



**HAL**  
open science

# A unified framework for the computational comparison of adaptive mesh refinement strategies for all-quadrilateral and all-hexahedral meshes: locally adaptive multigrid methods versus h-adaptive methods

Daria Koliesnikova, Isabelle Ramière, Frédéric Lebon

## ► To cite this version:

Daria Koliesnikova, Isabelle Ramière, Frédéric Lebon. A unified framework for the computational comparison of adaptive mesh refinement strategies for all-quadrilateral and all-hexahedral meshes: locally adaptive multigrid methods versus h-adaptive methods. *Journal of Computational Physics*, 2021, 437, pp.110310. 10.1016/j.jcp.2021.110310 . hal-02136085v2

**HAL Id: hal-02136085**

**<https://hal.science/hal-02136085v2>**

Submitted on 11 Jan 2021

**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.

# A unified framework for the computational comparison of adaptive mesh refinement strategies for all-quadrilateral and all-hexahedral meshes: locally adaptive multigrid methods versus h-adaptive methods.

KOLIESNIKOVA Daria<sup>1,2</sup>, RAMIÈRE Isabelle<sup>1</sup>, LEBON Frédéric<sup>2</sup>

<sup>1</sup>CEA, DES, IRESNE, DEC, SESC, LSC, F-13108 Saint-Paul-lez-Durance, France

<sup>2</sup>Aix-Marseille Université, CNRS, Centrale Marseille, LMA, F-13453 Marseille Cedex 13 , France

January 11, 2021

## Abstract

This paper provides a detailed comparison in a solids mechanics context of adaptive mesh refinement methods for all-quadrilateral and all-hexahedral meshes. The adaptive multigrid Local Defect Correction method and the well-known hierarchical  $h$ -adaptive refinement techniques are placed into a generic algorithmic setting for an objective numerical comparison. Such a comparison is of great interest as local multigrid AMR approaches are from now rarely employed to adaptively solve implicit systems in solid mechanics. However they present various interesting features mainly related to their intrinsic idea of partitioning the degrees of freedom on different mesh levels. For this study, we rely on a fully-automatic mesh refinement algorithm providing the desired refined mesh directly from the user-prescribed accuracy. The refinement process is driven by an *a posteriori* error estimator combined to mesh optimality criteria. In this study, the most efficient strategy based on mesh optimality criterion and refinement ratio is identified for all-quadrilateral and all-hexahedral finite elements meshes. The quality of refined meshes is finally appreciated in term of number of nodes but also through the verification of final solution's accuracy. A special attention is devoted to the fulfillment of local precisions which are of great importance from an engineering point of view. Numerical  $2D$  and  $3D$  experiments of different complexities revealing local phenomena enable to highlight the essential features of the considered mesh refinement methods within an elastostatic framework. This study points out the great potentialities of locally adaptive multigrid method, which clearly appears to be the most powerful strategy in terms of standard metrics of efficiency (dimension of systems to be solved, storage requirements, CPU time).

**Keywords:** Adaptive mesh refinement, h-adaptive methods, local multigrid method, mesh optimality criterion, local error, all-quadrilateral and all-hexahedral mesh

# 1 Introduction

Numerical simulations become extensively used for solving challenging problems in physics and engineering. This is particularly the case for non-linear or non-smooth behaviors, complex geometries and bolted or bonded connections. For these last examples, an accurate local solution is often necessary. Problem resolution must combine a discretization technique (finite elements, etc.) and a high-performance solver. There are then questions about the reliability, accuracy, cost and memory space of the resolution methods. For obvious reasons of cost and memory space, it is often impossible to use a uniform very fine discretization. To overcome this issue, one technique consists in looking for an optimal mesh, which is a mesh for which the discretization error is lower than a prescribed tolerance with a minimal number of elements. Obviously, the knowledge of this mesh is *a priori* impossible. A common approach is to get that optimal discretization during computations by relying on one of the adaptive mesh refinement (AMR) methods. These well-known numerical tools enable to automatically adjust the mesh through enriching it in critical regions where the solution is less regular and the error is high.

Among popular state-of-the-art adaptive techniques (e.g. [1, 2, 3, 4, 5]), see Figure 1a, aiming to reduce the discretization error (behaving as  $\mathcal{O}(h^p)$  (with  $h$  being the mesh step and  $p$  being the degree of interpolation function)),  $h$ -adaptive approaches (e.g. [1, 6, 7]) have gained widespread success in engineering thanks to their simplicity and efficiency. These methods enable to achieve optimal convergence rates, in particular when singularities are present [8]. They work on the mesh size  $h$ , thus they are easily implementable over existing industrial solvers. Moreover, these approaches usually involve linear or quadratic elements. Higher-order elements (generally required for  $p$  or  $hp$ -adaptivity) are quite rarely employed in practice within the solid mechanics community, due to lacking smoothness of the underlying solution and potential locking effect [9]. An attractive alternative to widely used  $h$ -adaptive strategies are locally adaptive multigrid (or multilevel) methods (e.g. [10, 11, 12, 13]). They consist in enriching the mesh in regions of interest by adding nested local meshes with finer and finer mesh sizes, see Figure 1b. The problems on the different levels of refinement are solved separately in a sequential manner and linked to each other through an iterative process [14] based on transfer operators. These transfer operators only act as pre- and post-treatments making the implementation of such techniques non-intrusive over existing solvers.

Locally adaptive multigrid methods have been initially introduced and applied for problems in computational fluid dynamics and have only been recently successfully extended to solids mechanics [15, 16, 17]. Among the existing locally adaptive multigrid approaches [13], the Local Defect Correction (LDC) method [18] seems to be the most pertinent choice for solid mechanics problems with localized effects [15].

As  $h$ -adaptive techniques, the LDC method also targets mesh-size  $h$ , but exploits a conceptually different multilevel resolution. From a theoretical point of view, the LDC method can be seen as a partitioned conforming resolution of hierarchical non-conforming  $h$ -adaptive strategies [19]. Both methods have then the same *a priori* error estimator: they converge with the same order in local and global mesh steps. In spite of the theoretical aforementioned common features, to the best of our knowledge, the LDC method has never been numerically compared to  $h$ -adaptive methods.

One of the novelties of our paper is to rigorously compare the performance of locally adaptive multigrid LDC method, and hierarchical  $h$ -adaptive approaches. Applications to 2D and 3D problems with different complexities prove the consistency and importance of the proposed study. It is worth highlighting that in the proposed comparison, the error is measured in energy norm

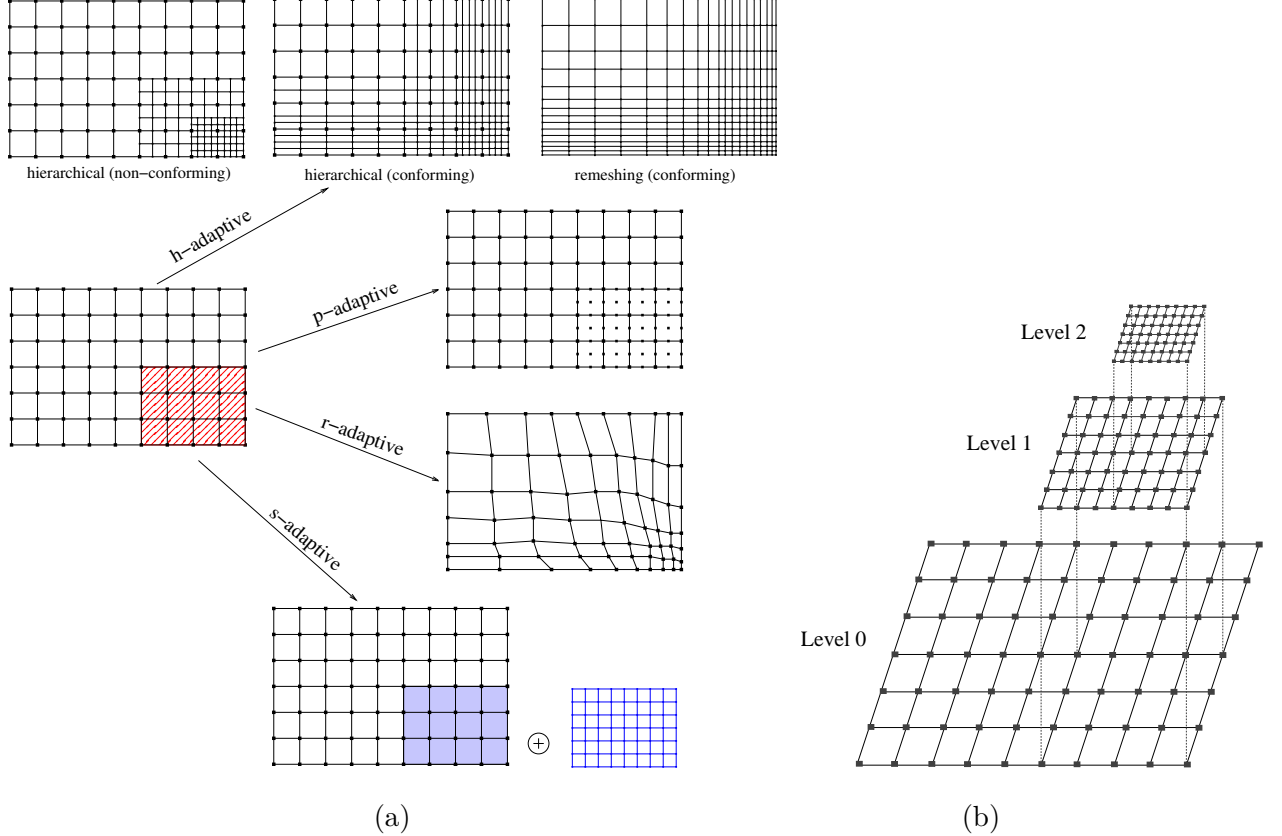


Figure 1: (a) Adaptive methods:  $h$ -,  $p$ -,  $r$ -,  $s$ -refinement approaches; (b) Local multigrid method

(equivalent to the  $H^1$  semi-norm), which is usually the variable of interest in engineering.

Another novelty of our study is to carry out an in-depth study of local multigrid methods and to show the potential of the LDC method as a method for solving complex engineering problems.

Moreover, we pay a special attention to control the local element-wise error and the associated critical region where this error may be high. As mentioned before, sufficiently accurate solution in such regions is often required and has to be controlled. However, it is often never checked in the literature. Generally, only the global accuracy of the solution is looked at on the final refined mesh.

Note that mesh coarsening (also called unrefinement) is not considered here. In addition, mesh adaptation over time is beyond the scope of this paper and will be considered in future studies.

Motivated by a growing interest for quadrilateral/hexahedral meshes, it was decided in this study to deal exclusively with this kind of elements (hybrid meshes, as for example in [20], are excluded). Indeed, such elements are appealing for their tensor-product structure (low bandwidth matrix, efficiency, refinement flexibility [21, 22, 23]) and offer great advantages in industrial applications (remarkable modeling properties [24, 25, 26], no special difficulty to deal with unilateral contact relations [17, 27], plasticity [28], etc). A bilinear (or trilinear) approximation, by so-called  $Q_1$  finite elements, is performed.

In this article we adopt a so-called hierarchical mesh refinement scheme [29, 30] which consists in directly subdividing elements requiring refinement. This strategy generally results in optimal refined meshes and enables to better control the mesh refinement thanks to an explicit hierarchical



data-structure.

Two ways to generate all-quad/hexa  $h$ -adaptive meshes are considered in this study, see Figure 1a. The most straightforward strategy, named here *hierarchical non-conforming h-adaptive technique*, leads to non-conforming meshes which required artificial constraints to maintain the solution continuity. Another approach, named here *hierarchical conforming h-adaptive technique*, aims to restore the mesh conformity, which is generally a challenging task while dealing with all-quad/hexa meshes. In this contribution we decided to avoid the use of refinement template-based schemes, cf. for example the Schneiders’ approach [31] for structured meshes and its extension for non-structured meshes [32], which are nontrivial to implement and require a dedicated mesh generator. We adopt refinement techniques based on layers refinement (see the review paper of [33]), consisting in substituting some layers (so-called sheets) of elements by layers made of more refined elements, see dicing [34], pillowing [35] or matching [36] algorithms.

This paper provides a generic fully-automatic refinement algorithm effective for the three AMR approaches considered here. The mesh adaptation is driven by an *a posteriori* error estimator, which enables to assess the discretization error. As in other types of AMR comparative studies [37, 38, 39], only one estimator is tested since the objective of this paper is the comparison of mesh adaptive methods and not of estimators. Based on the estimated element-wise *a posteriori* error distribution, elements to be refined are marked relying on mesh optimality criteria [40, 41, 42] which are directly linked to the user-prescribed error tolerance. Such marking strategy allows us to precisely reach the desired accuracy and to avoid appearance of the so-called pollution error [43], arising especially while using nested refinement levels as required for the LDC method.

The remaining part of this paper is organized as follows. Section 2 is devoted to the introduction of the generic refinement algorithm. Section 2.1 presents resolution algorithms for each considered AMR method. Sections 2.2 to 2.5.1 define the error estimator, the marking strategy (linked to mesh optimality criterion) and the choice of the refinement ratio, respectively. They are common to the three AMR techniques. Then, sections 2.5.2 to 2.5.5 details the generation of the refined meshes for each refinement method considered here. The in-depth computational comparison based on several ( $2D$  and  $3D$ ) experiments revealing stress concentration or singularities in a linear elastostatic framework is presented and discussed in Section 3. Finally, conclusions are given.

## 2 Adaptive mesh refinement algorithm

An adaptive mesh refinement method generally follows the well-established iterative refinement loop based on following steps, see Figure 2.

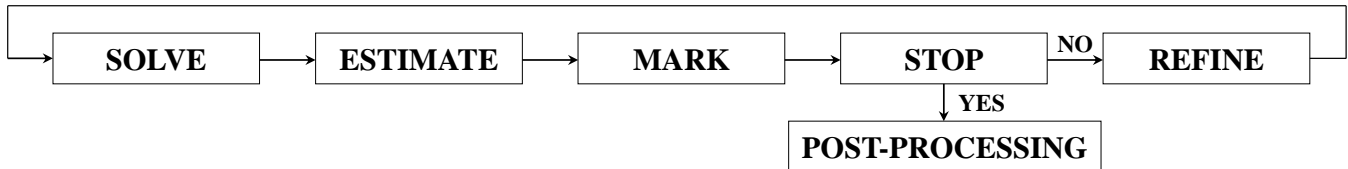


Figure 2: Generic adaptive mesh refinement algorithm

It consists in generating a sequence of approximate solutions  $\{\mathbf{u}_i\}_{i=0}^k$  which aims to converge

toward an acceptable (up to desired accuracy) solution. Thus, starting from an initial discretization  $\mathcal{G}_0$  of a domain  $\Omega \subset \mathbb{R}^D$ , with  $D$  being the dimension of the problem, a series of global meshes with more and more locally refined elements  $\{\mathcal{G}_i\}_{i=1}^k$  is produced, see examples in Figs. 13, 14 and 21. The sub-meshes  $\hat{\mathcal{G}}_i$ ,  $1 \leq i \leq k$ , gather the refined elements of refinement step  $i$  ( $\hat{\mathcal{G}}_i \subset \mathcal{G}_i$ ). The solution  $\mathbf{u}_k$  approximated on the last mesh  $\mathcal{G}_k$  aims to satisfy the prescribed accuracy requirements. Let  $T_i$  be an element of the mesh  $\mathcal{G}_i$ ,  $0 \leq i \leq k$ .

Relying on common steps included into the refinement procedure depicted in Figure 2, we propose here a generic fully-automatic refinement algorithm (Algorithm 1) applicable for the AMR methods considered here ( $h$ -adaptive approaches and local multigrid techniques). The refinement process aims to satisfy a user-defined accuracy through verifying (at each refinement iteration) the achievement of global and local error tolerances.

---

**Algorithm 1** Adaptive mesh refinement algorithm

---

**Input:** user-defined global error tolerance  $\epsilon_\Omega$ , user-defined local volume control parameter  $\delta$ , initial mesh  $\mathcal{G}_0$

**Output:** Converged numerical solution

Set  $i := 0$

$\hat{\mathcal{G}}_0 \leftarrow \mathcal{G}_0$

boolean **stop** := false

**while not stop do**

- $\mathbf{u}_i := \text{SOLVE}_{\text{METHOD}}(\mathcal{G}_i \text{ or } \{\hat{\mathcal{G}}_g\}_{g=0}^i)$
- $\{\xi_{T^i}\}_{T^i \in \mathcal{G}_i} := \text{ESTIMATE}(\mathbf{u}_i, \mathcal{G}_i)$
- $\mathcal{M}_i^0 := \text{MARK}(\mathbf{u}_i, \{\xi_{T^i}\}_{T^i \in \mathcal{G}_i}, \mathcal{G}_i)$
- **stop** := **STOP**( $\{\xi_{T^i}\}_{T^i \in \mathcal{G}_i}, \mathcal{M}_i^0, \epsilon_\Omega, \delta$ )  
     **if stop** = true **then** exit **end if**
- $\mathcal{G}_{i+1}, \hat{\mathcal{G}}_{i+1} := \text{REFINE}_{\text{METHOD}}(\mathcal{M}_i, \{\beta_{T^i}\}_{T^i \in \mathcal{M}_i}, \mathcal{G}_i, \hat{\mathcal{G}}_i)$
- $i \leftarrow i + 1$

**end while**

---

To make this paper self-contained, each of the used modules is presented in a detailed manner in the following sections.

## 2.1 Discrete solution : module SOLVE

A generic boundary-value problem of the following form is considered:

$$(\mathcal{P}) : \begin{cases} \mathcal{L}(\mathbf{u}) = \mathbf{f} & \text{in } \Omega \\ \text{Boundary conditions} & \text{on } \partial\Omega \end{cases} \quad (1)$$

with  $\Omega$  being a bounded domain in  $\mathbb{R}^D$  (with  $D$  – the dimension) and boundary  $\partial\Omega$ . We denote by  $\mathbf{u}$  the problem's solution, by  $\mathcal{L}$  a (non-linear) differential operator and  $\mathbf{f}$  a right-hand side. Any type of boundary conditions are considered on  $\partial\Omega$  while problem  $(\mathcal{P})$  is well-posed.

After discretization, the following discrete problem has to be solved on a triangulation (or quadrangulation)  $\mathcal{G}_i$  of the domain  $\Omega$ :

$$(\mathcal{P}_i) : \begin{cases} \mathcal{L}_i(\mathbf{u}_i) = \mathbf{f}_i & \text{in } \mathcal{G}_i \\ \text{Boundary conditions} & \text{on } \partial\mathcal{G}_i \end{cases} \quad (2)$$

with  $\mathcal{L}_i$  and  $\mathbf{f}_i$  – the discretized operator and right-hand side, respectively. Then,  $\mathbf{u}_i$  is called approximation of  $\mathbf{u}$ .

The module **SOLVE** stands for application of a numerical method to solve the discrete problem (2):

$$\mathbf{u}_i := \text{SOLVE}_{\text{METHOD}}(\mathcal{G}_i \text{ or } \{\hat{\mathcal{G}}_g\}_{g=0}^i) \quad (3)$$

### 2.1.1 $h$ -adaptive methods

#### **Hierarchical non-conforming $h$ -adaptive method: module $\text{SOLVE}_{\text{NCHAM}}$**

Generally, the hierarchical non-conforming  $h$ -adaptive method (NCHAM) leads to optimal meshes in terms of number of degrees of freedom. However, since the appearance of hanging nodes is unavoidable, it requires the use of solver dedicated to non-conforming meshes. Several approaches to manage the non-conformity relations exist (e.g. [44, 45, 46, 47, 48] and the references therein). In this study, Lagrange multipliers [44] have been chosen to impose interelements continuity. The module  $\text{SOLVE}_{\text{NCHAM}}$  is given by Algorithm 2.

---

#### **Algorithm 2 $\text{SOLVE}_{\text{NCHAM}}$**

---

**Input:** Mesh  $\mathcal{G}_i$ , data of the problem

**Output:** Solution  $\mathbf{u}_i$  in  $\mathcal{G}_i$

- Introduce in  $(\mathcal{P}_i)$  non-conformity relations with a chosen approach
  - Solve the modified problem  $(\mathcal{P}_i)$  in  $\mathcal{G}_i$  to obtain  $\mathbf{u}_i$
- 

#### **Hierarchical conforming $h$ -adaptive method: module $\text{SOLVE}_{\text{CHAM}}$**

The hierarchical conforming  $h$ -adaptive method (CHAM) aims to generate a conforming locally refined mesh at each refinement iteration. The module  $\text{SOLVE}_{\text{CHAM}}$  (cf. Algorithm 3) lies on the standard solution of a discrete problem on the refined conforming mesh  $\mathcal{G}_i$ .

---

#### **Algorithm 3 $\text{SOLVE}_{\text{CHAM}}$**

---

**Input:** Mesh  $\mathcal{G}_i$ , data of the problem

**Output:** Solution  $\mathbf{u}_i$  in  $\mathcal{G}_i$

- Solve the discrete problem  $(\mathcal{P}_i)$  in  $\mathcal{G}_i$  to obtain  $\mathbf{u}_i$
- 

### 2.1.2 Locally adaptive multigrid methods

Intruduced in a multilevel mesh refinement context, the idea behind locally adaptive multigrid methods [11, 13] is to provide a possibility to “zoom“ the computational domain in the regions of interest and thus to locally improve the accuracy of the solution. Starting from an initial coarse mesh covering the whole computational domain  $\Omega$ , these techniques consist in adding levels of meshes with finer and finer discretization steps only in regions where higher precision is required, see Figure 1b. The resolution consists in solving in a sequential manner the problems associated to the different levels of (conforming) meshes and coupling them through a multigrid-like iterative

process, see Figure 3.

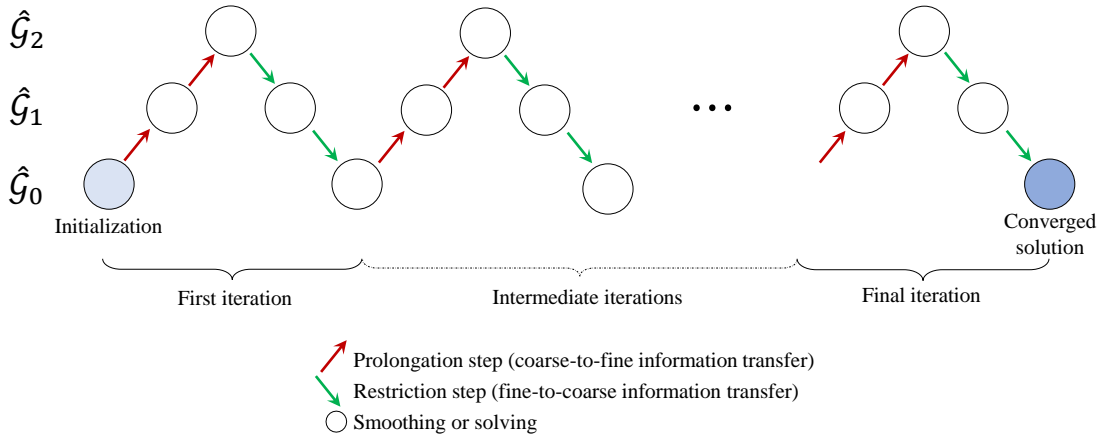


Figure 3: Locally adaptive multigrid  $\Lambda$ -cycle scheme with a sequence of three levels of meshes  $\{\hat{\mathcal{G}}_i\}_{i=0}^2$ .

