

MESH ADAPTATION: A SHORT INTRODUCTION

PLAN

- What is mesh adaptation?
- A priori error estimator
- How to use it in practice

WHAT IS MESH ADAPTATION

- The numerical models leads to spatial discretization errors
- We want to adapt the mesh size to minimize/control this error
- => Need of a posteriori estimators (that use the approximated solution)
- 2 strategies:
 - uniform refinement of the mesh until the solution convergency (mesh independency): very expensive
 - non-uniform refinement: lot of nodes where the solution has large variations, few where the solution doesn't vary.
- Possibility to take into account the directions of the solution: anisotropic mesh

A PRIORI ERROR ESTIMATOR: UPPER BOUND OF THE ERROR IN THE LINEAR INTERPOLATION

- We suppose that we know the exact solution of our problem, u
- We want to found an upper bound of the distance between u and its linear interpolation over the mesh : $(u - P_h u)$
- The starting point is a Taylor development of $u - P_h u$
- At the end, we can show that :

$$\|u - P_h u\|_{\infty} \leq c_d \max_{x \in T} \max_{\vec{e} \in E'_t} \langle \vec{e}, H_u(x) | \vec{e} \rangle$$

(see: <https://www.ljll.math.upmc.fr/frey/publications/RR-4759.pdf> , p18 -> 23)

- More particularly, in 2D, $C_d = 2/9$

HOW TO USE THIS MAJORATION IN PRACTICE

- If H is linear over K and if we note V_i the vertices of the element K :

$$\text{For } \bar{M} = \max_{v \in V_i} H_u(v),$$
$$\max_{x \in T} \max_{\vec{e} \in E_T} \langle \vec{e}, H_u(x) | \vec{e} \rangle \leq \max_{\vec{e} \in E_T} \langle \vec{e}, \bar{M} \vec{e} \rangle$$

- Thus, the final upper bound of the error of interpolation is:

$$\|u - P_h u\|_{\infty, K} \leq c_d \max_{\vec{e} \in E_T} \langle \vec{e}, \bar{M} \vec{e} \rangle$$

- \Rightarrow The error of interpolation depends on the edge length over the mesh

HOW TO USE THIS MAJORATION IN PRACTICE

- We want a maximal error of ϵ :

$$\epsilon = c_d \langle \vec{e}, \bar{M} \vec{e} \rangle$$

and $1 = \langle \vec{e}, M \vec{e} \rangle$ with $M = \frac{c_d}{\epsilon} M$

- Then, to compute an anisotropic size map : just compute M at each mesh node;
- to compute an isotropic size map, compute the smallest size (s) prescribed by M: if we note L the largest eigenvalue of M, $s^2 = 1/L$.