régulier 22 $\times$ 22: time 3



(d) Régulier 24 $\times$ 24: time 4

Figure : Propagation of a source point in a cavity by DG Q^{10} on 5 $\times$ 5 uniform mesh : uniform remeshing with same accuracy as adaptive 1%

Example 2: what happens when combining basis functions?

Simulation of the propagation of a source point by a DG method

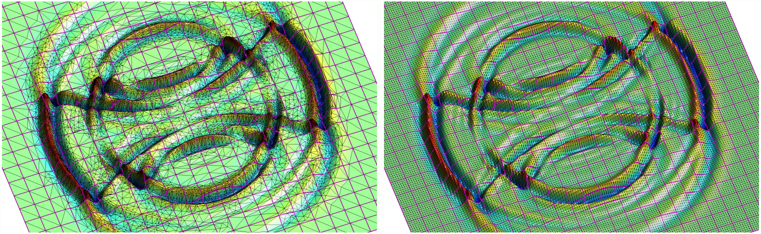
Method	time 1		time 2		time 3		time 4	
	triangles	error	triangles	error	triangles	error	triangles	error
Previsio 1%	3088	1.00%	3728	1.00%	3928	1.00%	5580	1.00%
Uniform 8 × 8	3200	14.30%	-	-	-	-	-	-
Uniform 35 × 35	61250	1.04%	-	-	-	-	-	-
Uniform 9 × 9	-	-	4050	5.72%	-	-	-	-
Uniform 25 × 25	-	-	31250	1.02%	-	-	-	-
Uniform 9 × 9	-	-	-	-	4050	5.82%	-	-
Uniform 22 × 22	-	-	-	-	24200	1.07%	-	-
Uniform 11 × 11	-	-	-	-	-	-	6050	4.43%
Uniform 24 × 24	-	-	-	-	-	-	28800	1.03%

Summary:

- Good stability of the number of triangles created adaptively (not controlled).
- Uniform refinement much less stable → the number of refinements is difficult to *a priori* predict.

Example 3: what happens on more realistic simulations?

Elastic wave in a three layered medium



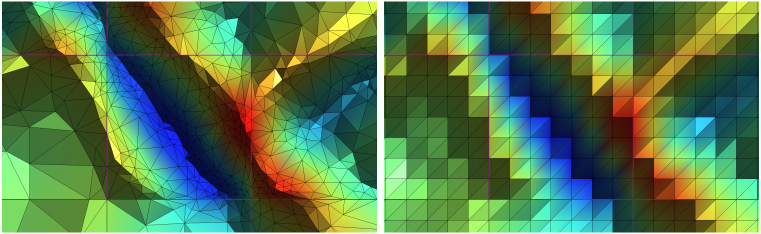
(a) Adaptive 1% (b) Uniform 7×7 (1 point per dof)

Figure : Propagation of an elastic wave in a 3 layered medium by DG Q^7 on 24×24 uniform mesh: global view

$\rightarrow$ No notable differences between those results!?

Example 3: what happens on more realistic simulations?

Elastic wave in a three layered medium



(a) Previsio 1% (b) Uniform 7 $\times$ 7

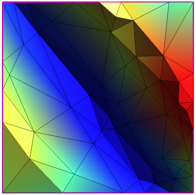
Figure : Propagation of an elastic wave in a 3 layered medium by DG Q^7 on 24 $\times$ 24 uniform mesh: zooming on one cell

$\rightarrow$ Accuracy problems are expected for later extractions!!

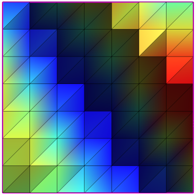
(this cell is meaningful from a physical point of view: interface between two layers with reflection and refraction processes)

Example 3: what happens on more realistic simulations?

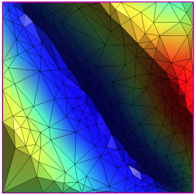
Elastic wave in a three layered medium



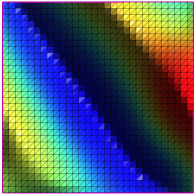
(a) Adaptive, same number of elements than uniform 7 × 7



(b) Uniform 7 × 7



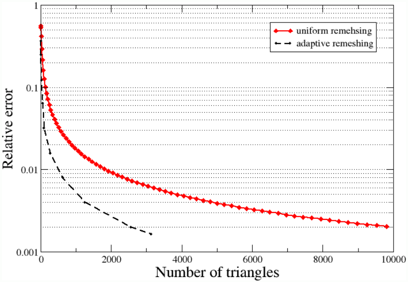
(c) Adaptive 1%



(d) Uniform 1%

Figure : Propagation of an elastic wave in a 3 layered medium: same cell, comparisons

Elastic wave in a three layered medium



	number of triangles	relative error
Uniform 7 $\times$ 7	98	12.65%
Adaptive	97	3.19%
Uniform 14 $\times$ 14	392	4.10%
Adaptive 1%	453	0.99%
Uniform 30 $\times$ 30	1800	1.03%
Uniform 70 $\times$ 70	9800	0.2%
Adaptive 0.2%	2553	0.2%

Example 3: what happens on more realistic simulations?