### Local Defect Correction method: module $\text{SOLVE}_{\text{LDC}}$

As mentioned in Introduction, in this study we have chosen to focus on the adaptive multigrid Local Defect Correction approach, initially proposed by Hackbusch in [18], which seems to be the most suitable choice for structural mechanics problems involving localized effects. The reader is referred to [18] for a more detailed presentation of the LDC algorithm and to [15, 42] for its adaptation to linear elasticity.

Integrated in the generic AMR process (Algorithm 1), the LDC multilevel iterative resolution is carried out on a sequence of local sub-meshes  $\{\hat{\mathcal{G}}_g\}_{g=0}^i$  at each refinement step  $i$ . As any locally adaptive multigrid methods, the LDC prolongation operator consists in prescribing Dirichlet boundary conditions on the so-called internal boundary of  $\hat{\mathcal{G}}_g$  defined as  $\partial\hat{\mathcal{G}}_g \setminus \partial\Omega$  (see Figure 4a). Dirichlet values are obtained from the next coarser solution lying on  $\hat{\mathcal{G}}_{g-1}$ . The LDC restriction operator enables us to obtain a restriction of the next finer solution on the current mesh, which serves to derive a residual (also called defect) acting as an additional source term in the current level problem. The module  $\text{SOLVE}_{\text{LDC}}$  is detailed by Algorithm 4. Note, that performing the  $\Lambda$ -cycle LDC Algorithm (see Figure 3) at each iteration of the adaptive refinement Algorithm 1 returns to a progressive  $M$ -cycles version [49] of the LDC method.

In Algorithm 4,  $\mathcal{L}_g$  denotes the discrete operator associated to mesh  $\hat{\mathcal{G}}_g$ , and  $\mathbf{f}_g$  the right-hand side of the problem. The notation  $\mathbf{f}_g^0$  corresponds to the discretization of problem's right-hand side  $\mathbf{f}$  (see Eq.(1)) on sub-mesh  $\hat{\mathcal{G}}_g$ . The iterative process stops once the solution on the coarsest level has converged.

The restriction operator requires to define a set of restriction nodes  $A_g$  on which the next finer solution is restricted. This set  $A_g$  is made of the nodes of  $\hat{\mathcal{G}}_g$  strictly included in the domain discretized by the fine mesh  $\hat{\mathcal{G}}_{g+1}$  (cf. Figure 4b). A set of correction nodes, referred to as  $\hat{A}_g$ , is a region where the coarse problem's right-hand side is updated (cf. Figure 4c). It is defined as the set of interior nodes of  $A_g \cup (\partial\hat{\mathcal{G}}_g \cap \partial\Omega)$  (i.e.  $\hat{A}_g \in A_g \cup (\partial\hat{\mathcal{G}}_g \cap \partial\Omega)$ ) in the sense of the discretization

---

**Algorithm 4 SOLVE<sub>LDC</sub>**


---

**Input:** Sub-meshes  $\{\hat{\mathcal{G}}_g\}_{g=0}^i$ , data of the problem,  $\{\mathbf{f}_g^0\}_{g=0}^i$  RHS discretized on  $\{\hat{\mathcal{G}}_g\}_{g=0}^i$ , convergence parameter  $\tau$

**Output:** Solutions  $\{\mathbf{u}_g\}_{g=0}^i$  in  $\{\hat{\mathcal{G}}_g\}_{g=0}^i$

if  $i = 0$ ,  $\hat{\mathcal{G}}_0 \leftarrow \mathcal{G}_0$

Solve the discrete problem  $(\mathcal{P}_0)$  in  $\hat{\mathcal{G}}_0$  with right-hand side  $\mathbf{f}_0^0$  to obtain  $\mathbf{u}_0$

else

Set  $j := 1$ , boolean *convergence* := false

**while not convergence do**

Set  $\mathbf{u}_0^j := \mathbf{u}_0^{j-1}$  with  $\mathbf{u}_0^0 := \mathbf{u}_0$

**for**  $g = 1, \dots, i$

**if**  $(j = 1 \text{ and } g < i)$  **then**  $\mathbf{f}_g^1 := \mathbf{f}_g$  **else**  $\mathbf{f}_g^j := \mathbf{f}_g^{j-1}$  **end if**

**end for**

- **Prolongation step.** Resolution on fine meshes  $\hat{\mathcal{G}}_g$

**for**  $g = 1, \dots, i$

◦ Impose boundary conditions on the mesh boundaries  $\partial\hat{\mathcal{G}}_g$ :

– boundary conditions of the continuous problem on  $\partial\hat{\mathcal{G}}_g \cap \partial\Omega$ ;

– and Dirichlet boundary condition imposed on the internal boundary

$\Gamma_g^f = \partial\hat{\mathcal{G}}_g \setminus (\partial\hat{\mathcal{G}}_g \cap \partial\Omega)$ . Dirichlet values are derived from the next coarser solution:

$$\mathbf{u}_{g,\Gamma_g^f}^j = \mathcal{P}_{g-1}^g(\mathbf{u}_{g-1}^j)|_{\Gamma_g^f} \quad (4)$$

with  $\mathcal{P}_{g-1}^g$  a prolongation operator (typically interpolation operator)

◦ Solve the problem  $(\mathcal{P}_g)$  in  $\hat{\mathcal{G}}_g$  with the right-hand side  $\mathbf{f}_g^j$  to obtain  $\mathbf{u}_g^j$

- **Restriction step.** Correction on coarse meshes  $\hat{\mathcal{G}}_g$

**for**  $g = i - 1, \dots, 0$

◦ Compute the restricted fine displacement  $\mathbf{u}_{g+1}^j$  in  $A_g$ :

$$\tilde{\mathbf{u}}_g^j(x) = \left( \mathcal{R}_{g+1}^g(\mathbf{u}_{g+1}^j) \right)(x) \quad \forall x \in A_g \quad (5)$$

with  $\mathcal{R}_{g+1}^g$  ( $\neq (\mathcal{P}_g^{g+1})^\top$ ) a restriction operator (interpolation or injection operator)

◦ Compute the residual in  $\hat{A}_g$ :

$$\mathbf{r}_g^j(\mathbf{u})(x) = \left( \mathcal{L}_g \tilde{\mathbf{u}}_g^j - \mathbf{f}_g^0 \right)(x) \quad \forall x \in \hat{A}_g \quad (6)$$

◦ Correct the right-hand side  $\mathbf{f}_g^j$  for the problem in  $\hat{\mathcal{G}}_g$ :

$$\mathbf{f}_g^j = \mathbf{f}_g^0 + \chi_{\hat{A}_g} \mathbf{r}_g^j(\mathbf{u}) \quad (7)$$

where  $\chi_{\hat{A}_g}$  is the characteristic function of  $\hat{A}_g$ :  $\chi_{\hat{A}_g}(x) = \begin{cases} 1 & \text{if } x \in \hat{A}_g \\ 0 & \text{elsewhere} \end{cases}$

◦ Solve the discrete problem  $(\mathcal{P}_g)$  in  $\hat{\mathcal{G}}_g$  with corrected right-hand side  $\mathbf{f}_g^j$  to update  $\mathbf{u}_g^j$

- **Check the convergence:** if  $\frac{\|\mathbf{u}_0^j - \mathbf{u}_0^{j-1}\|}{\|\mathbf{u}_0^j\|} \leq \tau$  **then** *convergence* := true

$j \leftarrow j + 1$

**end while**

**end if**

$\mathbf{u}_g \leftarrow \mathbf{u}_g^j$  and  $\mathbf{f}_g \leftarrow \mathbf{f}_g^j \quad \forall g = 0, \dots, i$

---

scheme:  $x \in A_g \cup (\partial\hat{\mathcal{G}}_g \cap \partial\Omega)$  is called interior node (i.e.  $x \in \mathring{A}_g$ ) if  $\mathcal{L}_g \mathbf{u}_g(x)$  involves only  $\mathbf{u}_g(y)$  with  $y \in A_g \cup (\partial\hat{\mathcal{G}}_g \cap \partial\Omega)$ .

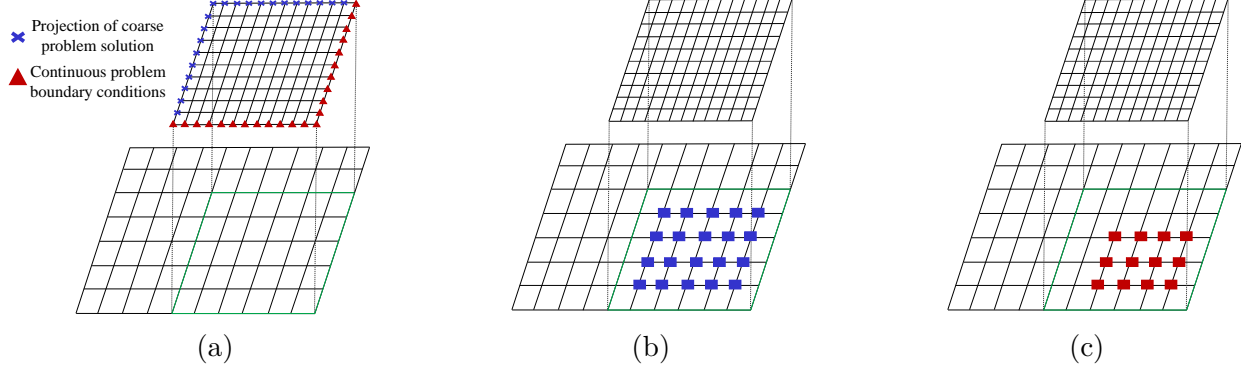


Figure 4: Illustration of LDC prolongation and restriction steps between two consecutive levels : (a) Boundary conditions; (b) Set of restriction nodes  $A_g$  ; (c) Set of correction nodes  $\mathring{A}_g$

Note that since the transfer operators of the LDC method affect the problem's right-hand side only, the matrix factorization can be kept. Thus, the locally adaptive multigrid process is computationally cheap and can be implemented in any existing solver in a non-intrusive way. In addition, regular and (quasi-)uniform meshes can be used on each level leading to the generally well-conditioned problems. Moreover, local multigrid methods are really generic and flexible as they enable us to change the refinement ratio, the model, the solver, etc. between each level (see for example [18, 50, 51]).

In the numerical part of this manuscript, the prolongation operator  $\mathcal{P}_{g-1}^g$  is a basis function-based (bi or trilinear) interpolation while a canonical restriction  $\mathcal{R}_g^{g-1}$  is employed since hierarchical nested meshes are generated. The chosen operators orders are in agreement with the expected first-order accuracy of the approach, as already discussed in [15]. The convergence parameter in Algorithm 4 is set  $\tau = 10^{-7}$  and the Euclidean norm is used.

## 2.2 Error estimation: module ESTIMATE

In Algorithm 1, the key ingredient enabling to automatically drive the mesh adaptation is the error estimator. The module **ESTIMATE** is dedicated to the *a posteriori* error estimation.

Since the scope of this study *is not to compare the performance of error estimators* (see for example [52, 53, 54] for this purpose), an error estimator is chosen as a tool to lead the mesh refinement, as in other studies focusing of AMR methods comparison (e.g. [39, 38, 37]). Hence, in this paper, the widely used recovery-based *a posteriori* error estimator introduced by Zienkiewicz and Zhu in [40, 55], called ZZ error estimator, has been chosen for its robustness, simplicity of implementation and performance (limited computational cost).

The element-wise error distribution  $\{\xi_{T^i}\}_{T^i \in \mathcal{G}_i}$  is thus assessed based on the difference between a reconstructed smoothed gradient (here, the stress field)  $\sigma_i^s$  and a discrete one  $\sigma_i$  (obtained from the computation). In a linear elasticity framework, the stress field is defined as  $\sigma_i = \mathbf{C} \varepsilon_i$  (with  $\mathbf{C}$  being the fourth order elasticity tensor), and the strain field as  $\varepsilon_i = \frac{1}{2}(\mathbf{grad} \mathbf{u}_i + \mathbf{grad}^T \mathbf{u}_i)$ . For the reconstruction of the smoothed gradient we consider the recovery procedure introduced in

[40]. The smoothed stress field  $\boldsymbol{\sigma}_i^s$  is obtained by projection on the primal (here, the displacement) basis functions. The element-wise error distribution in the energy norm is estimated as:

$$\xi_{T^i} := \|\mathbf{e}\|_{T^i} \simeq \left( \int_{T^i} (\boldsymbol{\sigma}_i^s - \boldsymbol{\sigma}_i) : (\boldsymbol{\varepsilon}_i^s - \boldsymbol{\varepsilon}_i) dT^i \right)^{\frac{1}{2}} \quad \text{for } T^i \in \mathcal{G}_i \quad (8)$$

with  $\boldsymbol{\varepsilon}_i^s$  the strain field associated to the smoothed stress field  $\boldsymbol{\varepsilon}_i^s = \mathbf{C}^{-1} \boldsymbol{\sigma}_i^s$ .

The global absolute estimated error is then obtained by summing the elementary contributions:

$$\xi_{\mathcal{G}_i} := \left( \sum_{T^i \in \mathcal{G}_i} \xi_{T^i}^2 \right)^{\frac{1}{2}} \quad (9)$$

## 2.3 Marking stage: module MARK

### 2.3.1 Mesh optimality criteria: sub-module OPTIMALITY

The marking strategy lies either on a qualitative or quantitative use of the estimated element-wise error field. The qualitative approach is based on a parametric detection of the elements to be refined, for example, quantile marking, Dörfler marking or maximum marking (see [56, 57, 58, 59] for more details). The optimal choice of the required parameter is not obvious as it is not directly related to the desired accuracy and may be problem dependent [16]. Empiric values are often used, which require a non-nested refinement process in order to reach the user-prescribed accuracy. In this case, the final refined mesh is generally not optimal in terms of degrees of freedom.

In this study, we focus on a quantitative-based marking procedure relying on the use of the mesh optimality criteria. This procedure consists in defining a maximal permissible elementary error based on different considerations about the distribution of error over elements in an optimal mesh (e.g. [60, 40, 41, 61, 42]). The use of mesh optimality criteria most often naturally leads to nested refined regions, constructed for a given error tolerance  $\epsilon_\Omega$ , without penalizing performance of AMR methods.

Generally in the literature (see for example [39, 61, 60]), the mesh is assumed acceptable when the following condition holds:

$$\frac{\xi_{\mathcal{G}_i}}{\epsilon_\Omega \omega_{\mathcal{G}_i}} = 1 \quad (10)$$

with  $\omega_{\mathcal{G}_i}$  the global strain energy, used to obtain the relative global error and  $\epsilon_\Omega$  the user-defined tolerance on the global error. For the ZZ error estimator [40], the estimated error is weighted with respect to the global strain energy  $\omega_{\mathcal{G}_i}$  obtained by summing the elementary contributions  $\omega_{T^i}$ :

$$\omega_{\mathcal{G}_i} := \left( \sum_{T^i \in \mathcal{G}_i} \omega_{T^i}^2 \right)^{\frac{1}{2}} \quad (11)$$

with

$$\omega_{T^i} := \|\mathbf{w}\|_{T^i} \simeq \left( \int_{T^i} \boldsymbol{\sigma}_i : \boldsymbol{\varepsilon}_i dT^i + \int_{T^i} (\boldsymbol{\sigma}_i^s - \boldsymbol{\sigma}_i) : (\boldsymbol{\varepsilon}_i^s - \boldsymbol{\varepsilon}_i) dT^i \right)^{\frac{1}{2}} \quad \text{for } T^i \in \mathcal{G}_i \quad (12)$$

which is close to the energy norm of computed stress field  $\boldsymbol{\sigma}_i$  but the additional term enables to set to one the upper bound of the ratio  $\frac{\xi_{\mathcal{G}_i}}{\omega_{\mathcal{G}_i}}$  (Eq.(10)) even in case of low stress. As mentioned in Introduction, in this work, a special interest is devoted to the verification of the imposed accuracy

locally (per element). Thus we are also interested in building a mesh, aiming to fulfill the user-prescribed accuracy  $\epsilon_\Omega$  on each element  $T^i \in \mathcal{G}_i$ :

$$\frac{\xi_{T^i}}{\epsilon_\Omega \omega_{T^i}} = 1 \quad (13)$$

The sub-module **OPTIMALITY** consists in defining the maximal permissible element-wise error  $\{\xi_{T^i}^{max}\}_{T^i \in \mathcal{G}_i}$  for a given mesh optimality criterion. This maximal permissible error is, moreover, a function of the user-prescribed tolerance  $\epsilon_\Omega$ .

In this study we consider mesh optimality criteria introduced by Oñate and Bugeda in [41], referred here as *OB criterion*, and the one proposed by Ramière et al. in [42], referred here as *LOC criterion*. These both criteria target to satisfy global (10) and local (13) accuracy requirements.

- **OB criterion: Equal distribution of the specific error.**

In [41, 62], Oñate and Bugeda postulate that the maximal permissible element-wise error distribution should be governed by the elements measure. Following this, the ratio between the elementary error's square and the elements measure must be constant over the whole mesh and equal to the ratio of the global error's square over the domain's measure:

$$\frac{(\xi_{T^i}^{max})^2}{\mu(T^i)} = \frac{(\xi_{\mathcal{G}_i}^{max})^2}{\mu(\Omega)} \quad \forall T^i \in \mathcal{G}_i \quad (14)$$

with  $\mu(T^i)$  being the measure of element  $T^i$  and  $\mu(\Omega)$  – the computational domain's total measure. Setting for the global optimal mesh condition  $\xi_{\mathcal{G}_i}^{max} = \epsilon_\Omega \omega_{\mathcal{G}_i}$  (Eq.(10)), the maximal permissible error  $\{\xi_{T^i}^{max,OB}\}_{T^i \in \mathcal{G}_i}$  delivered by the sub-module **OPTIMALITY<sub>OB</sub>** reads:

$$\xi_{T^i}^{max,OB} := \epsilon_\Omega \omega_{\mathcal{G}_i} \left( \frac{\mu(T^i)}{\mu(\Omega)} \right)^{\frac{1}{2}} \quad \text{for } T^i \in \mathcal{G}_i \quad (15)$$

- **LOC criterion: Local element-wise error.**

The alternative criterion was briefly mentioned in [41] but effectively introduced and exploited in [42, 17] in the context of structural mechanics problems. It consists in prescribing the desired accuracy directly on the local element-wise error (Eq.(13)), thus the maximal permissible error  $\{\xi_{T^i}^{max,LOC}\}_{T^i \in \mathcal{G}_i}$  delivered by the module **OPTIMALITY<sub>LOC</sub>** reads:

$$\xi_{T^i}^{max,LOC} := \epsilon_\Omega \omega_{T^i} \quad \text{for } T^i \in \mathcal{G}_i \quad (16)$$

As proved in [42], it automatically implies the fulfillment of the global prescribed accuracy ( $\frac{\xi_{\mathcal{G}_i}}{\epsilon_\Omega \omega_{\mathcal{G}_i}} \leq 1$ ).

**Remark.** The most popular mesh optimality criterion relying on the error equidistribution principle proposed by Zienkiewicz and Zhu [40] targets to satisfy only the global accuracy (10). Thus, this mesh optimality criterion is not of interest for our purposes. Moreover, as already discussed in [41], this criterion may lead to an inconsistent oscillatory refinement process of some mesh regions (for *h*-adaptive methods). This criterion has been tested on our numerical examples and such behavior has been also observed.



### 2.3.2 Detect elements: sub-module DETECT

The sub-module **DETECT** takes as input the element-wise estimated error distribution  $\{\xi_{T^i}\}_{T^i \in \mathcal{G}_i}$  as well as the maximal permissible error  $\{\xi_{T^i}^{max}\}_{T^i \in \mathcal{G}_i}$  defined by the sub-module **OPTIMALITY**. It delivers the set of marked elements  $\mathcal{M}_i^0$ :

$$\mathcal{M}_i^0 := \{T^i \in \mathcal{G}_i; \quad \xi_{T^i} > \xi_{T^i}^{max}\} \quad (17)$$

It has to be specified, that for the LDC approach or any refinement strategy implying nested refinement zones, the elements for refinement are selected in the sub-mesh  $\hat{\mathcal{G}}_i$ , as follows:

$$\mathcal{M}_i^0 \subset \hat{\mathcal{G}}_i \quad \forall i \quad (18)$$

Note, that terms involving  $\mathcal{G}_i$  for the definition of  $\xi_{T^i}^{max}$  ( $\omega_{\mathcal{G}_i}$ ,  $\#\mathcal{G}_i$ , etc.) remain evaluated on the (eventually composite) mesh  $\mathcal{G}_i$ .

## 2.4 Stopping criteria: module STOP

The module **STOP** outputs a boolean (**stop**) which indicates if the chosen stopping criteria have been satisfied or not. If both stopping criteria defined below are satisfied, we assign to the boolean **stop** := true, which allows us to turn off the refinement process, cf. Algorithm 1.

In the ideal scenario, a refinement process has to achieve:

$$\mathcal{M}_i^0 = \emptyset \quad (19)$$

However, as the solution may converge very slowly near a singularity, for mesh optimality criteria based on the local relative error (especially LOC criterion, see Eq.(16)), the respect of condition (19) can lead to a costly refinement. Thus, a common practice is to impose other stopping criteria, which permit to turn off the refinement process once the mesh is supposed to be acceptable.

In the present study we are concerned with the verification of both global and local accuracies. Two stopping criteria have thus to be satisfied in order to stop the refinement process:

- **Global criterion.**

The fulfillment of the global accuracy relies on the verification of the following condition:

$$\frac{\xi_{\mathcal{G}_i}}{\epsilon_{\Omega} \omega_{\mathcal{G}_i}} \leq 1 \quad (20)$$

- **Local criterion.**

The fulfillment of the local accuracy is verified based on local stopping criterion, initially proposed in [42]. It consists in user-controlling the zone where the local accuracy is not respected:

$$\frac{\mu(\Omega_{\mathcal{M}_i^0})}{\mu(\Omega)} \leq \delta \quad (21)$$

with  $\mu(\Omega_{\mathcal{M}_i^0})$  the measure of zone  $\Omega_{\mathcal{M}_i^0} = \{\cup \bar{T}^i; T^i \in \mathcal{M}_i^0\}$  and  $\delta$  the user-specified volume control parameter ( $\delta \leq 1$ ).

