

## Research Article

# Efficient Finite Element Methodology Based on Cartesian Grids: Application to Structural Shape Optimization

**E. Nadal, J. J. Ródenas, J. Albelda, M. Tur, J. E. Tarancón, and F. J. Fuenmayor**

*Centro de Investigación de Tecnología de Vehículos (CITV), Universidad Politécnica de Valencia, 46022 Valencia, Spain*

Correspondence should be addressed to J. J. Ródenas; [jjrodena@mcm.upv.es](mailto:jjrodena@mcm.upv.es)

Received 25 January 2013; Accepted 20 February 2013

Academic Editor: Juan J. Nieto

Copyright © 2013 E. Nadal et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work presents an analysis methodology based on the use of the Finite Element Method (FEM) nowadays considered one of the main numerical tools for solving Boundary Value Problems (BVPs). The proposed methodology, so-called cg-FEM (Cartesian grid FEM), has been implemented for fast and accurate numerical analysis of 2D linear elasticity problems. The traditional FEM uses geometry-conforming meshes; however, in cg-FEM the analysis mesh is not conformal to the geometry. This allows for defining very efficient mesh generation techniques and using a robust integration procedure, to accurately integrate the domain's geometry. The hierarchical data structure used in cg-FEM together with the Cartesian meshes allow for trivial data sharing between similar entities. The cg-FEM methodology uses advanced recovery techniques to obtain an improved solution of the displacement and stress fields (for which a discretization error estimator in energy norm is available) that will be the output of the analysis. All this results in a substantial increase in accuracy and computational efficiency with respect to the standard FEM. cg-FEM has been applied in structural shape optimization showing robustness and computational efficiency in comparison with FEM solutions obtained with a commercial code, despite the fact that cg-FEM has been fully implemented in MATLAB.

## 1. Introduction

Researchers have devoted many efforts to solve Boundary Value Problems (BVPs) that in one way or another are relevant to different disciplines. The advent of computers allowed the researchers to develop methods to obtain numerical solutions of this type of problems. One of the most common methods used to solve BVPs is the Finite Element Method (FEM) which, nowadays, is broadly used for industrial applications. In the FEM the BVP is discretized subdividing (meshing) the problem domain  $\Omega$  into small subdomains (elements) of simple geometry. The original FEM strongly relies on a mesh that must conform the geometry of the domain to be analyzed and must be adapted to the solution of the problem, for example, using  $h$ -adapted meshes where elements of smaller size are used in regions where the solution is less smooth. According to [1], a study at Sandia National Laboratories revealed that the process of creating an analysis-suitable geometry and the meshing of that geometry require about 80% of overall analysis time, whereas only 20% of overall time is devoted to the analysis itself. One way to

decrease this 80% of overall time prior to the analysis is to make the geometry of the mesh used for the analysis independent of the domain to be analyzed. This approach can considerably reduce the time devoted to prepare an analysis-suitable model and mesh the domain and is especially useful in applications that would require continuous remeshings during the analysis: structural shape optimization problems, wear modelling, and so forth.

There have been several variations of the original FEM that followed this path to improve the efficiency of the FEM. Two of these improvements of the original FEM are the Extended Finite Element Method (XFEM) [2, 3] developed by Belytschko and his group at Northwestern University (USA) and the Generalized Finite Element Method (GFEM) [4, 5] developed by Melenk and Babuška at the University of Texas at Austin (USA). XFEM is mainly devoted to the analysis of inclusions or cracks. It uses the Partition of Unity Method (PUM) [6] to introduce enrichment functions to represent the displacement discontinuity between the crack faces, the singular fields around the crack tip, and the geometrical description of the crack by means of the Level Set Method

(LSM) [7]. The method improves the accuracy of the results and is particularly interesting in crack growing problems as the mesh can remain unchanged when the crack evolves. GFEM [4] follows a similar approach also based in the PUM to include enrichment functions to describe the known behavior of the solution. In GFEM the mesh used for the analysis can be independent of the geometry. For example, a Cartesian grid is used in the GFEM III implementation described in [4]. Both XFEM and GFEM require the use of an integration mesh, purely for integration purposes, in the elements cut by the boundary to take into account the part of the element actually lying within the domain. Another approach to represent the geometry for nonconforming meshes is to use the LSM for boundary representation [8]. Other authors also combine the LSM for boundary representation with the XFEM to represent a solution with gradient discontinuities into an element containing more than one material [9, 10].