Elastic wave in a three layered medium

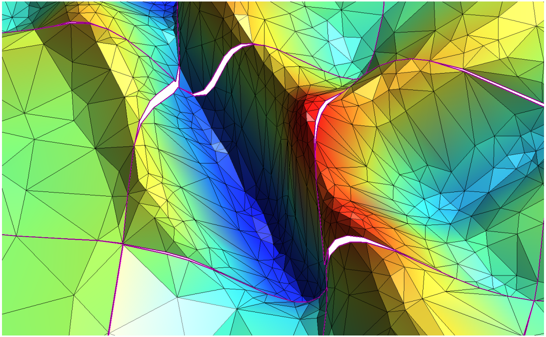
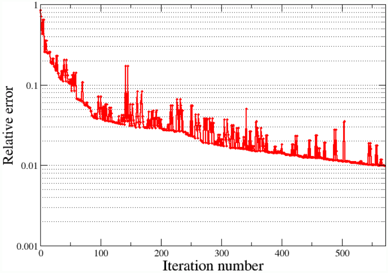
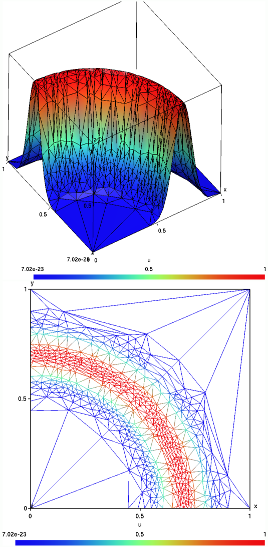


Figure : Adatpive 1%: zoom on solution gaps

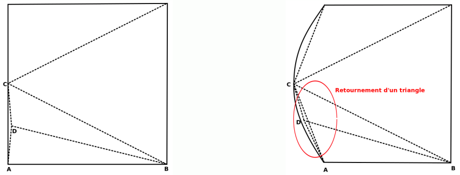
Perspectives and forthcoming works



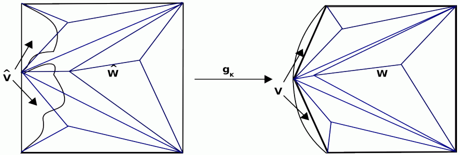


- L^∞ norm $\rightarrow$ no monotonicity.
- Voronoï swaps can yield an increase of the error.
- Idea: swaps driven by a metric associated to g_K and $f_{num} \circ g_K$.

Perspectives and forthcoming works

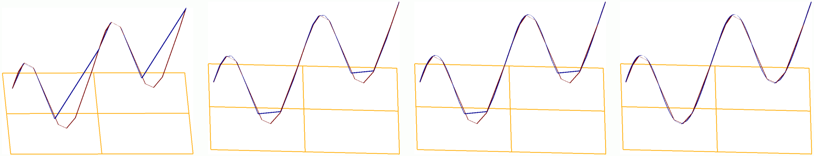


Loss of the injectivity of $P^1 g_K$ when the point D is added $\rightarrow$ loss of the objective O_3.



Solution: add only points in the domain $\widehat{V}$ where $P^1 g_K$ is an injective function.

Remark: domain $\widehat{V}$ defined by geometry but new points driven by both geometry and f_{num}!



(a) L^2 5 vs L^∞ 5%   (b) L^2 1% vs L^∞ 5%   (c) L^2 0.1% vs L^∞ 5%   (d) L^2 0.01% vs L^∞ 5%

Figure : Extraction over a rectilinear curve: comparison between L^∞ and L^2 approaches.

- However, the construction driven by $\delta_{\mathcal{T}(K)}\left(f^K_{num}, f^K_{vis}\right)$ is not sufficient to control the extractions of quantities of interest:

$\leftarrow |f_{vis} \circ P^1 g_K(\hat{x}) - f_{num} \circ g_K(\hat{x})| \leq \varepsilon$ where $P^1 g_K(\hat{x})$ can be different of $g_K(\hat{x})$

$\rightarrow$ For example, $f_{vis}(x)$ does not necessarily "control" the value $f_{num}(x)$.

Fixed by introducing additional constrained (objective O_4) linked to the quantity of interest and the way it is extracted from linear function on simplexes.

Thank you for your attention.