Using criterion (21), the prescribed mesh optimality criterion aims to be fulfilled on  $(1 - \delta) \times 100$  percent of the domain. This stopping criterion is interesting as it enables to automatically determine a discrete approximation of the critical region (singularity zone), see [42] for more details.

**Remark.** Setting  $\delta = 1$  returns to setting only the standard global stopping criterion (10), while the choice  $\delta = 0$  allows us to reach (19).

## 2.5 Mesh refinement: module REFINE

### 2.5.1 Refinement ratio: sub-module RATIO

The sub-module **RATIO** aims to output the refinement ratio  $\beta_{T^i}^0 \in \mathbb{N}^+$  for  $T^i \in \mathcal{M}_i^0$ . In this work, influence of different choices of refinement ratios is studied. We consider here two possible ways to obtain the refinement ratio  $\beta_{T^i}^0$ :

- **Fixed refinement ratio.**

The easiest strategy, permitting to progressively refine the mesh, consists in dividing the marked elements with an *a priori* fixed uniform refinement ratio:

$$\beta_{T^i}^{0,fix} := const \quad \forall T^i \in \mathcal{M}_i^0, \forall i \quad (22)$$

In this study, the following value is chosen:

$$\beta_{T^i}^{0,fix} := 2 \quad (23)$$

as usually in practical industrial simulations (see for example [15, 17, 20, 63, 64]).

- **Adjusted refinement ratio.**

The second approach aims to minimize the number of refinement steps (maximal number of  $i$  in Algorithm 1) by using an adjusted refinement ratio.

To compute the adjusted refinement ratio  $\beta_{T^i}^{0,adj}$ , an ansatz of the *a priori* local estimator [40], derived from the classical *a priori* global estimator [65], is usually used:

$$\xi_{T^i} = \mathcal{O}(C_{T^i} h_{T^i}^p) \quad (24)$$

with  $C_{T^i}$  being a constant depending on the element size and  $p$  representing the polynomial order of the discrete basis functions. In the literature, in order to easily determine  $\beta_{T^i}^{0,adj}$ , the constant  $C_{T^i}$  is supposed to be independent of the element  $T^i$ :  $C_{T^i} \simeq C$ .

The adjusted refinement ratio is defined as:

$$\beta_{T^i}^{0,adj} = \frac{h_{T^{i+1}}}{h_{T^i}} \quad \text{for } T^i \in \mathcal{M}_i^0 \quad (25)$$

with  $h_{T^{i+1}}$  and  $h_{T^i}$  being the mesh size of the refined mesh and of the current mesh, respectively.

We consider  $\xi_{T^{i+1}}$  to be the element-wise error required for the mesh  $\mathcal{G}_{i+1}$ . By definition, it is equivalent to the maximal permissible error  $\xi_{T^i}^{max}$ , see Section 2.3.1. Considering expressions (24) and (25), we derive the following formulation:

$$\beta_{T^i}^{0,adj} := \left\lceil \left( \frac{\xi_{T^i}}{\xi_{T^i}^{max}} \right)^{\frac{1}{p}} \right\rceil \quad \text{for } T^i \in \mathcal{M}_i^0 \quad (26)$$

where  $\lceil \cdot \rceil$  states for rounding to the next higher integer.

Using the expression of  $\xi_{T^i}^{max}$  in Eq.(15) for OB mesh optimality criterion and Eq.(16) for LOC mesh optimality the adjusted refinement ratios are derived:

$$\beta_{T^i}^{0,OB} := \left\lceil \left( \frac{\xi_{T^i}}{\epsilon_\Omega \omega_{\mathcal{G}_i}} \left( \frac{\mu(\Omega)}{\mu(T^i)} \right)^{\frac{1}{2}} \right)^{\frac{1}{p}} \right\rceil \quad \text{for } T^i \in \mathcal{M}_i^{0,OB} \quad (27)$$

$$\beta_{T^i}^{0,LOC} := \left\lceil \left( \frac{\xi_{T^i}}{\epsilon_\Omega \omega_{T^i}} \right)^{\frac{1}{p}} \right\rceil \quad \text{for } T^i \in \mathcal{M}_i^{0,LOC} \quad (28)$$

**Remark.** As already mentioned in Introduction,  $Q_1$  finite elements are used to discretize the problems. The polynomial order of interpolation functions is then  $p = 1$ .

**Remark.** The adjusted refinement ratio strategy does not fit the philosophy of the LDC method (generation of (quasi-)uniform sub-meshes) and would return to perform an  $h$ -adaptive approach on each LDC sub-mesh, which is not the aim of this study. In order to keep the LDC strategy unchanged, the adjusted ratio won't be coupled to the LDC method. Thus, the so-called mean adjusted refinement ratio is applied to the LDC method. Its values are obtained by weighting the adjusted ratio values with respect to the element measures:

$$\bar{\beta}_{T^i}^0 := \left\lceil \frac{1}{\mu(\Omega_{\mathcal{M}_i^0})} \sum_{T^i \in \mathcal{M}_i^0} \beta_{T^i}^{0,adj} \cdot \mu(T^i) \right\rceil \quad \text{for } T^i \in \mathcal{M}_i^0 \quad (29)$$

where  $\beta_{T^i}^{0,adj}$  represents an adjusted refinement ratio (cf. Eqs.(27) and (28)),  $\mu(T^i)$  – the measure of element  $T^i$  and  $\mu(\Omega_{\mathcal{M}_i^0})$  – the measure of the marked zone.

### 2.5.2 Safe refinement: sub-module `SAFE_ZONE`

In order to ensure that the refinement zones properly capture the features of interest, we rely on a safe refinement. It consists in enlarging the marked zone  $\mathcal{M}_i^0$  by adding additional layers of elements whatever the refinement strategy.

While uniform refinement ratio ( $\beta_{T^i}^{0,fix}$  or  $\bar{\beta}_{T^i}^0$ ) is prescribed, layers of elements are added to the set  $\mathcal{M}_i^0$ . While using the adjusted ratio  $\beta_{T^i}^{0,adj}$ , each subset of elements of  $\mathcal{M}_i^0$  having the same prescribed refinement ratio is individually expanded, see Figure 5.

The sub-module `SAFE_ZONE`, detailed in Algorithm 5, outputs the enlarged marked region  $\mathcal{M}_i^*$  as well as the corresponding refinement ratios  $\{\beta_{T^i}^*\}_{T^i \in \mathcal{M}_i^*}$ . Note, that if  $nb_{safe} = 0$  is set, one defines  $\mathcal{M}_i^* := \mathcal{M}_i^0$  and  $\{\beta_{T^i}^*\}_{T^i \in \mathcal{M}_i^*} := \{\beta_{T^i}^0\}_{T^i \in \mathcal{M}_i^0}$ .

**Definition: Vertex-neighbor element.** In Algorithms 5 and 6, an element is called a *vertex-neighbor* to  $T^i$  if it shares at least one vertex with  $T^i$ .

---

**Algorithm 5 SAFE\_ZONE**


---

**Input:** number of layers of elements in the safe zone  $nb_{safe}$ , marked zone  $\mathcal{M}_i^0$ , refinement ratio  $\{\beta_{T^i}^0\}_{T^i \in \mathcal{M}_i^0}$ , current mesh  $\tilde{\mathcal{G}}_i := \mathcal{G}_i$  for  $h$ -adaptive methods or  $\tilde{\mathcal{G}}_i := \hat{\mathcal{G}}_i$  for LDC method

**Output:** enlarged marked region  $\mathcal{M}_i^*$  containing safe zones and corresponding refinement ratio  $\{\beta_{T^i}^*\}_{T^i \in \mathcal{M}_i^*}$

Store in decreasing order unique elements of  $\{\beta_{T^i}^0\}_{T^i \in \mathcal{M}_i^0}$  in a list  $L$

$\mathcal{M}_i^* := \mathcal{M}_i^0$

$\beta_{T^i}^* := \beta_{T^i}^0 \quad \forall T^i \in \mathcal{M}_i^*$

**for**  $\ell = 1, \dots, \#L$  **do**

Set  $L_\ell$  to the  $\ell^{\text{th}}$  value of the list  $L$

$\mathcal{Z}^{L_\ell} := \{T^i \in \mathcal{M}_i^*; \beta_{T^i}^* = L_\ell\}$

$\mathcal{Z}^{-L_\ell} := \{T^i \in \mathcal{M}_i^*; \beta_{T^i}^* < L_\ell\} \cup \{\tilde{\mathcal{G}}_i \setminus \mathcal{M}_i^*\}$

**for**  $e = 1 \dots nb_{safe}$  **do**

$\mathcal{Z}^{*L_\ell} := \{T^i \in \mathcal{Z}^{-L_\ell}; T^i \text{ is a vertex-neighbor to at least one element of } \mathcal{Z}^{L_\ell}\}$

$\beta_{T^i}^* := L_\ell \quad \forall T^i \in \mathcal{Z}^{*L_\ell}$

**Update:**

$\mathcal{Z}^{L_\ell} \leftarrow \mathcal{Z}^{L_\ell} \cup \mathcal{Z}^{*L_\ell}$

$\mathcal{Z}^{-L_\ell} \leftarrow \mathcal{Z}^{-L_\ell} \setminus \mathcal{Z}^{*L_\ell}$

$\mathcal{M}_i^* \leftarrow \mathcal{M}_i^* \cup \mathcal{Z}^{*L_\ell}$

**end for**

**end for**

---

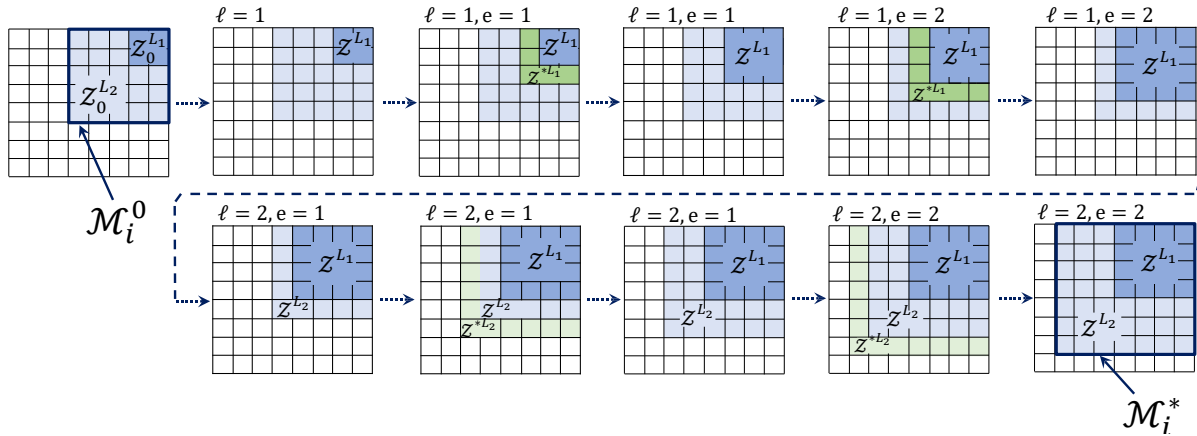


Figure 5: Illustration of the sub-module **SAFE\_ZONE**. Mesh  $\mathcal{G}_i$  and the marked zone  $\mathcal{M}_i^0 := \mathcal{Z}_0^{L_1} \cup \mathcal{Z}_0^{L_2}$  with  $\beta_{T^i \in \mathcal{Z}_0^{L_1}}^0 = L_1$  and  $\beta_{T^i \in \mathcal{Z}_0^{L_2}}^0 = L_2$  with  $L_1 > L_2$ . Hence,  $L = \{L_1, L_2\}$  and  $\ell = 1, \dots, \#L$  refers to the  $\ell^{\text{th}}$  refinement zone (with  $\beta^0 = L_\ell$ ). The index  $e = 1, \dots, nb_{safe}$  indicates the layer being added (here  $nb_{safe} = 2$ ). Finally, the enlarged marked zone  $\mathcal{M}_i^*$  is obtained.

### 2.5.3 Updated refinement ratio: sub-module UPDATE\_RATIO<sub>METHOD</sub>

Due to methodological features of an AMR method, some elements can be affected by the refinement of neighboring elements. In other words, refinement of non initially marked elements, or

modification of some elements refinement ratio may be necessary. Typically, it arises in order to avoid mesh non-conformity (hierarchical conforming  $h$ -adaptivity), or to restrict the mesh irregularity (hierarchical non-conforming  $h$ -adaptivity). The sub-modules **UPDATE\_RATIO**<sub>METHOD</sub> dedicated to each AMR technique considered here aim to output the final marked zone  $\mathcal{M}_i$  and the associated (corrected) refinement ratios.

Naturally, the hierarchical element division relies on an isotropic refinement, aiming to uniformly split an element  $T^i$  in  $\beta_{T^i}^D$  elements. To this end, each edge of element  $T^i$  is uniformly divided in  $\beta_{T^i}$  segments. However, the hierarchical conforming  $h$ -adaptive strategy imposes an anisotropic refinement in some regions which returns to impose non unique refinement ratios to the edges of element  $T^i$ . We hence consider a partition of the edges of an element  $T^i$  in  $D$  groups of edges  $\mathcal{E}_{T^i}^d$ ,  $d = 1, \dots, D$ . Each group of edges is composed by  $2^{D-1}$  edges that do not share any vertex (also called topologically parallel edges in the literature [36, 66]), see Figures 7 and 8.

Thereby, for the sake of generality, we assume a generic formulation for the refinement ratio associated to an element  $T^i$ . We will use a set  $\underline{\beta}_{T^i}$  of  $D$  refinement ratios

$$\underline{\beta}_{T^i} := \{\beta_{T^i}^d\}_{d=1}^D \quad (30)$$

where the refinement ratio  $\beta_{T^i}^d$  is assigned to the  $d^{\text{th}}$  group of edges  $\mathcal{E}_{T^i}^d$ .

### Sub-module UPDATE\_RATIO<sub>LDC</sub>

The philosophy of the LDC method consists in generating (quasi-)uniform local sub-meshes at each refinement step. Thus, only a uniform refinement ratio will be applied to all marked elements of a sub-mesh (cf. fixed uniform or mean adjusted refinement ratios). Thus, the sub-module **UPDATE\_RATIO**<sub>LDC</sub> does not provide any modification of the prior defined marked zone and refinement ratio. We assume

$$\mathcal{M}_i := \mathcal{M}_i^* \quad (31)$$

and

$$\underline{\beta}_{T^i} := \{\beta_{T^i}^d\}_{d=1}^D, \text{ with } \beta_{T^i}^d = \beta_{T^i}^{*,fix} \text{ or } \bar{\beta}_{T^i}^* \quad \forall d = 1, \dots, D \text{ and } \forall T^i \in \mathcal{M}_i \quad (32)$$

### Sub-module UPDATE\_RATIO<sub>NCHAM</sub>

For the non-conforming hierarchical  $h$ -adaptive method (NCHAM) we adopt the “one-irregularity rule” (also called “one-hanging node rule”) argued in [67, 68], which is widely used in the literature. It results in restricting the number of irregular nodes on each edge. Such restriction has numerous advantages: it allows the refinement to be gradual (where neighboring elements are not of greatly differing sizes), to guarantee that each element is in the support of a limited number of basis functions, to reduce the implementation effort, and to simplify computations (e.g., matrix assembly and further mesh refinement). To ensure this, only values of refinement ratio  $\beta_{T^i} = 2^m$ ,  $m \in \mathbb{N}$  may be prescribed. For doing so, we first correct the computed refinement ratios with expression 33. By denoting  $\mathcal{B}_{T^i}$  the element’s total refinement level (cf. equation 34), the “one-irregularity rule” is guaranteed if the ratio between total refinement level of two neighboring elements (greater  $\mathcal{B}_{T^i}$  value over the lower one) is equal to 2 at most. It implies the correction of refinement ratios of elements for which this requirement does not hold. It may potentially result in enlarging the

set of marked elements  $\mathcal{M}_i^*$ . Figure 6 is a step-by-step illustration of Algorithm 2 which provides the implementation details of the non-conforming  $h$ -adaptive method ensuring the previously described “one-irregularity rule”.

---

**Algorithm 6** UPDATE\_RATIO<sub>NCHAM</sub>

---

**Input:** marked zone  $\mathcal{M}_i^*$ , refinement ratios  $\{\beta_{T^i}^*\}_{T^i \in \mathcal{M}_i^*}$ , current mesh  $\mathcal{G}_i$   
**Output:** zone  $\mathcal{M}_i$  to be refined and corresponding refinement ratios sets  $\{\underline{\beta}_{T^i}\}_{T^i \in \mathcal{M}_i}$

**if**  $i = 0$  **then**  $\mathcal{B}_{T^{-1}} := 1$  **end if**

**if** there is one unique value of  $\{\beta_{T^i}^*\}_{T^i \in \mathcal{M}_i^*}$  and  $\mathcal{G}_i \setminus \mathcal{M}_i^* = \emptyset$

$\mathcal{M}_i := \mathcal{M}_i^*$

$\beta_{T^i}^{corr} := \beta_{T^i}^*$

**else**

Set  $\beta_{T^i}^* := 1 \forall T^i \in \mathcal{G}_i \setminus \mathcal{M}_i^*$  and correct refinement ratio:

$$\beta_{T^i}^{corr} := 2^{\min(\text{argmin}|2^m - \beta_{T^i}^*|)}$$
 (33)

Compute total refinement level  $\mathcal{B}_{T^i} \forall T^i \in \mathcal{G}_i$ :

$$\mathcal{B}_{T^i} := \mathcal{B}_{T^{i-1}} \beta_{T^i}^{corr} \quad \text{with } T^i \subset T^{i-1} \text{ and } T^{i-1} \in \mathcal{G}_{i-1}$$
 (34)

Store in decreasing order unique elements of  $\{\mathcal{B}_{T^i}\}_{T^i \in \mathcal{G}_i}$  in a list  $L$

Set  $\ell := 1$

**while**  $\ell \neq \#L$  **do**

$\mathcal{Z}^{L_\ell} := \{T^i \in \mathcal{G}_i; \mathcal{B}_{T^i} = L_\ell\}$

$\mathcal{Z}^{-L_\ell} := \{T^i \in \mathcal{G}_i; \mathcal{B}_{T^i} < L_\ell\}$

$\mathcal{Z}_i^{*L_\ell} := \{T^i \in \mathcal{Z}^{-L_\ell}; T^i \text{ is a vertex-neighbor to at least one element of } \mathcal{Z}^{L_\ell}\}$

**for all**  $T^i \in \mathcal{Z}_i^{*L_\ell}$  **such that**  $\mathcal{B}_{T^i} \neq L_\ell/2$  **do**

$\mathcal{B}_{T^i} := L_\ell/2$  and  $\beta_{T^i}^{corr} := \lceil \mathcal{B}_{T^i} / \mathcal{B}_{T^{i-1}} \rceil$  **for**  $T^i \subset T^{i-1}$  and  $T^{i-1} \in \mathcal{G}_{i-1}$

**if**  $\mathcal{B}_{T^i} \notin L$  **then** insert  $\mathcal{B}_{T^i}$  in the sorted list  $L$  and update  $\#L$  **end if**

$\ell \leftarrow \ell + 1$

**end while**

$\mathcal{M} := \{T^i \in \mathcal{G}_i; \beta_{T^i}^{corr} > 1\}$

**end if**

$\underline{\beta}_{T^i} := \{\beta_{T^i}^d\}_{d=1}^D$ , with  $\beta_{T^i}^d = \beta_{T^i}^{corr} \quad \forall d = 1, \dots, D$  and  $\forall T^i \in \mathcal{M}_i$

---

**Sub-module** UPDATE\_RATIO<sub>CHAM</sub>

The hierarchical conforming  $h$ -adaptive method (CHAM) consists in preserving the mesh conformity within the whole refinement process. Irregular nodes are eliminated by propagating elements subdivision beyond the set of marked elements  $\mathcal{M}_i^*$ . As mentioned before, we rely here on a technique based on sheet operations [33, 34]. More precisely, we adopt a dicing-like approach aiming to restore the mesh conformity through layers refinement.

The general principle of the sub-module UPDATE\_RATIO<sub>CHAM</sub>, detailed in Algorithm 7, is the following. We recursively build layers (or sheets)  $\mathcal{L}_j^d$  of elements associated to the original

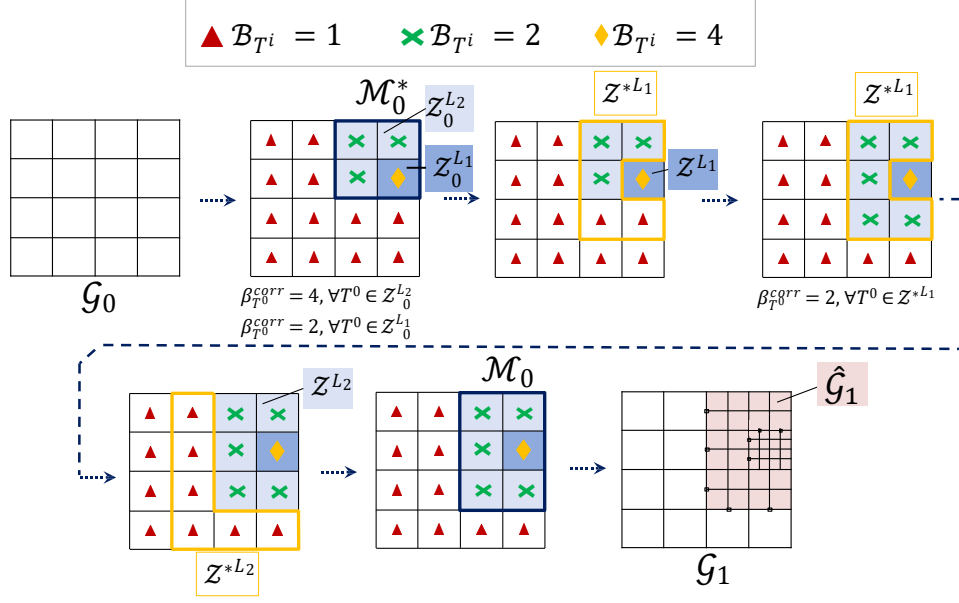


Figure 6: Illustration of the sub-module **UPDATE\_RATIO**<sub>NCHAM</sub>. Initial mesh  $\mathcal{G}_0$ . Marked zone on  $\mathcal{G}_0$ :  $\mathcal{M}_0^* = \mathcal{Z}_0^{L_1} \cup \mathcal{Z}_0^{L_2}$  with  $\mathcal{B}_{T^0 \in \mathcal{Z}_0^{L_1}} = L_1$  and  $\mathcal{B}_{T^0 \in \mathcal{Z}_0^{L_2}} = L_2$  with  $L_1 > L_2$ . Hence,  $L = \{L_1, L_2\}$ . Mesh  $\mathcal{G}_0$  with the zone to be refined  $\mathcal{M}_0$  including transition region, refined mesh  $\mathcal{G}_1$  with the sub-mesh  $\hat{\mathcal{G}}_1$

group of edges  $\mathcal{E}_{T^{*i}}^d$  of the  $j^{\text{th}}$  marked element  $T^{*i}$  in  $\mathcal{M}_i^*$ : all elements sharing at least one edge of  $\mathcal{E}_{T^{*i}}^d$  are added in the layer and the associated groups of edges in these elements are identified. These new groups of edges are then used to extend the layer of elements until no new elements sharing the identified edges can be found. The construction of layers is depicted in Figure 7 for a two-dimensional case and in Figure 8 for a three-dimensional case.