Another variation of the FEM, developed to improve its performance, consists of using an auxiliary domain  $\Omega_E$  which embeds the problem domain  $\Omega$ . In this approach, the discretized domain is  $\Omega_E$  instead of the problem domain. Generally  $\Omega_E$  is a domain with a simple geometry that can be easily meshed. This technique is therefore closely related to the GFEM. The analysis methodology presented in this paper is based on this idea. In our implementation for 2D problems  $\Omega_E$  is a square whose discretization into squared quadrilateral elements of uniform size is trivial. These techniques have been used both in the Finite Volume Method (FVM) and in the FEM. According to [11] the literature of these techniques starts in the 60s when VK Saul'ev published, in Russian, the paper. *Solution of certain boundary-value problems on high-speed computers by the fictitious-domain method* (Sibirsk. Mat. Z. 1963.4: 912-925). Subsequently it has been applied in various fields, such as acoustics [12–14], fluid dynamics and fluid-structure interaction [15, 16], biomedical problems [17, 18], convection-diffusion [19], and optimization [20–23]. The present work will only focus on the FEM framework.

These techniques have several names in the literature, such as *Fictitious Domain* [11, 12, 19, 20, 24], *Implicit Meshing* [25], *Immersed FEM* [26], *Immersed Boundary Method* [15, 27], and *Fixed grid FEM* [28, 29]. They have been described in [30] under the name *Finite Elements in Ambient Space*.

Since the mesh is not conforming with the geometry, these methods require the information of the problem domain to be available during the evaluation of element integrals. The accuracy of the results provided by these techniques depends on the accuracy of the integration process. Hence, the methodology proposed in this paper includes an efficient integration procedure which would be even able to consider the actual boundary providing the exact element integrals (up to the accuracy of the numerical integration and round-off errors).

One major difference of these methodologies with the standard FEM is the consideration of the Dirichlet boundary conditions as in the proposed methodology; in the general case, there are no nodes lying on these boundaries. A procedure based on the use of the Lagrange multipliers technique has been used to apply these boundary conditions.

All these previous approaches are mainly interested in decoupling the geometry representation from the FE mesh where the solution is interpolated. Generally, in these last techniques, the computational cost is concentrated along the elements falling on the boundary. The same occurs in the Boundary Element Method (BEM) where only the external boundary is considered for the analysis. In this framework, recent works of Simpson et al. [31] and Scott et al. [32] where the Isogeometric Analysis IGA [33] with NURBS and T-Splines, respectively, have been adapted to the BEM. Thus, the geometry and the BEM mesh are strongly coupled in this approach.

In this paper we will consider the FE solution of linear elasticity problems, where the Zienkiewicz and Zhu [34] (ZZ) error estimator in energy norm is commonly used to quantify the accuracy of the numerical solution. The information provided by the ZZ error estimator at element level can be used to improve the model by means of  $h$ -adaptive procedures. The ZZ error estimator is widely used due to its robustness, accuracy, and easy implementation but also because it requires the evaluation of a recovered stress solution  $\sigma^*$ , more accurate than the raw FE solution  $\sigma^h$ , that can be used instead of  $\sigma^h$ . From all recovery techniques we can highlight those based on the Superconvergent Patch Recovery (SPR) technique [35] also developed by Zienkiewicz and Zhu. The publication of the original SPR technique was followed by several works aimed at improving its quality [36–38]. Ródenas and coworkers also proposed several improvements [39–42] of the original SPR in order to yield a highly accurate recovered field which locally fulfills the equilibrium and compatibility equations. These techniques allowed us to obtain the first procedures to get practical upper error bounds for FEM and XFEM based on recovery techniques [40, 42] instead of using the traditional residual-based error estimators [43].

Reference [25] indicates the following: “*Unfortunately, for an implicit mesh it would be very difficult to implement such a superconvergent recovery scheme of the stress field for elements that intersect the boundary.*” However in the XFEM framework, where the mesh is independent of the crack, efficient recovery techniques have been already proposed based on the Moving Least Squares (MLS) technique [44–47] and some on the SPR technique [42, 48], which introduce worthy improvements to the solution specially along the boundaries, even in elements trimmed by the crack. These SPR-based techniques have been adapted in this work to the context of Cartesian grids.

According to some authors [11, 25, 27, 28], the main drawback of the use of the techniques under the large umbrella of *finite elements in ambient space* is the low accuracy along the boundaries since they are not explicitly represented. In the proposed methodology, the recovery techniques developed by Ródenas and coworkers [39, 40, 42, 48] have been specially adapted both to be used in the Zienkiewicz and Zhu error estimator [34] that will guide the  $h$ -adaptive refinement process to improve the quality of the solution and to neutralize the possible lack of accuracy along the boundaries in the cg-FEM framework providing an enhanced solution (for which an error estimator is available) that will be used instead of the FE solution.