On the identified groups of edges associated to the layer, we assign the maximal refinement ratio of elements  $T^i \in \mathcal{L}_j^d$ . For the sake of optimality, in Algorithm 7 (and illustrated in Figure 7) the set of marked elements  $\mathcal{M}_i^*$  is hence initially sorted by  $\beta_{T^i}^*$  in decreasing order.

Algorithm 7 can be easily performed over any FEM existing solver and conserves the original structure (geometric topology) of the mesh. This algorithm is generic, in particular it can deal with unstructured meshes and self-intersecting layers. However, it is especially interesting for structured Cartesian meshes, as it can be a little simplified since  $k$  involved in set  $\mathcal{N}$  is always equal in this case to  $d$ . The index  $d$  then represents the direction of the mesh (cf. underlying tensor-product mesh), see Figures 7 and 8. Note, that while using such algorithm, the mesh coarsening is straightforward.

#### 2.5.4 Sub-module DIVIDE

The elements refinement is performed with the sub-module **DIVIDE** $(T^i, \underline{\beta}_{T^i})$  which inputs an element  $T^i$  with the corresponding refinement ratio set  $\underline{\beta}_{T^i}$  and outputs the produced refined elements. This sub-module aims to uniformly divide each edge from the groups of edges  $\{\mathcal{E}_{T^i}^d\}_{d=1}^D$  of an element  $T^i$  in  $\beta_{T^i}^d$  ( $d = 1, \dots, D$ ) equal segments. The set of refined elements is then generated by relying the new obtained opposite nodes, making a paving in quadrilateral (2D) or hexahedral (3D) elements of the original element  $T^i$ . The sub-module **DIVIDE** is depicted in Figure 9a for

---

**Algorithm 7 UPDATE\_RATIO<sub>CHAM</sub>**


---

**Input:** marked zone  $\mathcal{M}_i^*$ , refinement ratios  $\{\beta_{T^i}^*\}_{T^i \in \mathcal{M}_i^*}$ , current mesh  $\mathcal{G}_i$

**Output:** zone  $\mathcal{M}_i$  to be refined and corresponding refinement ratios sets  $\{\underline{\beta}_{T^i}\}_{T^i \in \mathcal{M}_i}$

- $\mathcal{M}_i := \mathcal{M}_i^*$
  - Range elements  $T^i$  from  $\mathcal{M}_i^*$  in a set  $\mathcal{S}$  with respect to  $\beta_{T^i}^*$  sorted in the decreasing order
  - Set  $\beta_{T^i}^d := 1 \quad \forall d = 1, \dots, D, \quad \forall T^i \in \mathcal{G}_i \setminus \mathcal{M}_i^*$
  - $\mathcal{F} \leftarrow \emptyset$
  - **for**  $j = 1, \dots, \#\mathcal{S}$ 
    - $T^{*i} := j^{\text{th}}$  element of  $\mathcal{S}$
    - for**  $d = 1, \dots, D$ 
      - if**  $\mathcal{E}_{T^{*i}}^d \notin \mathcal{F}$  **then**
        - $\beta_{T^{*i}}^d := \beta_{T^{*i}}^*$
        - $\mathcal{L}_j^d := T^{*i}$
        - $\mathcal{F}_j^d := \mathcal{E}_{T^{*i}}^d$
        - $\mathcal{N} := \{\mathcal{E}_{T^i}^k; k \in \{1, \dots, D\}, T^i \in \mathcal{G}_i \setminus T^{*i}, \mathcal{E}_{T^i}^k \cap \mathcal{E}_{T^{*i}}^d \neq \emptyset\}$
        - while**  $\mathcal{N} \neq \emptyset$  **do**
          - $\beta_{T^i}^k := \beta_{T^{*i}}^* \quad \forall \{k, T^i\}$  such that  $\mathcal{E}_{T^i}^k \in \mathcal{N}$
          - $\mathcal{F}_j^d \leftarrow \mathcal{F}_j^d \cup \mathcal{N}$
          - $\mathcal{L}_j^d \leftarrow \mathcal{L}_j^d \cup \{T^i \in \mathcal{G}_i \setminus \mathcal{L}_j^d, \mathcal{E}_{T^i}^k \in \mathcal{N}\}$
          - $\mathcal{N} := \{\mathcal{E}_{T^i}^k; k \in \{1, \dots, D\}, \mathcal{E}_{T^i}^k \notin \mathcal{F}_j^d, \mathcal{E}_{T^i}^k \cap \mathcal{F}_j^d \neq \emptyset\}$
        - end while**
        - $\mathcal{M}_i \leftarrow \mathcal{M}_i \cup \mathcal{L}_j^d$
        - $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{F}_j^d$
      - end if**
    - end for**
  - **end for**
- $\underline{\beta}_{T^i} := \{\beta_{T^i}^d\}_{d=1}^D, \quad \forall d = 1, \dots, D$  and  $\forall T^i \in \mathcal{M}_i$
-



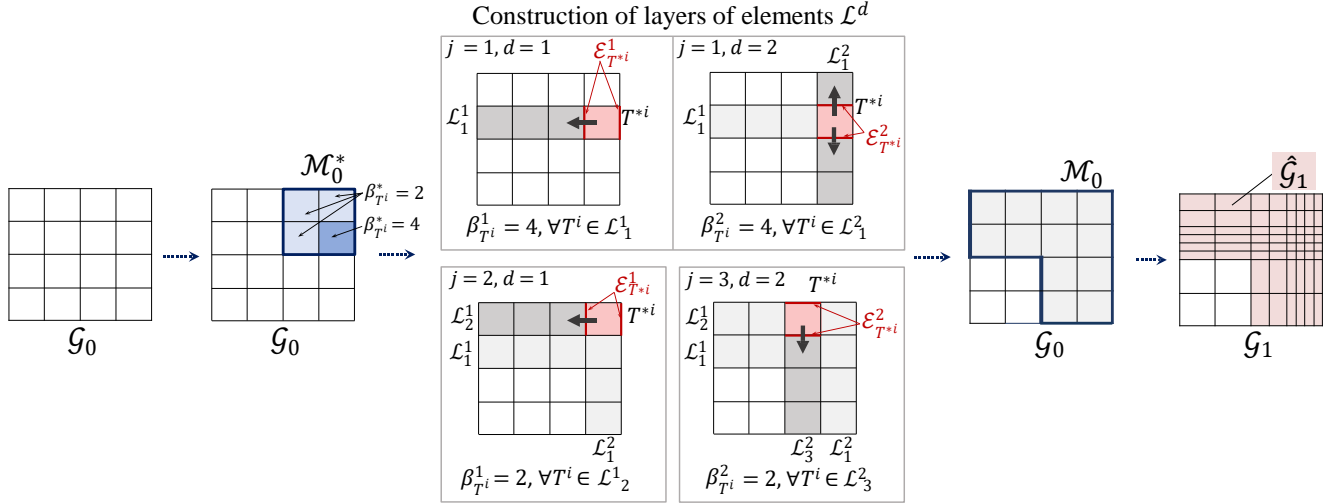


Figure 7: Illustration of the sub-module **UPDATE\_RATIO<sub>CHAM</sub>**. Initial mesh  $\mathcal{G}_0$ . Marked zone on  $\mathcal{G}_0$ :  $\mathcal{M}_0^*$  with  $\beta_{T^i}^*$ . Construction of layers of elements  $\mathcal{L}_j^d$  and correction of refinement ratios assigned to the groups of edges  $\mathcal{E}_{T^i}^d$  of elements  $\forall T^i \in \mathcal{L}_j^d$ . Mesh  $\mathcal{G}_0$  with the zone to be refined  $\mathcal{M}_0$ , refined mesh  $\mathcal{G}_1$  with the sub-mesh  $\hat{\mathcal{G}}_1$

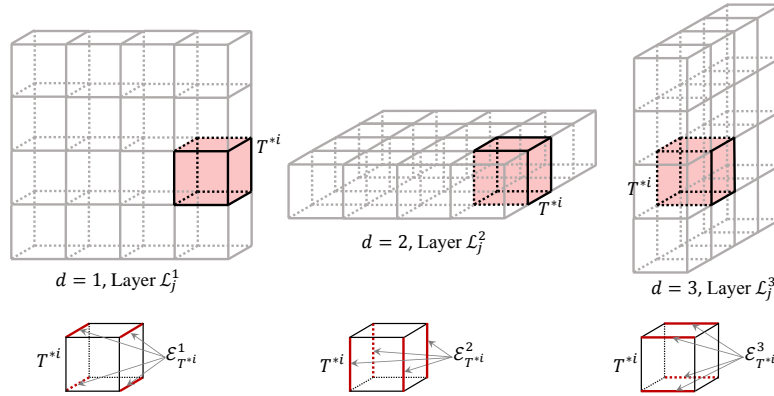


Figure 8: Sketch of layers of elements  $\mathcal{L}_j^d$  and groups of edges  $\mathcal{E}_{T^i}^d$  for a three-dimensional case

the two-dimensional case and in Figure 9b for the three-dimensional case.

The sub-module **DIVIDE** applied for a region is expressed as:

$$\mathbf{DIVIDE}(\mathcal{M}_i, \{\underline{\beta}_{T^i}\}_{T^i \in \mathcal{M}_i}) := \bigcup_{T^i \in \mathcal{M}_i} \mathbf{DIVIDE}(T^i, \underline{\beta}_{T^i}) \quad (35)$$

### 2.5.5 Sub-module NEW\_MESH

The sub-module **NEW\_MESH** outputs the local sub-mesh  $\hat{\mathcal{G}}_{i+1}$  corresponding to the refinement of  $\mathcal{M}_i$ , and the locally refined mesh  $\mathcal{G}_{i+1}$  covering the whole computational domain  $\Omega$  corresponding to the new mesh for  $h$ -adaptive methods and to the composite mesh for LDC method.

$$\mathcal{G}_{i+1}, \hat{\mathcal{G}}_{i+1} := \mathbf{NEW\_MESH}(\mathcal{M}_i, \{\underline{\beta}_{T^i}\}_{T^i \in \mathcal{M}_i}, \mathcal{G}_i, \hat{\mathcal{G}}_i) \quad (36)$$

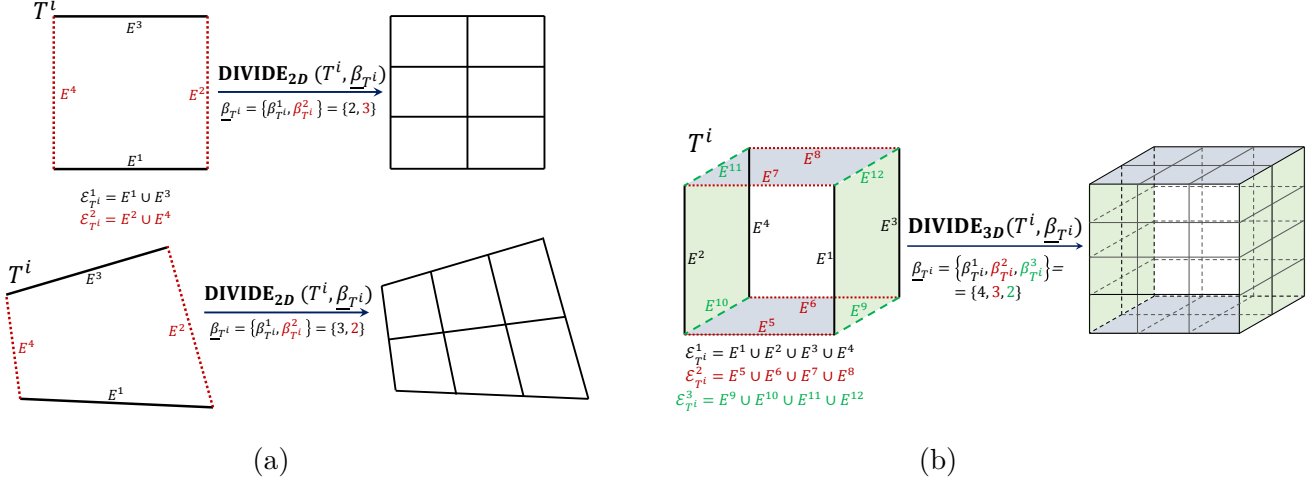


Figure 9: Example for the sub-module  $\mathbf{DIVIDE}(T^i, \underline{\beta}_{T^i})$  for an element  $T^i$  and resulting sets of refined elements  $\cup T^{i+1} \subset T^i$ ; (a)  $\mathbf{DIVIDE}_{2D}$  with refinement ratio set  $\underline{\beta}_{T^i} = \{\beta_{T^i}^1, \beta_{T^i}^2\} = \{2, 3\}$ ; (b)  $\mathbf{DIVIDE}_{3D}$  with refinement ratio set  $\underline{\beta}_{T^i} = \{\beta_{T^i}^1, \beta_{T^i}^2, \beta_{T^i}^3\} = \{4, 3, 2\}$

with the refined sub-mesh  $\hat{\mathcal{G}}_{i+1}$ :

$$\hat{\mathcal{G}}_{i+1} := \mathbf{DIVIDE}(\mathcal{M}_i, \{\underline{\beta}_{T^i}\}_{T^i \in \mathcal{M}_i}) \quad (37)$$

and the global (composite) mesh  $\mathcal{G}_{i+1}$ :

$$\mathcal{G}_{i+1} := \{\mathcal{G}_i \setminus \mathcal{M}_i\} \cup \{\hat{\mathcal{G}}_{i+1}\} \quad (38)$$

**Remark.** In the case of nested refinement zones, the mesh  $\mathcal{G}_{i+1}$  can be defined as:

$$\mathcal{G}_{i+1} := \left\{ \bigcup_{l=0, \dots, i} \hat{\mathcal{G}}_l \setminus \mathcal{M}_l \right\} \cup \{\hat{\mathcal{G}}_{i+1}\} \quad (39)$$

### 3 Numerical experiments

In this section, the AMR approaches considered here are compared on 2D and 3D numerical test cases. The studied problems are implemented through the Cast3M finite elements software [69], developed by CEA (French Alternative Energies and Atomic Energy Commission).

#### 3.1 Post-processing

The achieved global and local accuracies are verified through the following metrics:

- Number of performed refinement iterations  $k$  (maximal number of iterations  $i$  in Algorithm 1)
- **Relative global exact error in energy norm:**

$$e_{\mathcal{G}_k}^{\text{ex}} = \frac{\xi_{\mathcal{G}_k}^{\text{ex}}}{\omega_{\mathcal{G}_k}^{\text{ex}}} \quad (40)$$

with  $\xi_{\mathcal{G}_k}^{\text{ex}}$  and  $\omega_{\mathcal{G}_k}^{\text{ex}}$  computed using Eq.(9) and (11) respectively, with analytical or reference solutions  $\sigma_k^s = \sigma_k^{\text{ex}}$  and  $\varepsilon_k^s = \varepsilon_k^{\text{ex}}$ .

A mesh that satisfies  $\mathbf{e}_{\mathcal{G}_k}^{\text{ex}} \leq \epsilon_\Omega$  is said to be globally acceptable. The globally optimal mesh is the one that fulfills this condition with the minimum number of nodes.

- **Relative local exact error**, evaluated quantitatively with the following measure:

$$\eta = \frac{\mu(\Omega_{\mathcal{D}_k})}{\mu(\Omega)} \quad (41)$$

where  $\mu(\Omega_{\mathcal{D}_k})$  is the measure of the zone  $\Omega_{\mathcal{D}_k} = \{\cup \bar{T}^k; T^k \in \mathcal{D}_k\}$  with  $\mathcal{D}_k$  – the set of elements where the real local relative error  $\mathbf{e}_{T^k}^{\text{ex}}$  exceeds the prescribed error tolerance:  $\mathcal{D}_k = \{T^k \in \mathcal{G}_k; \mathbf{e}_{T^k}^{\text{ex}} > \epsilon_\Omega\}$ . The local relative error is defined as

$$\mathbf{e}_{T^k}^{\text{ex}} = \frac{\xi_{T^k}^{\text{ex}}}{\omega_{T^k}^{\text{ex}}} \quad (42)$$

where  $\xi_{T^k}^{\text{ex}}$  (Eq.(8)) and  $\omega_{T^k}^{\text{ex}}$  (Eq.(12)) are computed with  $\sigma_k^s = \sigma_k^{\text{ex}}$  and  $\varepsilon_k^s = \varepsilon_k^{\text{ex}}$ .

The value  $\eta$  indicates that the local error  $\mathbf{e}_{T^k}^{\text{ex}}$  (Eq.(42)) is respected on  $(100 - \eta)\%$  of the computational domain.

The measure  $\eta$  has to be compared to the user-defined local volume control parameter  $\delta$ . A mesh is said to be locally acceptable if it meets  $\eta \leq \delta$ . The locally optimal mesh is the one that fulfills this condition with the minimum number of nodes.

The following metrics are used to compare the efficiency of the considered AMR methods:

- **Number of nodes:** total number of nodes  $N_{\text{tot}}$  is defined as  $N_{\text{tot}} = N_{\mathcal{G}_k}$  for  $h$ -adaptive methods and  $N_{\text{tot}} = \sum_{i=0}^k N_{\hat{\mathcal{G}}_i}$  for LDC approach, where  $N_*$  refers to the number of nodes of corresponding sub-mesh. Number  $N_{\mathcal{G}_k}$  for the LDC approach refers to the number of nodes of the composite final mesh  $\mathcal{G}_k$ . The number of degrees of freedom (DOFs) is directly proportional to the number of nodes:  $\#\text{DOFs} = D \times N_*$ , with  $D$  being the problem's dimension.
- **Linear system:** maximal size of linear systems to be solved  $((D \times N_{\text{max}}) \times (D \times N_{\text{max}}))$  with  $N_{\text{max}} = \max_i N_{\mathcal{G}_i}$  for  $h$ -adaptive methods and  $N_{\text{max}} = \max_i N_{\hat{\mathcal{G}}_i}$  for LDC approach. As unrefinement is not allowed during refinement steps,  $N_{\text{max}} = N_{\mathcal{G}_k}$  for  $h$ -adaptive techniques. Number of hanging nodes is explicitly given for the non-conforming  $h$ -adaptive method.
- **CPU time:** total runtime including the construction of new global meshes for  $h$ -adaptive methods or local sub-meshes for the LDC approach, the definition and resolution of problems on the generated refined meshes. Time consumptions for problems resolution (module **SOLVE**), error estimation and marking stage (modules **ESTIMATE** and **MARK**) and mesh refinement itself (module **REFINE**) are explicitly given for each AMR technique.

The following global error thresholds (Eq.(20)) are studied in this paper:  $\epsilon_\Omega = 4\%, 2\%, 1\%, 0.5\%$ . The volume control parameter  $\delta$  (Eq.(21)) is set to 3% for all numerical examples.

Linear elastostatic framework is considered for all numerical examples treated in this paper. Denoting  $\boldsymbol{\sigma}$  and  $\boldsymbol{\varepsilon}$  the stress and strain tensors,  $E$  the Young's modulus and  $\nu$  the Poisson coefficient, the problem reads:

$$\begin{cases} \operatorname{div} \boldsymbol{\sigma}(\mathbf{u}) = 0 & \text{in } \Omega \\ \boldsymbol{\sigma}(\mathbf{u}) = \frac{E}{1+\nu} \left( \boldsymbol{\varepsilon}(\mathbf{u}) + \frac{\nu}{1-2\nu} \operatorname{tr}(\boldsymbol{\varepsilon}(\mathbf{u})) \mathbb{I} \right) & \text{in } \Omega \\ \boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2} (\operatorname{grad} \mathbf{u} + \operatorname{grad}^T \mathbf{u}) & \text{in } \Omega \\ \text{Boundary conditions} & \text{on } \partial\Omega \end{cases} \quad (43)$$

with “tr” the trace operator and  $\mathbb{I}$  the identity matrix.

For this comparison study, each associated discrete linear problem ( $\mathcal{P}_i$ ) defined either on the global mesh  $\mathcal{G}_i$  or on each sub-mesh  $\hat{\mathcal{G}}_i$ , is solved with the same direct linear solver (based here on a Crout factorization). The use of the same linear solver for systems arising from the three AMR methods allows the comparison to be as fair as possible. For the non-conforming  $h$ -adaptive approach, a static condensation of Lagrange multipliers and associated hanging unknowns is used to avoid a saddle-point structure problem.

For all numerical examples the obtained results are post-processed in the same way. We verify the fulfillment of the global and the local conditions for all considered AMR methods, mesh optimality criteria and refinement ratios. To demonstrate the distribution of the local error (42) and to identify critical regions  $\mathcal{D}_k$ , the local error maps are shown for each AMR method, mesh optimality criteria and certain refinement ratios. Note, that for the LDC method, the local error obtained after convergence of the iterative process are presented on each sub-mesh  $\{\hat{\mathcal{G}}_i\}_{i=0}^k$  separately and on the composite mesh  $\mathcal{G}_k$ . For comparison purposes, results obtained with hierarchical uniform refinements are also provided. Performances of the studied AMR methods with respect to the aforementioned metrics are then discussed. We determine also the most efficient refinement strategy in terms of the mesh optimality criteria and refinement ratio.

### 3.2 Plate with circular hole

The first numerical example is the well-established benchmark problem of an infinite plate with a circular hole subject to a prescribed loading  $\boldsymbol{\sigma}_\infty = 1 \text{ Pa}$  in the  $\mathbf{e}_1$  direction (cf. Figure 10). The material is supposed to be linear elastic with Young's modulus  $E = 10^3 \text{ Pa}$  and Poisson's ratio  $\nu = 0.3$ . For symmetry reasons, only a quarter of the plate with  $a = 10 \text{ m}$  and radius  $R = 1 \text{ m}$  is considered. Dirichlet boundary conditions expressing the symmetry ( $\mathbf{u} = 0$ ) are prescribed on edges  $I$  and  $IV$ , while tractions computed from the exact solution ( $\boldsymbol{\sigma}\mathbf{n}$ ), cf. expression (44), are imposed on edges  $II$  and  $III$ , see Figure 10-right. The analytical solution of this problem has been pro-

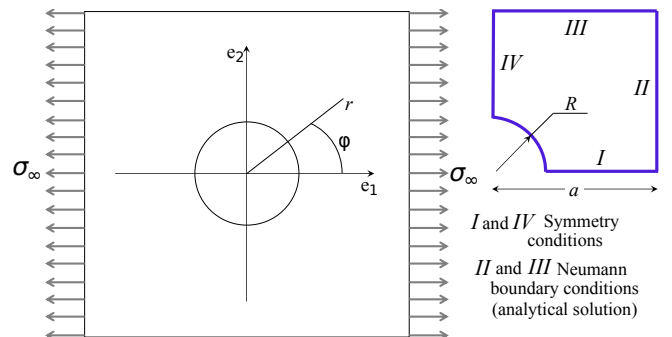


Figure 10: Infinite plate with central circular hole. Left: Sketch of the problem; Right: Computational domain and boundary conditions

vided in [39] and in polar coordinates  $(r, \varphi)$  reads:

$$\begin{aligned}\sigma_{11}(r, \varphi) &= \sigma_{\infty} \left( 1 - \frac{R^2}{r^2} \left( \frac{3}{2} \cos 2\varphi + \cos 4\varphi \right) + \frac{3 R^4}{2 r^4} \cos 4\varphi \right) \\ \sigma_{12}(r, \varphi) = \sigma_{21}(r, \varphi) &= \sigma_{\infty} \left( -\frac{R^2}{r^2} \left( \frac{1}{2} \sin 2\varphi + \sin 4\varphi \right) + \frac{3 R^4}{2 r^4} \sin 4\varphi \right) \\ \sigma_{22}(r, \varphi) &= \sigma_{\infty} \left( -\frac{R^2}{r^2} \left( \frac{1}{2} \cos 2\varphi - \cos 4\varphi \right) - \frac{3 R^4}{2 r^4} \cos 4\varphi \right)\end{aligned}\quad (44)$$

This analytical solution is used in this numerical example for error monitoring.

We rely on a hierarchical refinement which implies that the initial approximation of the hole is not improved. In view of the results, the geometry approximation error is very low compared to the discretization error. However, if we are interested in very low error tolerances (e.g.  $\epsilon_{\Omega} \leq 0.3\%$ ), the hole approximation has to be updated during the refinement process (i.e., quasi-hierarchical strategy, see [17]), see Figure 11.

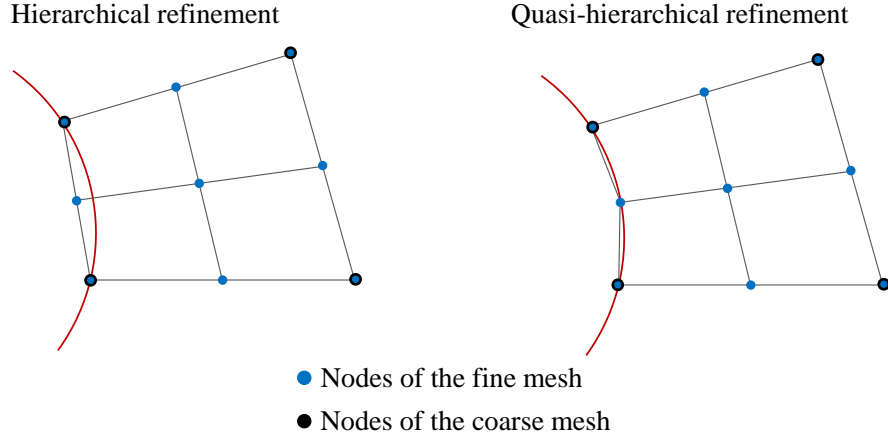


Figure 11: Illustration of hierarchical and quasi-hierarchical mesh refinement strategies

Figure 12 presents sub-meshes generated with the LDC method for different global error tolerances  $\epsilon_{\Omega}$ . Quantitative-based marking strategy naturally yields nested refinement zones. One can remark in Figure 12 that the sequence of meshes varies with the prescribed tolerance. For the chosen example, it can be seen that by fixing  $\epsilon_{\Omega} = 4\%$ , a real error  $\mathbf{e}_{\mathcal{G}_k}^{\text{ex}} = 2.63\%$  is reached with 2 sub-meshes. To satisfy  $\epsilon_{\Omega} = 2\%$ , larger refinement zones and an additional sub-mesh are automatically generated which lead to a real error  $\mathbf{e}_{\mathcal{G}_k}^{\text{ex}} = 1.35\%$ .

**Local error maps.** The generation of the series of refined meshes  $\{\mathcal{G}_i\}_{i=0}^k$  starts from the same initial coarse mesh (as uniform as possible)  $\mathcal{G}_0$ . The relative local error maps (local error distribution  $\mathbf{e}_{T^k}^{\text{ex}}$ ) obtained for the error tolerance  $\epsilon_{\Omega} = 2\%$  with the OB and LOC optimality criteria are reported in Figures 13 and 14, respectively, for various refinement ratios. It can be seen that the meshes generated automatically by the three AMR methods are similar, with refined elements concentrated around the central hole, as expected. Moreover, the sets of elements  $\mathcal{D}_i$  are also more and more localized around the hole. For each case, values of the relative global error  $\mathbf{e}_{\mathcal{G}_k}^{\text{ex}}$  (Eq.(40)) as well as the local error measure  $\eta$  (Eq.(41)) are explicitly provided. For a given AMR technique, the use of the considered refinement ratios (fixed, adjusted and mean adjusted ratios) leads to similar results in terms of the  $\mathbf{e}_{\mathcal{G}_k}^{\text{ex}}$  and  $\eta$  values. It can be noted that the OB and LOC optimality criteria both allow us to reach the prescribed accuracies.

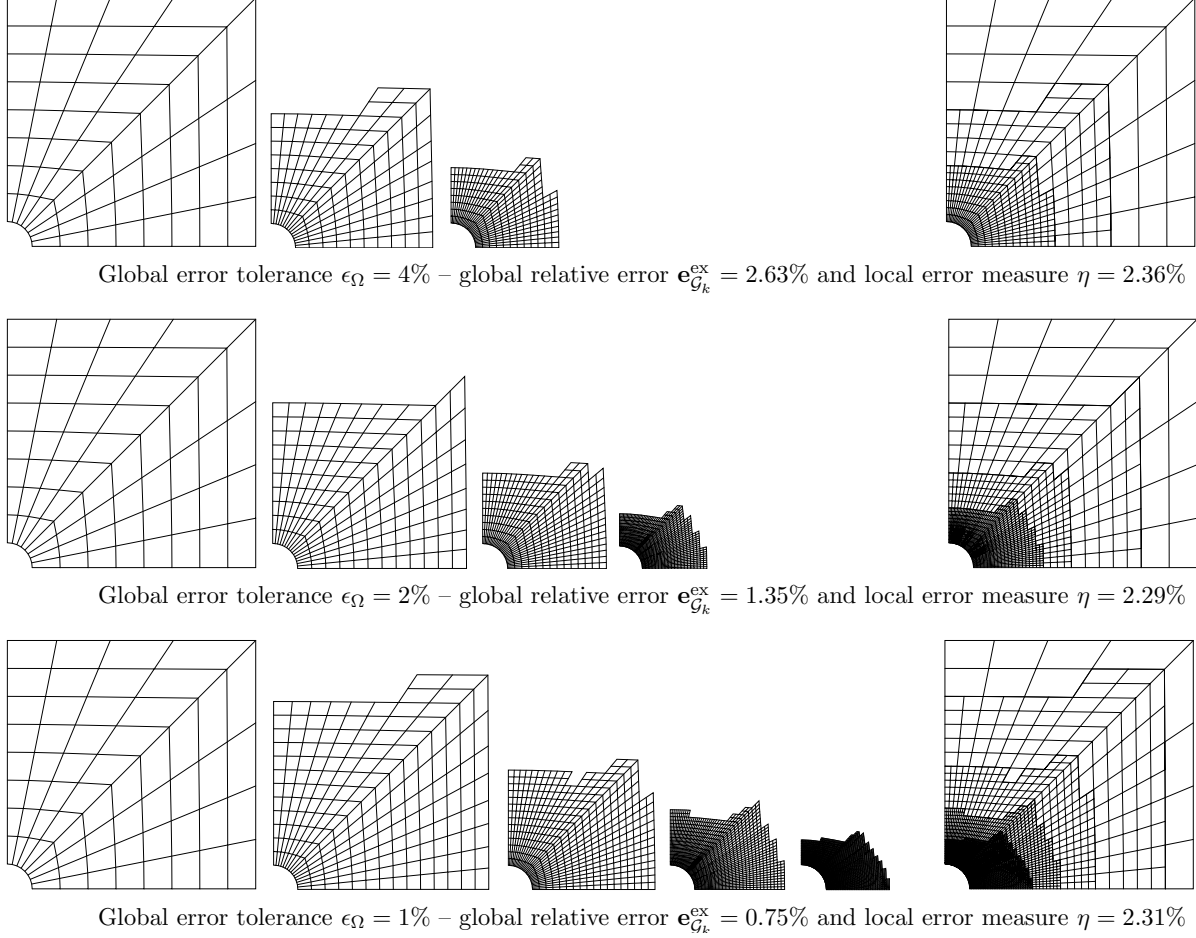


Figure 12: Infinite plate with circular hole – LOC optimality criterion – levels of sub-meshes  $\{\hat{\mathcal{G}}_i\}_{i=0}^k$  and composite grid  $\mathcal{G}_k$  generated with the LDC method for different global error tolerances  $\epsilon_\Omega$

**Local error verification.** The comparison of the three mesh optimality criteria is performed with regard to the local error. Values of the local error measure  $\eta$  are illustrated in Figure 15 for all refinement ratios, mesh optimality criteria and AMR methods. It can be seen that the OB and LOC criteria both provide locally acceptable meshes satisfying  $\eta \leq \delta$ , independently of the AMR technique and refinement ratio strategies.

**Fixed refinement ratio.** Figures 16a and 16b report evolution of the global relative error  $e_{\mathcal{G}_k}^{\text{ex}}$  in energy norm with respect to the number of nodes  $N_{\mathcal{G}_k}$  for OB and LOC criterion, respectively.

We see that all the considered refinement techniques, including the hierarchical uniform refinement, have the same convergence rates close to the optimal one in  $\mathcal{O}(N_{\mathcal{G}_k}^{-1/2})$ . Moreover, the optimal convergences are reached already for relatively small numbers of nodes. It can be noted that the non-conforming h-adaptive and LDC methods seems offer here an improvement of the convergence in the pre-asymptotic regime (in this case occurring early, before  $N_{\mathcal{G}_k} = 1 \cdot 10^3$ ). Note that the hierarchical uniform refinement yields an (almost) optimal convergence ( $\mathcal{O}(h) \approx \mathcal{O}(N_{\mathcal{G}_k}^{-1/2})$ ) due to the lack of singularities in the numerical example.

The moderate degradation of the slope for the smallest tolerances is due to the coarse approximation of the hole, as confirmed by Figure 16c which provides the convergence in number of nodes

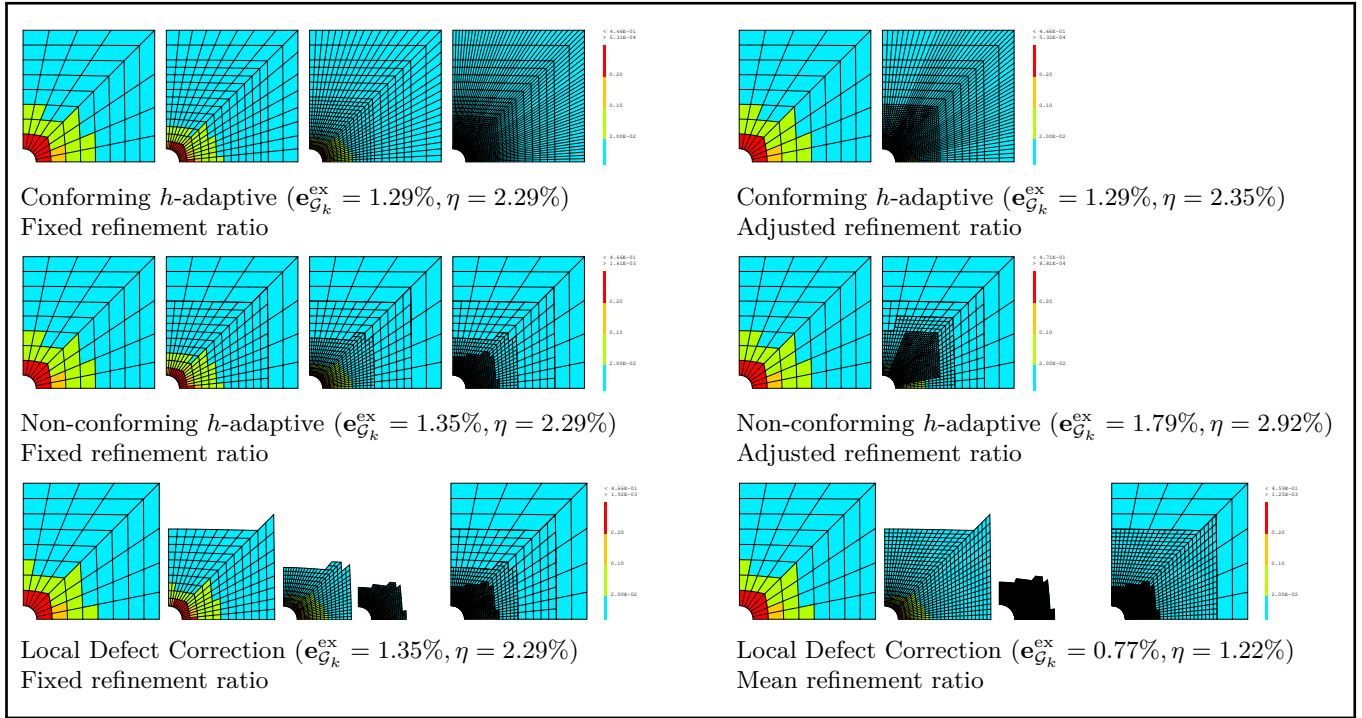


Figure 13: Infinite plate with circular hole – local error maps – OB optimality criterion – tolerance on the global error  $\epsilon_{\Omega} = 2\%$  – local volume control parameter  $\delta = 3\%$

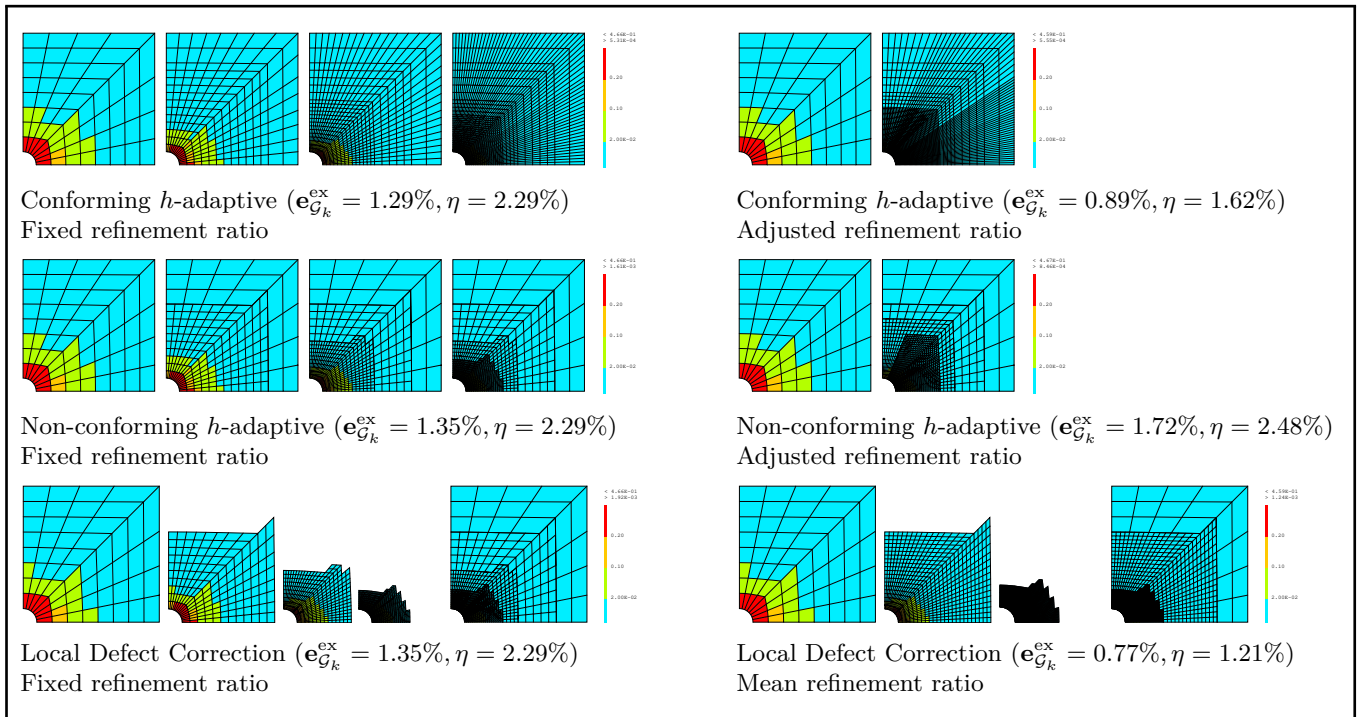


Figure 14: Infinite plate with circular hole – local error maps – LOC optimality criterion – tolerance on the global error  $\epsilon_{\Omega} = 2\%$  – local volume control parameter  $\delta = 3\%$

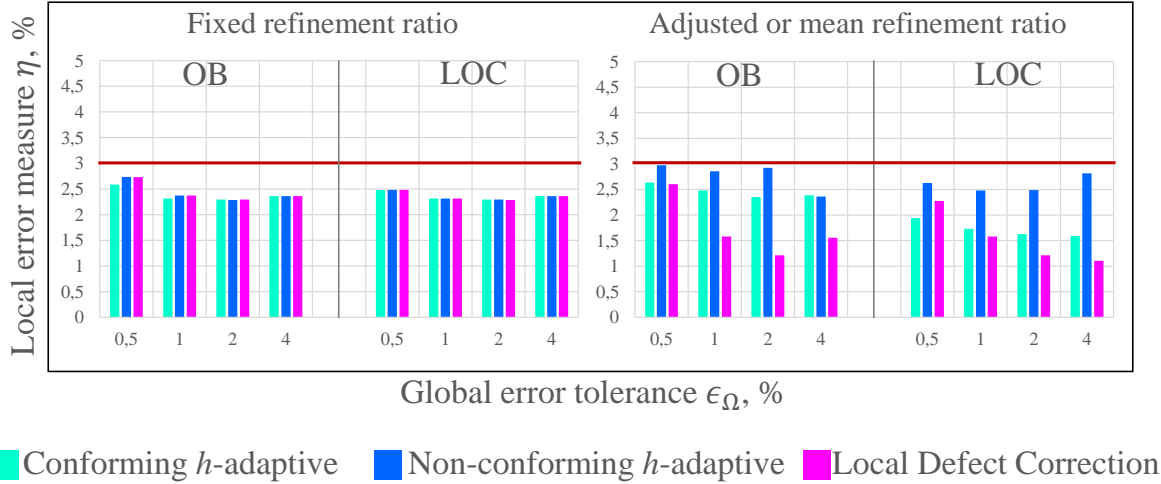


Figure 15: Infinite plate with circular hole – local error measure  $\eta$  vs global error tolerance  $\epsilon_\Omega$

obtained by performing a quasi-hierarchical mesh refinement. In latter case, the hole’s approximation is improved within the refinement process, see Fig 11. The optimal convergence rates for all AMR methods and uniform refinement are then recovered.

Figure 17 depicts the global relative error  $e_{\mathcal{G}_k}^{\text{ex}}$  in energy norm with respect to the total runtime for all user-prescribed tolerances  $\epsilon_\Omega$ . Globally, for all prescribed error tolerances  $\epsilon_\Omega$ , the OB and LOC criteria lead to the similar outcomes in terms of the total CPU time. It can be observed, that all AMR method enable to satisfy the prescribed global accuracy with lower CPU time, compared to a hierarchical uniform global refinement. Moreover, the LDC approach clearly appears to be the most efficient AMR strategy.