In this contribution we have applied cg-FEM to a structural shape optimization analysis. This type of processes will benefit from the computational efficiency and accuracy of cg-FEM but also from the data structure that will allow for sharing information between different geometries analyzed during the process, further improving the behavior of the optimization process.

The paper is organized as follows. The problem to be solved is defined in Section 2. The main characteristics of the proposed cg-FEM methodology are presented in Sections 3 and 4, respectively, devoted to the use of Cartesian grids independent of the geometry for the FE analysis and to the  $h$ -adaptive procedures specially adapted to the use of Cartesian grids. Numerical examples will be used in Section 5 to evaluate the performance of the cg-FEM methodology.

## 2. Problem Statement

Let us consider the 2D linear elastic problem. Let  $\mathbf{u}$  be the displacement field defined in  $\Omega \subset \mathbb{R}^2$ , the solution of the boundary value problem:

$$\begin{aligned} -\nabla \cdot \sigma(\mathbf{u}) &= \mathbf{b} & \text{in } \Omega, \\ \sigma(\mathbf{u}) \mathbf{n} &= \mathbf{t} & \text{in } \Gamma_N, \\ \mathbf{u} &= \bar{\mathbf{u}} & \text{in } \Gamma_D, \end{aligned} \quad (1)$$

where  $\Gamma_N$  and  $\Gamma_D$ , with  $\partial\Omega = \bar{\Gamma}_N \cup \bar{\Gamma}_D$  and  $\Gamma_N \cap \Gamma_D = \emptyset$ , are two disjoint boundary parts where the Neumann and Dirichlet boundary conditions are applied.  $\mathbf{b}$  are the body loads,  $\mathbf{t}$  are the tractions along  $\Gamma_N$ , and  $\bar{\mathbf{u}}$  are the prescribed displacements along  $\Gamma_D$ . The weak formulation of this problem reads as follow: find  $\mathbf{u} \in (V + \bar{\mathbf{u}})$  such that

$$a(\mathbf{u}, \mathbf{v}) = l(\mathbf{v}) \quad \forall \mathbf{v} \in V, \quad (2)$$

where  $V$  is the standard test space for the linear elasticity problem and

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &:= \int_{\Omega} \sigma(\mathbf{u})^T \epsilon(\mathbf{v}) \, d\Omega = \int_{\Omega} \sigma(\mathbf{u})^T \mathbf{D}^{-1} \sigma(\mathbf{v}) \, d\Omega, \\ l(\mathbf{v}) &:= \int_{\Omega} \mathbf{b}^T \mathbf{v} \, d\Omega + \int_{\Gamma_N} \mathbf{t}^T \mathbf{v} \, d\Gamma, \end{aligned} \quad (3)$$

where  $\mathbf{D}$  is the Hook's tensor and  $\sigma$  and  $\epsilon$  are the stress and strain operators, respectively.

This problem will be solved numerically by means of the cg-FEM methodology described in this paper, which includes  $h$ -adaptive refinement capabilities. The use of Cartesian grids and a hierarchical data structure [49] improves the manipulation information and reduces the computational cost of the analysis.

## 3. On the Use of Cartesian Grids for Efficient FE Structural Analysis

In the standard FEM the mesh must conform to the boundary of the domain. This requirement, together with the need to

adequately refine the mesh in order to accurately represent the behavior of interest whilst maintaining the geometry of the elements as undistorted as possible, is the origin of the high cost (both in terms of computing time and analyst's man-hours) of the process required to generate an adequate FE model. In cg-FEM the mesh does not need to conform the geometry. In other words, we will deal with two different domains,  $\Omega_E$  which is the domain to be meshed, a square in our case, and the domain of the problem to be solved  $\Omega$ . The only requirement between these domains is that  $\Omega \subset \Omega_E$ .