Table 1: Infinite plate with circular hole – Fixed refinement ratio

$\epsilon_\Omega = 0.5\%$ , $\delta = 3\%$	Uniform refinement	Conforming $h$ -adaptive		Non-conforming $h$ -adaptive		Local Defect Correction	
		OB	LOC	OB	LOC	OB	LOC
$k$	5	5	5	5	5	5	5
$e_{\mathcal{G}_k}^{\text{ex}}, \%$	0.40	0.41	0.41	0.46	0.46	0.46	0.46
$\eta, \%$	2.4	2.6	2.4	2.7	2.5	2.7	2.5
$N_{\mathcal{G}_k}$	66049	24415	26471	13134	13257	13421	13555
$N_{\text{tot}}$						17712	17880
$N_{\text{max}}$						10429	10543
$N_{\text{hanging\_nodes}}$				285	296		
CPU total	50.1s	33.8s	36.2s	8.4s	8.5s	4.8s	4.8s
CPU (SOLVE)	9.7s	3.0s	3.2s	4.7s	4.8s	1.2s	1.3s
CPU (EST.+MARK)	-	2.2s	2.9s	1.0s	1.0s	1.0s	1.0s
CPU (REFINE)	40.4s	28.6s	30.1s	2.7s	2.7s	2.6s	2.5s

In Table 1 we provide a more detailed performance comparison of the refinement methods for the tolerance  $\epsilon_\Omega = 0.5\%$ . Firstly, we can see that all AMR methods allow us to reach more efficiently the global and local errors similar to those obtained with a hierarchical uniform refinement. The following main differences between AMR techniques can be noted. For the given tolerance, LDC



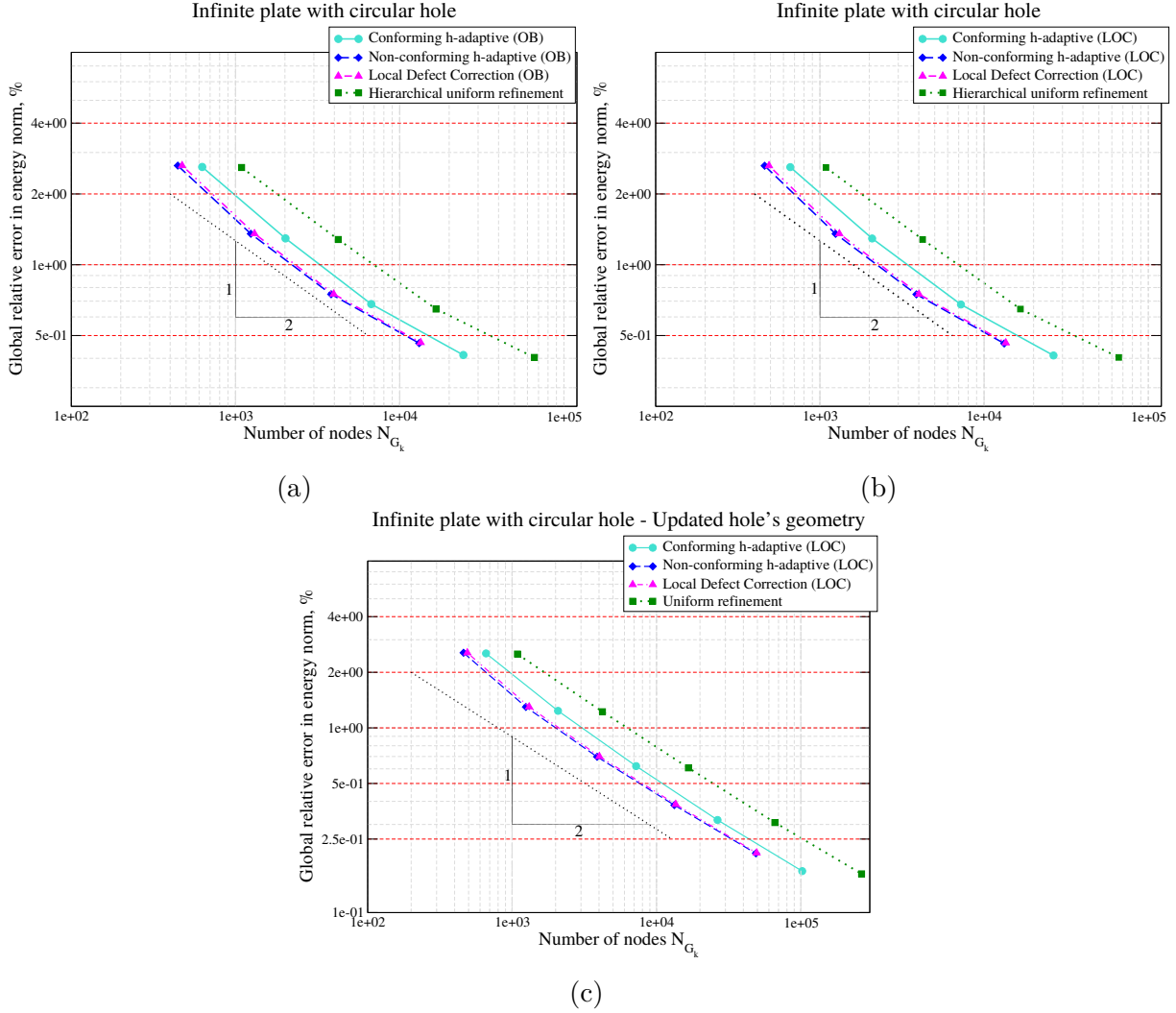


Figure 16: Infinite plate with circular hole – Fixed refinement ratio; Relative global error  $e_{G_k}^{\text{ex}}$  in energy norm vs number of nodes  $N_{G_k}$ : (a) OB optimality criterion, (b) LOC optimality criterion and (c) LOC optimality criterion with updated hole's approximation

approach is less time consuming compared to  $h$ -adaptive techniques and leads to similar numbers of nodes  $N_{G_k}$  as hierarchical non-conforming strategy. However, the total number of stored nodes  $N_{\text{tot}}$  is in between the numbers of nodes for non-conforming and conforming  $h$ -refinement methods. The maximal size of systems of equations confronted with LDC approach is smaller than with  $h$ -adaptive methods, which can provide high cost reduction, especially when direct solvers are used.

**Adjusted and mean adjusted refinement ratio.** We provide here performance results of the two  $h$ -adaptive methods with the adjusted refinement ratio and LDC method with the mean adjusted refinement ratio. Figures 18a and 18b present the global relative error  $e_{G_k}^{\text{ex}}$  with respect to the number of nodes  $e_{G_k}^{\text{ex}}$  for OB and LOC criterion, respectively. Globally, the reached convergence rates are close to those obtained with the fixed progressive refinement ratio. The (quasi) optimal rates are recovered with all AMR methods, independently of the mesh optimality criterion applied.

Figure 19 presents the global relative error  $e_{G_k}^{\text{ex}}$  with respect to the total runtime. We observe that the use of adjusted refinement ratios for  $h$ -adaptive methods results in similar (non-conforming

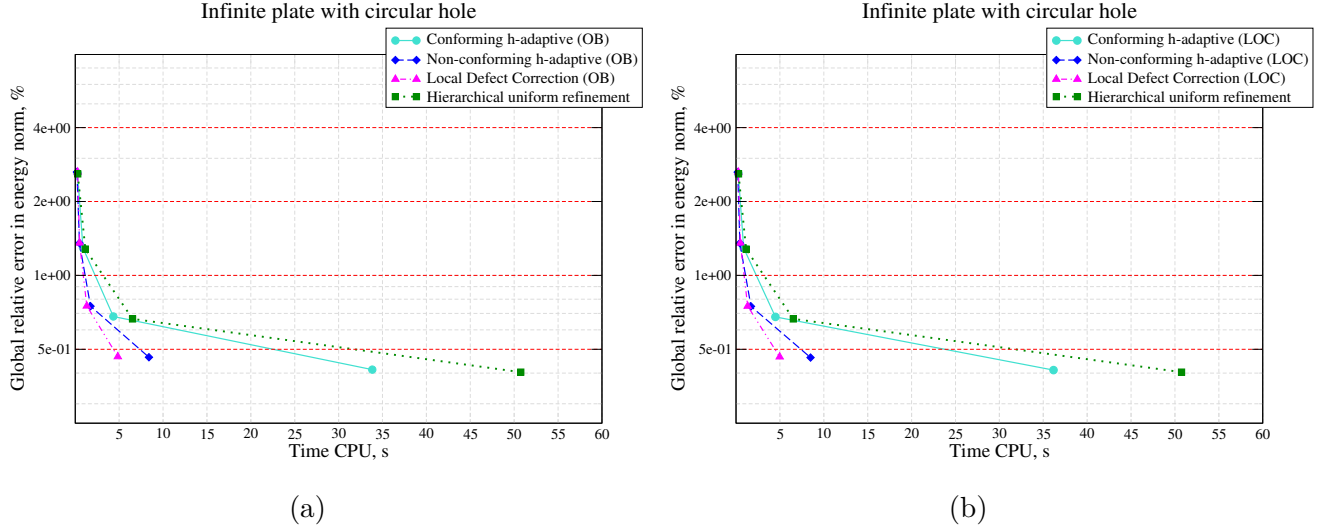


Figure 17: Infinite plate with circular hole – Fixed refinement ratio; Relative global error  $e_{\mathcal{G}_k}^{\text{ex}}$  in energy norm vs total CPU time: (a) OB optimality criterion and (b) LOC optimality criterion

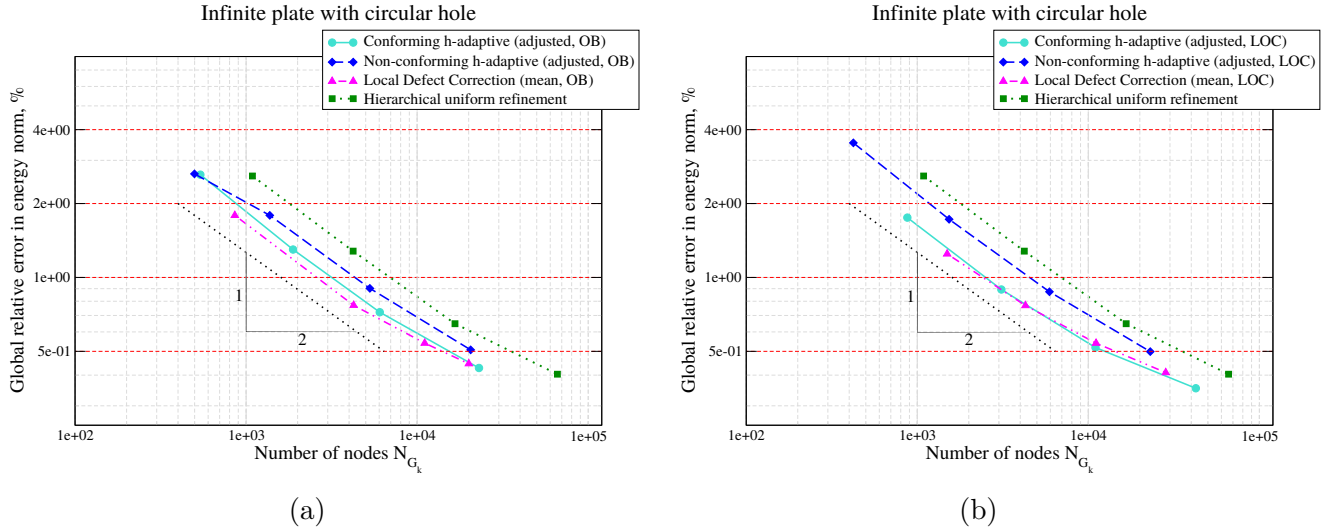


Figure 18: Infinite plate with circular hole – Adjusted and mean adjusted refinement ratios; Relative global error  $e_{\mathcal{G}_k}^{\text{ex}}$  in energy norm vs number of nodes  $N_{\mathcal{G}_k}$ ; (a) OB optimality criterion, (b) LOC optimality criterion

technique) or much lower (conforming technique) CPU times compared to the fixed refinement ratio strategy. In fact, if only one refinement iteration is sufficient to reach the prescribed accuracies, the use of adjusted refinement ratio can be beneficial in terms of CPU time, even if more refined meshes are generated. It worth to mention that resolution times are similar between fixed and adjusted/mean refinement strategies and the main difference relies in mesh refinement. Indeed, the key advantage of the adjusted refinement ratio is the possibility to reduce the number of refinement iterations, limiting the computational time for the hierarchical mesh refinement. It can be noticed that the non-conforming  $h$ -adaptive technique becomes more time consuming compared to the conforming strategy, certainly due to the high number of hanging nodes to be handled. We

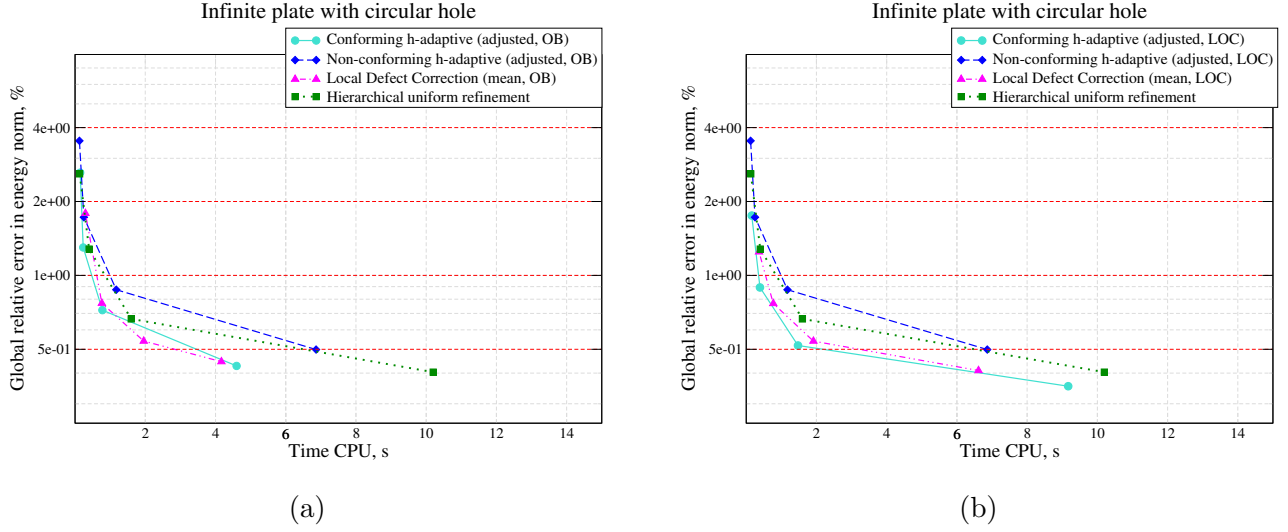


Figure 19: Infinite plate with circular hole – Adjusted and mean refinement ratios; Relative global error  $e_{\mathcal{G}_k}^{\text{ex}}$  in energy norm vs total CPU time: (a) OB optimality criterion and (b) LOC optimality criterion

observe that LOC optimality criterion leads to slightly higher total runtimes, at the same time, naturally providing lower global and local errors, compared to the OB criterion.

Table 2 details results for  $\epsilon_\Omega = 0.5\%$ . It can be seen that in nearly every case, more refined meshes are generated without significant improvement of global and local errors, compared to the fixed ratio. Indeed, for  $\epsilon_\Omega = 0.5\%$ , the final meshes are nearly twice more refined compared to those obtained with the fixed ratio refinement. It has to be noted that in this case, more refined meshes are generated even with more progressive mean adjusted refinement applied to the LDC method. It can be explained by the fact that the adjusted refinement ratio may be excessively high and sometimes inappropriate to meet the required precision.

Table 2: Infinite plate with circular hole – Adjusted refinement ratio ( $h$ -adaptive methods), Mean adjusted refinement ratio (Local Defect Correction)

$\epsilon_\Omega = 0.5\%, \delta = 3\%$	Uniform refinement	Conforming $h$ -adaptive		Non-conforming $h$ -adaptive		Local Defect Correction	
		OB	LOC	OB	LOC	OB	LOC
$k$	1 ( $\beta_{Ti} = 32$ )	1	1	1	1	2	2
$e_{\mathcal{G}_k}^{\text{ex}}, \%$	0.40	0.43	0.35	0.51	0.49	0.41	0.40
$\eta, \%$	2.4	2.6	1.9	3.0	2.6	2.6	2.3
$N_{\mathcal{G}_k}$	66049	22990	42504	20549	23013	20041	28350
$N_{\text{tot}}$						20850	29165
$N_{\text{max}}$						18476	26791
$N_{\text{hanging nodes}}$				301	263		
CPU total	10.2s	4.6s	9.2s	5.7s	6.8s	4.2s	6.6s
CPU (SOLVE)	8.5s	2.7s	4.4s	3.5s	4.6s	1.8s	3.3s
CPU (EST.+MARK)	-	1.1s	3.6s	1.4s	1.4s	1.6s	2.5s
CPU (REFINE)	1.7s	0.8s	1.2s	0.8s	0.8s	0.8s	0.8s

For the sake of pertinent comparative study, the mean adjusted refinement ratio has been also

tested on  $h$ -adaptive techniques. For this numerical example, it results in more costly computations, in terms of both, number of nodes and CPU times, compared to the fixed and adjusted ratio strategies.

**General remarks.** The efficiency of LDC method in term of CPU time versus number of nodes clearly appears independently of the mesh optimality criterion used, especially for the fixed ratio refinement strategy. In this case, the same level of accuracy is reached with a total CPU time 50 to 80% less compared to the conforming  $h$ -adaptive approach, and 30 to 60% less compared to the non-conforming  $h$ -adaptive method. Progressive refinement with mean adjusted ratio applied for LDC method competes even the direct refinement (in one iteration) possible with  $h$ -adaptive methods.

For this academic numerical example, it has been observed that OB and LOC criteria both permit to construct locally and globally acceptable meshes for each prescribed tolerance. We also conclude that final meshes generated with the adjusted or mean adjusted refinement ratios are not optimal in terms of the number of nodes. Concerning the CPU time, the application of the adjusted refinement ratio to  $h$ -adaptive techniques is beneficial, since only one refinement iteration is sufficient to reach the prescribed accuracy.

### 3.3 Industrial test case: mechanical Pellet-Cladding Interaction

The so-called Pellet-Cladding interaction (PCI) is part of a wide range of physical and mechanical phenomena occurring in Pressurized Water Reactor (PWR) during irradiation. In  $2D$ , two test cases are usually used to separately model the two phenomena (namely the hourglass deformation and the pellets fragmentation) characterizing the PCI, as proposed in [27, 15, 42]. In  $3D$ , these two phenomena are combined on a three-dimensional geometry. The cladding's material is supposed to be linear elastic with Young's modulus  $E = 10^{11}$  Pa and Poisson ratio  $\nu = 0.3$ .

Problem definitions and boundary conditions of both,  $2D$  and  $3D$  test cases, are depicted in Figure 20. We prescribe symmetry conditions due to the partial modeling of the cladding

$$\mathbf{u} \cdot \mathbf{n} = \mathbf{0} \text{ on the surfaces } \theta = 0, \theta = \pi/8 \text{ and } z = 0 \quad (45)$$

and an uniform normal displacement

$$\mathbf{u} \cdot \mathbf{n} = \text{const. on } z = L/2, \forall \theta \text{ and } \forall r \quad (46)$$

Internal discontinuous pressure representing the contact with the cracked pellet are prescribed on  $r = R_{\text{int}} = 4.1\text{mm}$ :

$$\begin{aligned} \boldsymbol{\sigma} \mathbf{n} &= 0 \text{ MPa} && \text{on } 0 \leq \theta \leq \theta_f, \forall z \\ \boldsymbol{\sigma} \mathbf{n} &= -150 \text{ MPa} && \text{on } \theta_f < \theta \leq \pi/8 \text{ and } z \leq L_h \\ \boldsymbol{\sigma} \mathbf{n} &= -80 \text{ MPa} && \text{on } \theta_f < \theta \leq \pi/8 \text{ and } z > L_h \end{aligned} \quad (47)$$

External pressure is imposed on  $r = R_{\text{ext}} = 4.7\text{mm}$ :

$$\boldsymbol{\sigma} \mathbf{n} = -15.5 \text{ MPa } \forall \theta, \forall z \quad (48)$$

In the numerical application,  $\theta_f = \arctan(\frac{f}{R_{\text{int}}})$  with  $f = 8\mu\text{m}$ ,  $L_h = 0.6\text{mm}$ .

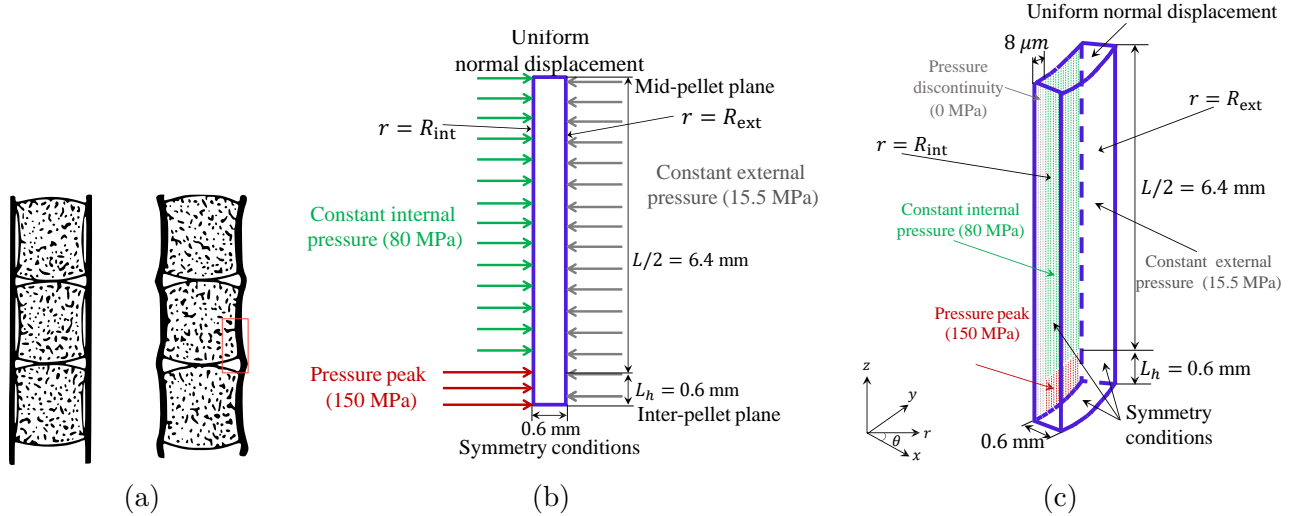


Figure 20: Industrial test case: Mechanical Pellet-Cladding Interaction; (a) Hourglass shape deformation; Computational domain and boundary conditions for (b) bi-dimensional axisymmetric test case and (c) three-dimensional test case

### 3.3.1 Bi-dimensional axisymmetric case

The 2D case studied here consists in modeling the cladding's response to the pellet's hourglass deformation, sketched in Figure 20a. This case can be seen as a pre-study for the three-dimensional more realistic test case. We focus here on determining the most efficient refinement strategy in terms of mesh optimality criterion and refinement ratio in presence of local singularity. The contact is modeled as a discontinuous pressure applied on the cladding's internal part (cf. Figure 20b). For symmetry reasons, only a half of the pellet's height is represented. In order to allow an overall normal displacement of the cladding, an uniform normal (*a priori* unknown) translation condition is assumed at the mid-pellet plane (top of the modeled domain). This test case reveals a local singularity in stress (of order 0.5) near the pressure discontinuity zone, see [42]. A bi-dimensional axisymmetric formulation is used (cf. Figure 20b).

Since the analytical solution of this problem is unknown, as the reference solution we here consider the one obtained with a very fine uniform discretization adapted to the pressure discontinuity with a mesh-step of approximately  $2.8 \mu\text{m}$  ( $\simeq 1 \cdot 10^6$  nodes).

**Local error maps.** The relative local error maps obtained using OB and LOC criteria for  $\epsilon_\Omega = 2\%$  are presented in Figure 21.

As we can see, whatever the AMR method, similar regions are refined. It can be noticed that regardless the AMR method, mesh optimality criterion and refinement ratio, the desired local accuracy ( $\eta \leq \delta$ ) is reached for this error threshold. In this numerical example, the LOC mesh optimality criterion leads to locally more precise simulations (lower  $\eta$  values) compared to OB criterion. One can observe that for hierarchical non-conforming  $h$ -refinement and LDC methods, the local accuracy may be not satisfied on some elements which have not been marked within the refinement iterations. It occurs mostly due to the under-estimation of the real error by the ZZ error estimator. Such an issue is compensated by the use of more local LOC mesh optimality criterion, resulting in the improvement of the local error's measure  $\eta$ .

**Local error verification.** The local error measure  $\eta$  is reported in Figure 22. For this test

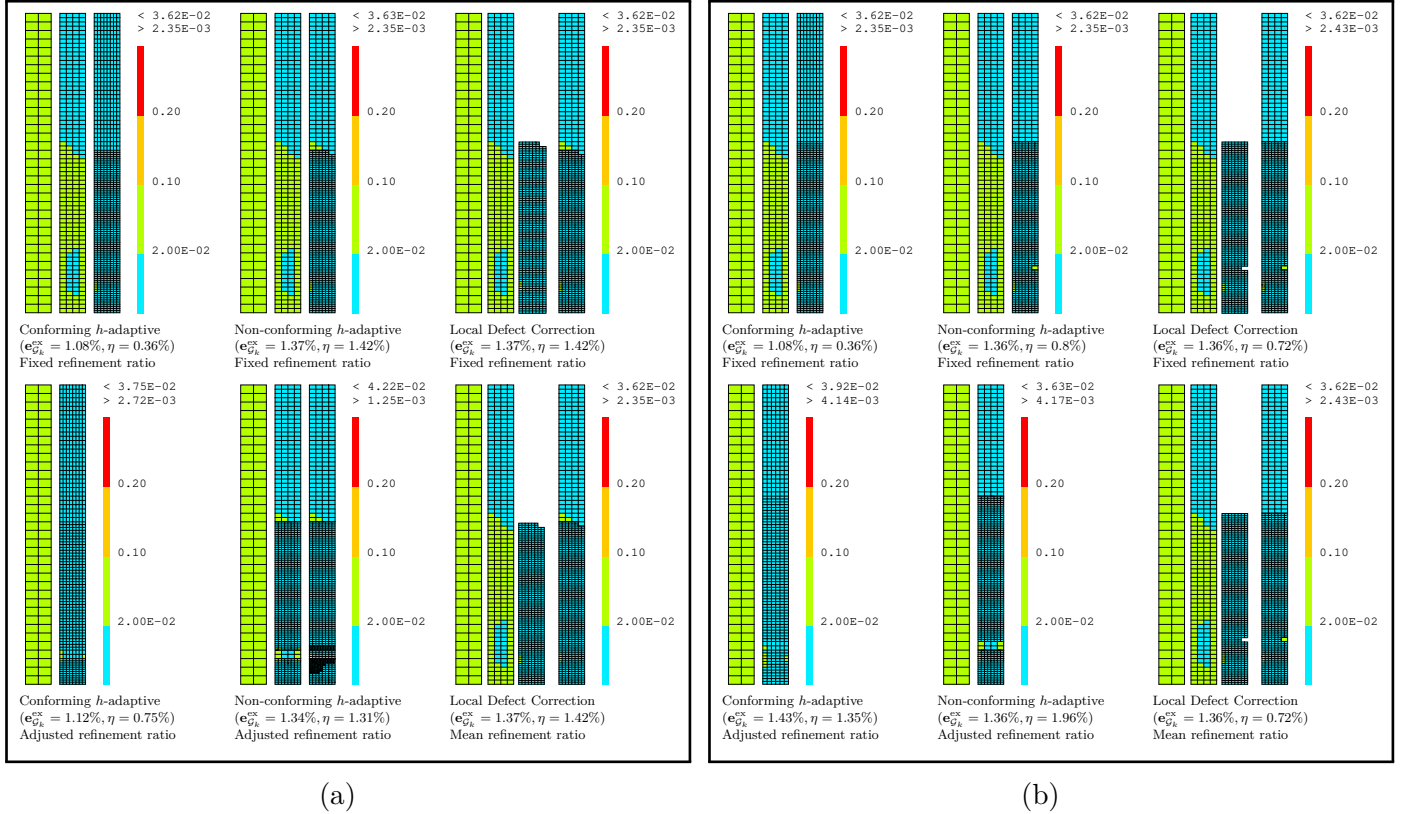


Figure 21: Mechanical Pellet-Cladding Interaction:  $2D$  test case – local error maps – tolerance on the global error  $\epsilon_{\Omega} = 2\%$  – local volume control parameter  $\delta = 3\%$ ; (a) OB optimality criterion and (b) LOC optimality criterion

case, only the LOC criterion guarantees the control of critical zone in the final mesh ( $\eta \leq \delta$ ) for all imposed  $\epsilon_{\Omega}$  values, independently of the refinement ratio and AMR method. In the presence of local singularity, the OB criterion does not allow us to satisfy the condition  $\eta \leq \delta$ , especially for lower error tolerances  $\epsilon_{\Omega}$ . Thus, the LOC optimality criterion is chosen for the remaining part of this comparative study. As expected, the conforming  $h$ -refinement leads to over-refined meshes accompanied with very limited  $\eta$  values at nearly every case. The non-conforming  $h$ -adaptive and LDC methods naturally lead to the same levels of final local accuracy.

**Refinement ratios.** Table 3 details the results for the global threshold  $\epsilon_{\Omega} = 0.5\%$  obtained for the LOC criterion with fixed and adjusted/mean refinement ratios. It can be seen that the application of adjusted refinement ratio results in more refined meshes, not optimal in terms of number of nodes. Moreover, the adjusted ratio refinement demands higher computational time, especially for the conforming  $h$ -adaptivity. It is due to a badly estimated adjusted refinement ratio  $\beta_{T_i}^{adj}$ , whose values, in this numerical example, are not sufficiently high to reach the user-prescribed accuracy in one refinement iteration. Since the refinement has to be repeated, the CPU time required for the hierarchical mesh adaptation becomes higher. Thus, the hierarchical conforming  $h$ -refinement technique becomes in this case really inefficient, in terms of the both, memory space and CPU time. Adjusted refinement ratio applied to the non-conforming  $h$ -adaptive technique leads to similar refined meshes than those obtained with the LDC method and mean refinement ratio. It has to be noted, that for this numerical example, application of the mean adjusted ratio

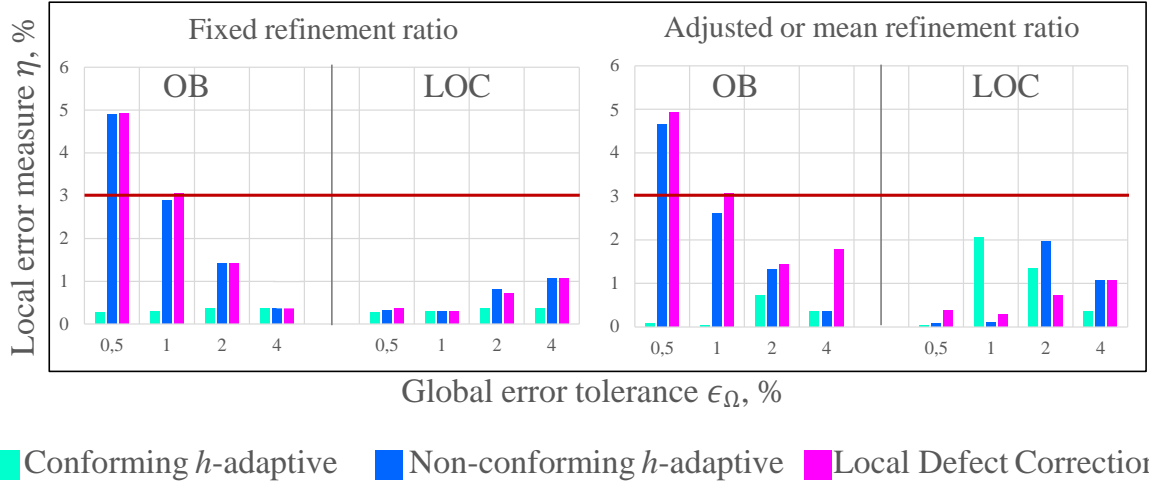


Figure 22: Mechanical Pellet-Cladding Interaction: 2D test case – local error measure  $\eta$  vs global error tolerance  $\epsilon_\Omega$

yields similar results in terms of number of nodes as well as CPU time, as the results obtained with the fixed ratio. Indeed, for the error tolerance  $\epsilon_\Omega = 0.5\%$ , the minimal mesh-step is equal to  $h_{T^2} = \frac{1}{16}h_{T^0}$  (with  $\bar{\beta}_{T^i=1,2} = \{8, 2\}$ ), which is the minimal mesh-step obtained for the fixed refinement (with  $k = 4$  refinement iterations). The only visible difference resides in the number of generated refinement steps.

Table 3: Mechanical Pellet-Cladding Interaction: 2D test case – LOC optimality criterion

$\epsilon_\Omega = 0.5\%, \delta = 3\%$	Fixed refinement ratio			Adjusted/mean refinement ratio		
	Conf. $h$ -adaptive	Non-conf. $h$ -adaptive	Local Defect Correction	Conf. $h$ -adaptive	Non-conf. $h$ -adaptive	Local Defect Correction
$k$	4	4	4	2	2	2
$e_{\mathcal{G}_k}^{\text{ex}}, \%$	0.27	0.35	0.35	0.19	0.34	0.36
$\eta, \%$	0.28	0.31	0.37	0.02	0.08	0.37
$N_{\mathcal{G}_k}$			12075			12075
$N_{\text{tot}}$	14586	11965	16237	23267	12349	14613
$N_{\text{max}}$			9728			9728
$N_{\text{hanging nodes}}$		109			141	
CPU total	22.5s	4.9s	3.8s	38.5s	4.6s	3.1s
CPU (SOLVE)	1.5s	1.9s	0.9s	2.8s	2.2s	0.6s
CPU (EST. + MARK)	0.9s	0.8s	0.6s	1.2s	0.7s	0.7s
CPU (REFINE)	20.1s	2.2s	2.3s	34.5s	1.7s	1.8s

It can be seen in Table 3 that, regardless the AMR method and optimality criterion, the global and local accuracies are respected. The LDC method confirms to be the most efficient approach in terms of the total runtime, regardless the refinement ratio applied. One gains approximately 30% of the total runtime compared to the hierarchical non-conforming  $h$ -refinement (non-conformity relations to be managed) and about 80% compared to the hierarchical conforming  $h$ -adaptive approach (global conforming meshes to be generated). Concerning the number of nodes, all AMR techniques lead to quite similar outcomes for the fixed refinement ratio strategy, but the adjusted/mean refinement ratio strategy is only interesting for non-conforming  $h$ -adaptive and LDC

methods.

**General remarks.** Based on both  $2D$  numerical examples, one can observe that application of the fixed refinement ratio  $\beta_{T^i}^{fix} = 2$  seems the most robust refinement strategy. It permits to progressively improve the mesh, finally generating the optimal one, and to precisely reach the prescribed level of accuracy. The direct use of adjusted refinement ratio has to be done with precaution, especially while dealing with problems having localized effects. However, its potential to achieve the imposed thresholds in fewer refinement iterations is a very attractive and promising feature. Nevertheless, the use of convergence rule (Eq.24) with two important hypotheses (i.e., independence of the constant  $C_{T^i}$  from the mesh-step and accordance of the local convergence order  $p$  to the one obtained globally) may be questioned. It seems natural that the constant  $C_{T^i}$  may depend on the mesh size as the constant of the global *a priori* estimator depends on the domain's measure [65], as already suggested in [41]. Moreover, it is likely that the local convergence order may change, for example near boundaries of the domain or singularities. Therefore, further exploration of *a priori* local error estimators may be greatly interesting in order to improve the formulation of the adjusted refinement ratio  $\beta_{T^i}^{adj}$ .

Finally, in presence of local singularities, only the LOC mesh optimality criterion enables to precisely control the local error.

### 3.3.2 Three-dimensional case

The three-dimensional case combines two phenomena representing the PCI, cf. Figure 20c. The interest of this problem is to obtain a realistic  $3D$  simulation of the cladding's response on the pellet modifications. For symmetry reasons, only  $1/32$  of the cladding in front of a pellet is represented ( $0 \leq \theta \leq \pi/8, 0 \leq z \leq L/2$ ).

This realistic industrial example reveals several crossed singularities of different characteristic length-scales. From the numerical point of view, finding an accurate solution to such three-dimensional problem is a challenging task since it may require solving high-dimensional (possibly ill-conditioned) systems. Thereby, this problem represents an attractive industrial benchmark to study the AMR methods considered here.

Since the reference solution is inaccessible in this example, the post-processing is based on estimated errors :  $\mathbf{e}_{\mathcal{G}_k}^{\text{est}} := \frac{\xi_{\mathcal{G}_k}}{\omega_{\mathcal{G}_k}}$  and  $\eta^{\text{est}} := \frac{\mu(\Omega_{\mathcal{M}_k^0})}{\mu(\Omega)}$ . These error metrics are the one used for the module **STOP**, cf. Section 2.4.

Here, the combination of the LOC optimality criterion and the fixed refinement ratio  $\beta_{T^i}^{fix} = 2$  is chosen due to its good performance in the previous numerical examples. Generation of the refined meshes  $\{\mathcal{G}_i\}_{i=1}^k$  starts with a coarse mesh  $\mathcal{G}_0$  with  $N_{\mathcal{G}_0} = 1596$ , which is not adapted to the pressure singularities.

As an example, Figure 23 depicts the sub-meshes obtained at each refinement iteration with the LDC approach and error tolerance  $\epsilon_{\Omega} = 1\%$ . We observe that the marked zones are progressively localized near the crossed singularities. The singularities become particularly well captured already at the refinement step  $i = 4$ , which validates the robustness of the ZZ error estimator and LOC criterion for this numerical example.

The displacement fields in the  $x$  direction (plotted on the deformed configuration) obtained with each studied AMR method for the global error tolerance  $\epsilon_{\Omega} = 4\%$  are depicted in Figure 24. It can be observed that the same regions are automatically detected and refined with each AMR method. One can also see that the global estimated error  $\mathbf{e}_{\mathcal{G}_k}^{\text{est}}$  is twice lower than the prescribed



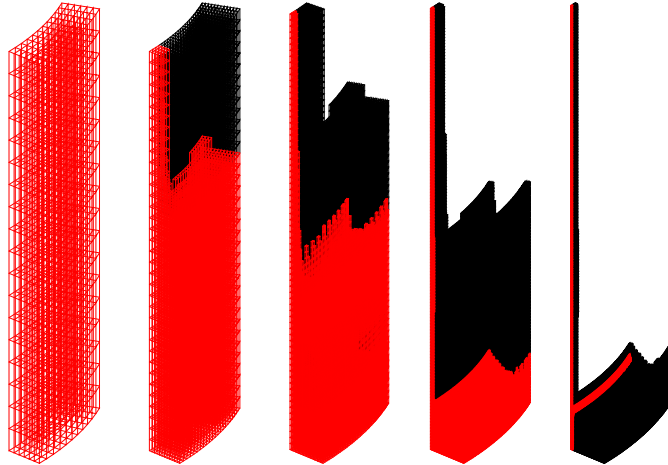


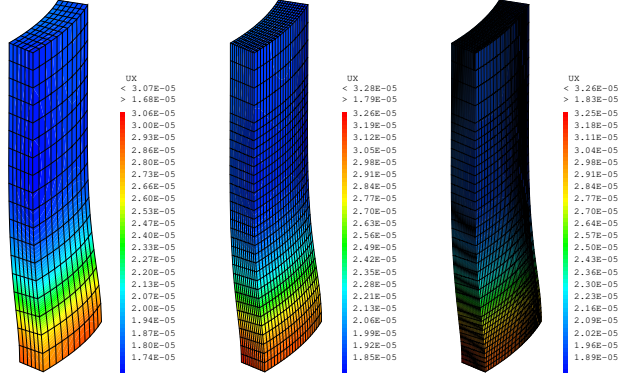
Figure 23: Mechanical Pellet-Cladding Interaction: 3D test case – LOC optimality criterion – tolerance on the global error  $\epsilon_\Omega = 1\%$  – local volume control parameter  $\delta = 3\%$ ; Example of the generated sub-meshes for the LDC method with sets of marked elements  $\mathcal{M}_i^0$  in red

tolerance of  $\epsilon_\Omega = 4\%$ . Moreover, the LOC criterion allows the estimated local error measure  $\eta^{\text{est}}$  to vanish (i.e.,  $\mathcal{M}_k = \emptyset$ ), which is a very satisfying result. Moreover, the generated refined sub-meshes for LDC approach have clearly much less elements compared to the global refined meshes generated with  $h$ -adaptive methods. Finally, it has to be noted that in this numerical example, the usage of  $h$ -adaptive techniques (especially of the conforming approach) can be very computationally expensive due to the perpendicularity of the singularities (see Table 4 for more details).

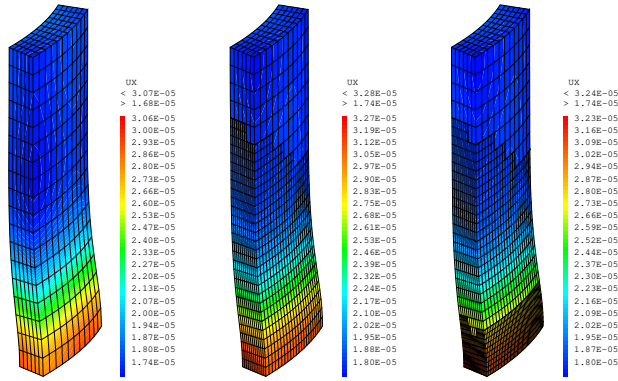
**Global and local accuracies.** Figure 25 shows the evolution of the estimated global error  $\mathbf{e}_{\mathcal{G}_k}^{\text{est}}$  and the local error measure  $\eta^{\text{est}}$  with respect to the number of refinement iterations for tolerance  $\epsilon_\Omega = 2\%$ . First, it can be seen that the number of performed refinement iterations is  $k = 3$ , regardless the applied AMR method. It is worth to note that one less refinement iteration could be carried out if the global accuracy was only of interest. For each case, the last iteration is performed to reach the local tolerance, i.e, the prescribed local control parameter  $\delta = 3\%$ .

Figure 26 provides the global estimated error  $\mathbf{e}_{\mathcal{G}_k}^{\text{est}}$  with respect to the number of nodes  $N_{\mathcal{G}_k}$  for all error tolerances ( $\epsilon_\Omega = 4, 2, 1\%$ ). The optimal rate of convergence  $\mathcal{O}(N_{\mathcal{G}_k}^{-1/3})$  is observed for all AMR methods. Note, that in this case, the uniform refinement would lead to a error in  $\mathcal{O}(N_{\mathcal{G}_k}^{-1/6})$  due to the stress singularity, see [15]. Hence, AMR techniques enable to recover the optimal rate of convergence with respect to the number of nodes. It has to be noted, that for  $N_{\mathcal{G}_k} < 1 \cdot 10^5$  an improved convergence could be obtained with the non-conforming  $h$ -adaptive and LDC methods.

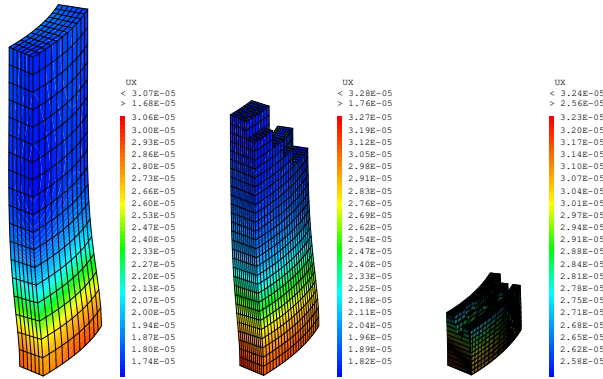
**AMR methods comparison.** Table 4 details the results obtained for error thresholds  $\epsilon_\Omega = 4, 2, 1\%$ , respectively, for all studied AMR methods. It can be seen, that essentially the same estimated global and local accuracies are reached by each AMR method. LDC technique enables to easily reach even the lowest error tolerances, for which the  $h$ -adaptive methods entail much higher computational cost. The cost associated with the conforming  $h$ -adaptive technique becomes dramatically high for low error tolerances  $\epsilon_\Omega$ . It has to be mentioned that for error tolerances  $\epsilon_\Omega = 2\%$  and  $1\%$ , the conforming  $h$ -adaptive technique requires to refine the whole computational



Conforming  $h$ -adaptive ( $e_{\mathcal{G}_k}^{\text{est}} = 1.44\%$ ,  $\eta^{\text{est}} = 0$ )



Non-conforming  $h$ -adaptive ( $e_{\mathcal{G}_k}^{\text{est}} = 1.93\%$ ,  $\eta^{\text{est}} = 0$ )



Local Defect Correction ( $e_{\mathcal{G}_k}^{\text{est}} = 1.95\%$ ,  $\eta^{\text{est}} = 0$ )

Figure 24: Mechanical Pellet-Cladding Interaction: 3D test case – Displacement field  $\mathbf{u}_x$  on the deformed configuration – LOC optimality criterion – tolerance on the global error  $\epsilon_{\Omega} = 4\%$  – local volume control parameter  $\delta = 3\%$

domain several times. For  $\epsilon_{\Omega} = 1\%$ , the last generated mesh is very computationally demanding and counts almost 5 millions of nodes, thus the associated problem solving has not been carried out. Furthermore, even though the computational cost of the conforming  $h$ -adaptive method is not

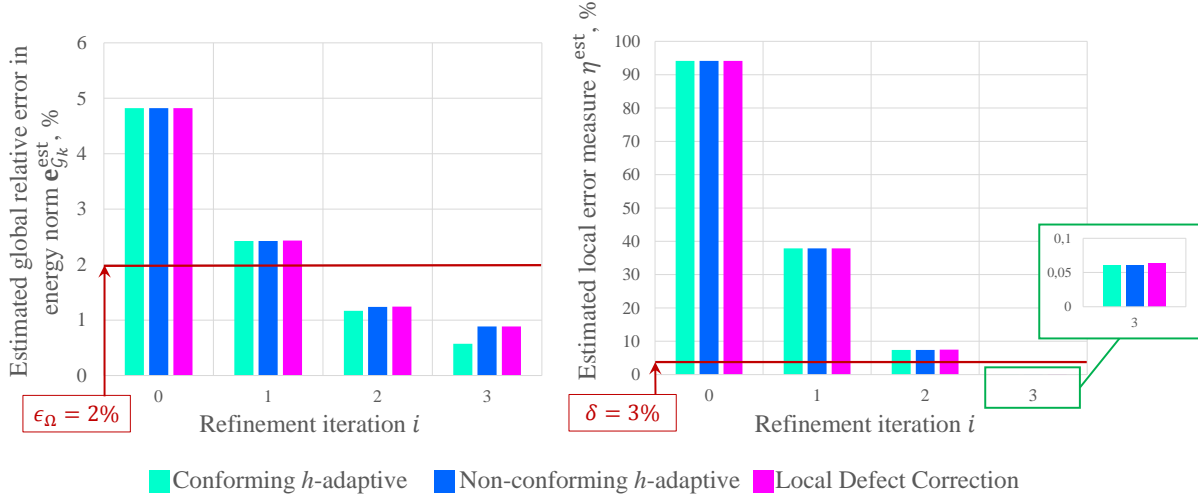


Figure 25: Mechanical Pellet-Cladding Interaction: 3D test case – Tolerance on the global error  $\epsilon_{\Omega} = 2\%$  – local volume control parameter  $\delta = 3\%$ ; Estimated global error  $e_{G_k}^{est}$  in energy norm vs refinement iteration; Estimated local error measure  $\eta^{est}$  vs refinement iteration

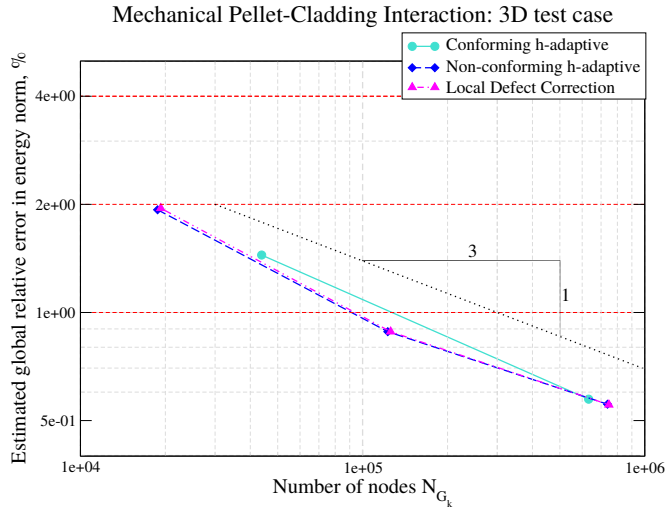


Figure 26: Mechanical Pellet-Cladding Interaction: 3D test case; Estimated global error  $e_{G_k}^{est}$  in energy norm vs number of nodes  $N_{G_k}$

that dramatic for the error tolerances  $\epsilon_{\Omega} > 1\%$ , it is still much higher than for the non-conforming and LDC methods. Concerning the non-conforming  $h$ -adaptive technique, the final refined meshes have similar number of nodes than the composite mesh of LDC approach. However, LDC method is much more efficient since it proceeds with smaller systems to be solved, see  $N_{max}$  for LDC that is almost 30% lower than  $N_{tot}$  of the non-conforming  $h$ -adaptive method (and four to ten times smaller than  $N_{tot}$  of the conforming  $h$ -adaptive approach). LDC method also enables to save the pre-processing time (e.g. for mesh enrichment) which can be quite important for complex realistic problems. Moreover, the computational cost of performing several iterations of the multigrid

iterative process is negligible compared to the cost of handling the non-conformity relations for the non-conforming  $h$ -adaptive strategy (cf. times of problems definition for non-conforming  $h$ -adaptive method).

Finally, in terms of total CPU time, LDC method is then computationally attractive: it allows us to save from 30% to 50% of the total runtime compared to the non-conforming  $h$ -adaptive method, and over 80% compared to the conforming  $h$ -adaptive technique. The LDC method computational savings are then really attractive despite the little extra storage consumption (storage of all the dofs of the generated sub-meshes (see  $N_{tot}$ ), stiffness matrices factorizations, etc.).

Table 4: Mechanical Pellet-Cladding Interaction: 3D test case – tolerances on the global error  $\epsilon_\Omega = 4, 2, 1\%$  – local volume control parameter  $\delta = 3\%$

	$\epsilon_\Omega = 4\%$			$\epsilon_\Omega = 2\%$			$\epsilon_\Omega = 1\%$		
	Conf. $h$ -adaptive	Non-conf. $h$ -adaptive	Local Defect Correction	Conf. $h$ -adaptive	Non-conf. $h$ -adaptive	Local Defect Correction	Conf. $h$ -adaptive	Non-conf. $h$ -adaptive	Local Defect Correction
$k$	2	2	2	3	3	3	4	4	4
$e_{G_k}^{est}, \%$	1.45	1.93	1.95	0.57	0.88	0.88	-	0.55	0.55
$\eta^{est}, \%$	0	0	0	0.06	0.06	0.06	-	0.08	0.08
$N_{G_k}$			19269			125727			746236
$N_{tot}$	43875	18766	21722	632345	122799	142991	4961841	737789	848487
$N_{max}$			12739			84943			495365
$N_{hanging\ nodes}$	-	1387	-	-	7059	-	-	33767	-
CPU total	72.8s	14.9s	10.5s	7330s	366s	202s	$\approx 500000s$	12211s	6476s
CPU (SOLVE + EST. + MARK)	41.9s	10.6s	6.5s	683s	236s	100s	-	2857s	850s
CPU (REFINE)	30.9s	4.3s	4s	6647s	130s	102s	-	9354s	5626s

It has to be underlined, that the hierarchical mesh refinement (based on elements subdivision) affects the total computational time for each AMR method. The time required for constructing refined meshes grows with the number of elements to be refined. Thus, it demands from 30% to 80% of the total runtime for the non-conforming  $h$ -adaptive and the LDC approaches, and it particularly badly affects the total runtime of the conforming  $h$ -adaptive method. It has to be emphasized that such an issue can be circumvented by using more efficient meshing techniques particularly using a parallelization of the sub-module **DIVIDE** applied on the marked zone procedure (only the final assembly must be done on the master processor).

To conclude, this nontrivial numerical example confirms the main conclusions deduced from more academic 2D problems presented in Sections 3.2 and 3.3.1. The adaptive multigrid LDC method is an attractive numerical tool to solve complex problems that proved its efficiency and robustness. Compared to the widely-used  $h$ -adaptive hierarchical techniques, LDC approach leads to promising results: it allows us to satisfy the imposed accuracy requirements with considerable computational savings. This 3D numerical test case clearly pointed out the limitations of the dicing-like hierarchical conforming  $h$ -refinement, with over-refined final meshes requiring expensive computations. The non-conforming  $h$ -refinement generally generates optimal meshes in terms of number of nodes, but its main difficulty resides in handling a large number of irregular nodes which, in addition, requires dedicated solvers or intrusive developments.

**General remarks.** It is worth underlying that the computations for this 3D test case were performed on a cluster with 240 Gb RAM. Note, that the computations on a workstation with 60 Gb RAM were possible only for LDC method. Indeed, systems solving on meshes implying more than 700 thousand nodes isn't reachable for this RAM. This implies that the solution for the error threshold  $\epsilon_\Omega = 1\%$  could not be obtained for the  $h$ -adaptive methods.

## 4 Conclusions

In this study, the widely used  $h$ -refinement methods have been compared to the adaptive multigrid Local Defect Correction (LDC) method in the context of linear elasticity for quadrilateral/hexahedral finite elements. Academic and more realistic engineering numerical examples in  $2D$  and  $3D$  revealing local stress concentrations or local stress singularities have been used to perform this comparison. The main conclusions can be summarized as follows:

- The LDC multigrid method permits to ensure the local character of the refinement (as non-conforming  $h$ -adaptive approaches) while circumventing the difficulties related to non-conforming meshes. It can be implemented in a non-intrusive way (only the pre- and post-processing operations to be carried out) in any existing software. Even if the LDC method needs little extra storage memory compared to non-conforming hierarchical  $h$ -adaptive approaches, the random-access memory requirements by problem resolution is really lower since problems sizes are limited. Hence, from the numerical point of view, the LDC method is shown to be attractive in terms of overall runtimes compared to both the hierarchical conforming and non-conforming  $h$ -adaptive approaches.
- The main bottleneck of the hierarchical conforming  $h$ -adaptive strategy resides in the mesh refinement: the conforming all-quadrangular or all-hexahedral mesh refinement is still an undergoing research field. Generally, local conforming mesh refinement is non trivial (based on patterns) while sheet-based refinement is far from being local, which naturally results in over-refined final meshes, affecting the computational time. In its turn, the main complexity of the hierarchical non-conforming  $h$ -adaptive technique resides in handling continuity constraints. The resulting system of equations to be solved is often computationally demanding.
- This comparative study intrinsically leads to a comparison of refinement strategies related to mesh optimality criterion and refinement ratio. For all numerical examples considered in this study, the so-called LOC mesh optimality criterion seems to be the most adapted to precisely control the local error. The OB criterion is not as much locally precise as the LOC criterion, especially in presence of local singularities, but is sufficient if only global accuracy is of interest. Finally, setting an *a priori* chosen fixed mesh-step refinement ratio (of value 2) on the detected elements has been proven to be the most efficient strategy. However, it appeared interesting to further explore the determination of an optimal adjusted refinement ratio enabling to reduce the number of refinement iterations.

The really promising results obtained with the locally adaptive multigrid LDC method open the way to apply this approach on nonlinear solids mechanics problems. Moreover, this technique offers others potentialities which would be interesting to test, especially the possibility to change the model (multimodel approach) between levels of refinement.

## 5 Acknowledgements

This work was developed within the framework of the MISTRAL joint research laboratory between Aix-Marseille University, CNRS, Centrale Marseille and CEA (Commissariat à l'Énergie Atomique et aux Énergies Alternatives). The authors are grateful to the PLEIADES project, financially

supported by the CEA, EDF (Électricité de France) and FRAMATOME, that funded this research work.

## References

- [1] L. Demkowicz, Ph. Devloo, and J.T. Oden. On an h-type mesh-refinement strategy based on minimization of interpolation errors. *Computer Methods in Applied Mechanics and Engineering*, 53(1):67–89, 1985.
- [2] I. Babuška and B. Szabo. On the rates of convergence of the Finite Element method. *International Journal for Numerical Methods in Engineering*, 18(3):323–341, 1982.
- [3] S. Ghosh and S.K. Manna. R-adapted Arbitrary Lagrangian-Eulerian Finite Element method in metal-forming simulation. *Journal of Materials Engineering and Performance*, 2(2):271–282, 1993.
- [4] J. Fish. The s-version of the Finite Element method. *Computers and Structures*, 43(3):539–547, 1992.
- [5] I. Babuška and B.Q. Guo. The h, p and h-p version of the Finite Element method; basis theory and applications. *Advances in Engineering Software*, 15(3):159–174, 1992.
- [6] T. Belytschko and M. Tabbara. H-adaptive Finite Element methods for dynamic problems, with emphasis on localization. *International Journal for Numerical Methods in Engineering*, 36(24):4245–4265, 1993.
- [7] P. Diez and A. Huerta. A unified approach to remeshing strategies for Finite Element h-adaptivity. *Computer Methods in Applied Mechanics and Engineering*, 176(1):215–229, 1999.
- [8] I. Babuška, A. Miller, and M. Vogelius. Adaptive methods and error estimation for elliptic problems of structural mechanics. *Adaptive Computational Methods for Partial Differential Equations*, page 20, 1983.
- [9] D. Davydov, J-P. Pelteret, D. Arndt, M. Kronbichler, and P. Steinmann. A matrix-free approach for finite-strain hyperelastic problems using geometric multigrid. *International Journal for Numerical Methods in Engineering*, n/a(n/a).
- [10] A. Brandt. Rigorous quantitative analysis of multigrid I: Constant coefficients Two-Level cycle with l2-norm. *SIAM Journal on Numerical Analysis*, 31(6):1695–1730, 1994.
- [11] D. Bai and A. Brandt. Local mesh refinement multilevel techniques. *SIAM Journal on Scientific and Statistical Computing*, 8(2):109–134, mar 1987.
- [12] M.J. Berger and J. Olinger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484–512, 1984.

- [13] K. Khadra, Ph. Angot, and J. P. Caltagirone. A comparison of locally adaptive multigrid methods : L.D.C., F.A.C, and F.I.C. In N. D. Melson, S. F. McCormick, and T. A. Manteuffel, editors, *NASA Conference Publication 3224, 6th Copper Mountain Conference on Multigrid Methods*, volume 1, pages 275–292, 1993.
- [14] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.
- [15] L. Barbié, I. Ramière, and F. Lebon. Strategies around the Local Defect Correction multi-level refinement method for three-dimensional linear elastic problems. *Computers and Structures*, 130:73–90, 2014.
- [16] L. Barbié, I. Ramière, and F. Lebon. An automatic multilevel refinement technique based on nested local meshes for nonlinear mechanics. *Computers and Structures*, 147:14–25, 2015.
- [17] H. Liu, I. Ramière, and F. Lebon. On the coupling of local multilevel mesh refinement and ZZ methods for unilateral frictional contact problems in elastostatics. *Computer Methods in Applied Mechanics and Engineering*, 323:1–26, 2017.
- [18] W. Hackbusch. Local Defect Correction method and Domain Decomposition techniques. *Defect Correction Methods: Theory and Applications*, pages 89–113. Springer Vienna, 1984.
- [19] P.J.J. Ferket and A.A. Reusken. Further analysis of the Local Defect Correction method. *Computing*, 56:117–139, 1996.
- [20] G. Nicolas, T. Fouquet, Geniaut S., and Cuvilliezam S. Improved adaptive mesh refinement for conformal hexahedral meshes. *Advances in Engineering Software*, 102:14–28, 2016.
- [21] I. Ramière, Ph. Angot, and M. Belliard. A fictitious domain approach with spread interface for elliptic problems with general boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 196(4):766–781, 2007.
- [22] A. Düster, J. Parvizian, Z. Yang, and E. Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197(45):3768–3782, 2008.
- [23] T. Belytschko, Ch. Parimi, N. Moës, N. Sukumar, and Sh. Usui. Structured extended finite element methods for solids defined by implicit surfaces. *International Journal for Numerical Methods in Engineering*, 56(4):609–635, 2003.
- [24] E. Wang, Th. Nelson, and R. Rauch. Back to elements - tetrahedra vs. hexahedra. Proceedings of the 2004 International ANSYS Conference, 2004.
- [25] R. Biswas and R.C. Strawn. Tetrahedral and hexahedral mesh adaptation for cfd problems. *Applied Numerical Mathematics*, 26(1):135–151, 1998.
- [26] S.H. Huo, Y.S. Li, S.Y. Duan, X. Han, and G.R. Liu. Novel quadtree algorithm for adaptive analysis based on cell-based smoothed finite element method. *Engineering Analysis with Boundary Elements*, 106:541–554, 2019.

- [27] B. Michel, C. Nonon, J. Sercombe, F. Michel, and V. Marelle. Simulation of pellet-cladding interaction with the pleiades fuel performance software environment. *Nuclear Technology*, 182:124–137, 05 2013.
- [28] E. Biotteau, A. Gravouil, T. Lubrecht, and A. Combescure. Three dimensional automatic refinement method for transient small strain elastoplastic finite element computations. *Computational Mechanics*, 49:123–136, 2012.
- [29] J. Z. Zhu, E. Hinton, and O. C. Zienkiewicz. Mesh enrichment against mesh regeneration using quadrilateral elements. *Communications in Numerical Methods in Engineering*, 9(7):547–554, 1993.
- [30] J. Ródenas, J. Albelda, M. Tur, and F. Fuenmayor. A hierarchical  $h$ -adaptivity methodology based on element subdivision. *Revista UIS Ingenierías*, 16:263–280, 2017.
- [31] R. Schneiders. Refining quadrilateral and hexahedral element meshes. *Fifth International Meshing Roundtable*, 1, 1998.
- [32] K. Tchon, J. Dompierre, and R. Camarero. Conformal refinement of all-quadrilateral and all-hexahedral meshes according to an anisotropic metric. In *in 11th International Meshing Roundtable*, pages 231–242, 2002.
- [33] F. Ledoux and J. F. Shepherd. Topological modifications of hexahedral meshes via sheet operations: a theoretical study. *Engineering with Computers*, 26:433–447, 2009.
- [34] D. J. Melander, S. E. Benzley, and T. J. Tautges. Generation of multi-million element meshes for solid model-based geometries: The dicer algorithm. 1997.
- [35] K. Merkley, C. Ernst, J. F. Shepherd, and M. J. Borden. Methods and applications of generalized sheet insertion for hexahedral meshing. In Michael L. Brewer and David Marcum, editors, *Proceedings of the 16th International Meshing Roundtable*, pages 233–250, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [36] M. L. Staten, J. F. Shepherd, and K. Shimada. Mesh matching – creating conforming interfaces between hexahedral meshes. In Rao V. Garimella, editor, *Proceedings of the 17th International Meshing Roundtable*, pages 467–484, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [37] W. F. Mitchell and M. A. McClain. A comparison of hp-adaptive strategies for Elliptic Partial Differential equations. *ACM Transactions on Mathematical Software*, 41(1):1–39, 2014.
- [38] P. Hennig, M. Kästner, Ph. Morgenstern, and D. Peterseim. Adaptive mesh refinement strategies in isogeometric analysis — a computational comparison. *Computer Methods in Applied Mechanics and Engineering*, 316:424–448, 2017. Special Issue on Isogeometric Analysis: Progress and Challenges.
- [39] W. Ehlers, M. Ammann, and S. Diebels. H-adaptive FE methods applied to single- and multiphase problems. *International Journal for Numerical Methods in Engineering*, 54(2):219–239, 2002.



- [40] O.C. Zienkiewicz and J.Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering*, 24(2):337–357, 1987.
- [41] E. Oñate and G. Bugeda. Mesh optimality criteria for adaptive Finite Element computations. *Engineering Computations*, pages 307–321, 1993.
- [42] I. Ramière, H. Liu, and F. Lebon. Original geometrical stopping criteria associated to multi-level adaptive mesh refinement for problems with local singularities. *Computational Mechanics*, 2019.
- [43] I. Babuška, T. Strouboulis, S.K. Gangaraj, and C.S. Upadhyay. Pollution error in the h-version of the finite element method and the local quality of the recovered derivatives. *Computer Methods in Applied Mechanics and Engineering*, 140(1):1–37, 1997.
- [44] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element method*. Springer-Verlag, Berlin, Heidelberg, 1991.
- [45] L. Demkowicz, J.T. Oden, W. Rachowicz, and O. Hardy. Toward a universal h-p adaptive finite element strategy, part 1. constrained approximation and data structure. *Computer Methods in Applied Mechanics and Engineering*, 77(1):79–112, 1989.
- [46] I. Babuška and M. Zlamal. Nonconforming elements in the Finite Element method with penalty. *SIAM Journal on Numerical Analysis*, 10(5):863–875, 1973.
- [47] C. Bernardi, Y. Maday, and A. T. Patera. *Domain Decomposition by the Mortar Element method*, pages 269–286. Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters. Springer Netherlands, Dordrecht, 1993.
- [48] J. Červený, V. Dobrev, and T. Kolev. Nonconforming mesh refinement for high-order finite elements. *SIAM Journal on Scientific Computing*, 41(4):C367–C392, 2019.
- [49] P. Cavin, A. Gravouil, A. A. Lubrecht, and A. Combescure. Efficient FEM calculation with predefined precision through automatic grid refinement. *Finite Elements in Analysis and Design*, 41(11-12):1043–1055, 2005.
- [50] M. Graziadei, R. M. M. Mattheij, and J. H. M. ten Thije Boonkamp. Local defect correction with slanting grids. *Numerical Methods for Partial Differential Equations*, 20(1):1–17, 2004.
- [51] D. Koliesnikova, I. Ramière, and F. Lebon. Analytical comparison of two multiscale coupling methods for nonlinear solid mechanics. *ASME, Journal of Applied Mechanics*, 2020. accepted.
- [52] T. Strouboulis and K.A. Haque. Recent experiences with error estimation and adaptivity. Part II: Error estimation for h-adaptive approximations on grids of triangles and quadrilaterals. *Computer Methods in Applied Mechanics and Engineering*, 100(3):359–430, 1992.
- [53] T. Grätsch and Kl.-J. Bathe. Review: A posteriori error estimation techniques in practical finite element analysis. *Computers and Structures*, 83(4-5):235–265, 2005.

- [54] M. Özakça. Comparison of error estimation methods and adaptivity for plane stress/strain problems. *Structural Engineering and Mechanics*, 15:579–608, 2003.
- [55] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7):1331–1364, 1992.
- [56] D. Fournier, R. Herbin, and R. Le Tellier. Discontinuous galerkin discretization and  $h$ -refinement for the resolution of the neutron transport equation. *SIAM Journal on Scientific Computing*, 35(2):936–956, 2013.
- [57] D. A. Di Pietro, M. Vohralik, and S. Yousef. An a posteriori-based, fully adaptive algorithm with adaptive stopping criteria and mesh refinement for thermal multiphase compositional flows in porous media. *Computers and Mathematics with Applications*, 68(12, Part B):2331–2347, 2014. *Advances in Computational Partial Differential Equations*.
- [58] W. Dörfler. A convergent adaptive algorithm for poisson’s equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124, 1996.
- [59] P. Morin, K. G. Siebert, and A. Veerer. A basic convergence result for conforming adaptive finite elements. *Mathematical Models and Methods in Applied Sciences*, 18(05):707–737, 2008.
- [60] G. Bugeda. A comparison between new adaptive remeshing strategies based on point wise stress error estimation and energy norm error estimation. *Communications in Numerical Methods in Engineering*, 18(7):469–482, 2002.
- [61] P. Ladevèze, J.P. Pelle, and P. Rougeot. Error estimation and mesh optimization for classical Finite Elements. *Engineering Computations*, 8:69–80, 1991.
- [62] G. Bugeda and J. Oliver. A general methodology for structural shape optimization problems using automatic adaptive remeshing. *International Journal for Numerical Methods in Engineering*, 36(18):3161–3185, 1993.
- [63] D. Fournier, R. Herbin, and R. Le Tellier. Discontinuous Galerkin Discretization and  $h - p$ -refinement for the resolution of the neutron transport equation. *SIAM Journal on Scientific Computing*, 35:A936–A956, 2013.
- [64] V. Faucher, F. Casadei, G. Valsamosand, and M. Larcher. High resolution adaptive framework for fast transient fluid-structure interaction with interfaces and structural failure – application to failing tanks under impact. *International Journal of Impact Engineering*, 127:62–85, 2018.
- [65] Ph. G. Ciarlet. *Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.
- [66] A.C. Woodbury, M. L. Staten J. F. Shepherd, and S. E. Benzley. Localized coarsening of conforming all-hexahedral meshes. *Engineering with Computers*, 27(1):95–104, 2011.
- [67] R. Bank and A. Sherman. Some refinement algorithms and data structures for regular local mesh refinement. *Applications of Mathematics and Computing to the Physical Sciences*, 1999.

- [68] R. H. Nochetto and A. Veerer. *Primer of Adaptive Finite Element Methods*. Springer Berlin Heidelberg, 2012.
- [69] CEA. Cast3m. <http://www.cast3m.cea.fr>, 2019.