**3.1. Generation of the Analysis Mesh.** cg-FEM is based on the use of a sequence of uniformly refined Cartesian meshes where hierarchical relations between the different mesh levels have been defined. This sequence of meshes used to discretize  $\Omega_E$  is called the *Cartesian grid pile* (see Figure 1(a)) which embeds all the problem domain  $\Omega$  and it is formed by bilinear (Q4) or biquadratic (Q8) squared elements of uniform size. A hierarchical data structure for  $h$ -adaptive FE analysis based on element splitting was presented in [49]. This data structure took into account the hierarchical relations between the elements of different levels, obtained during the element splitting process, to speed up FE computations. The data structure has been adapted to the particular case of a sequence of meshes given by the *Cartesian grid pile* where all elements are geometrically similar to the element used in the coarsest level (level 0) of the *Cartesian grid pile*, called the *reference element*. One of the main benefits of the data structure is that, as described later in Section 3.3.1, in the linear elastic case all elements of the *Cartesian grid pile* will have the same stiffness matrix that will be evaluated only for the *reference element* and shared with the rest of elements in the pile, making the evaluation of element stiffness matrices trivial. This and other hierarchical relations considered in the data structure allow for a simplification of the mesh refinement process and the preevaluation of most of the information used by the FE code, remarkably influencing the efficiency of the code.

The code uses functions that directly provide the nodal coordinates, mesh topology, hierarchical relations, neighborhood relations, and so forth, in an efficient manner, when required. Therefore, there is no need to store this information in memory, considerably reducing memory usage.

The first step of the analysis consists of creating the *analysis mesh* used to obtain the FE solution of the problem. This mesh is built taking a set of nonoverlapped elements of different sizes (see Figure 1(b)) taken from the different levels of the *Cartesian grid pile*. A maximum difference of 1 refinement level is allowed between adjacent elements in the *analysis mesh*. Multipoint constraints (MPCs) [50, 51] are used to enforce  $C^0$  continuity between adjacent elements of different levels.

**3.2. Geometry-Mesh Intersection: Integration.** The analysis mesh is formed by three element types (see Figure 2):

- (i) boundary elements: elements placed along the domain boundary,  $\partial\Omega$ . Only a part of each of these elements remains into the problem domain  $\Omega$ . The

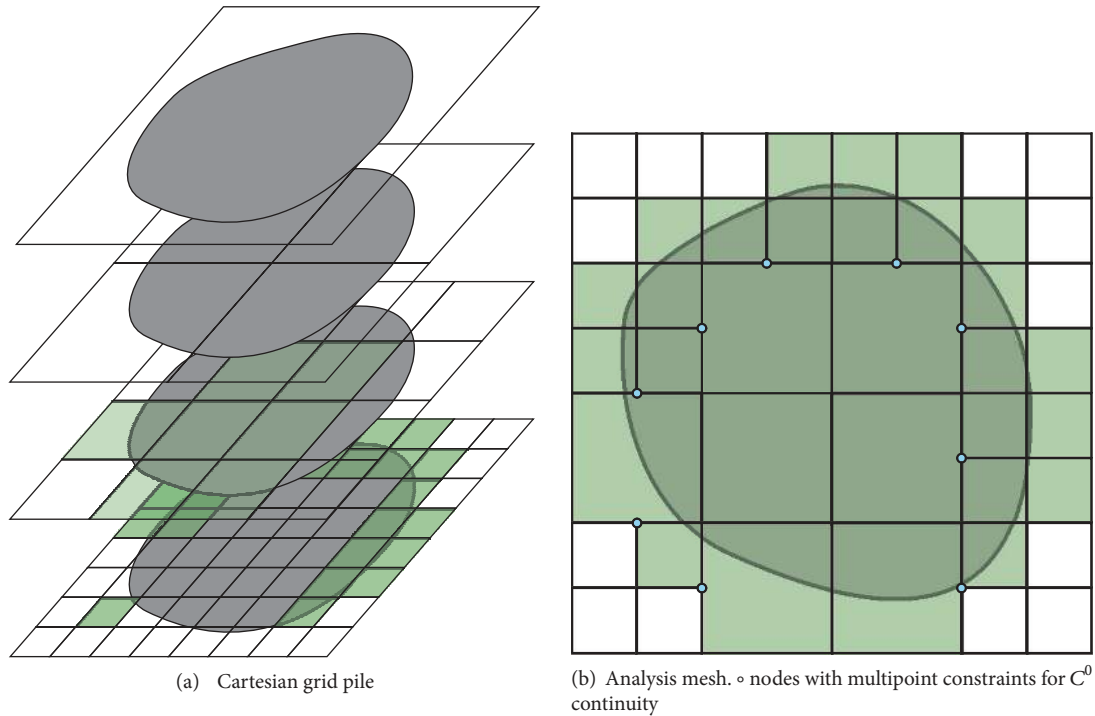


FIGURE 1: Difference between the *Cartesian grid pile* and the analysis mesh.

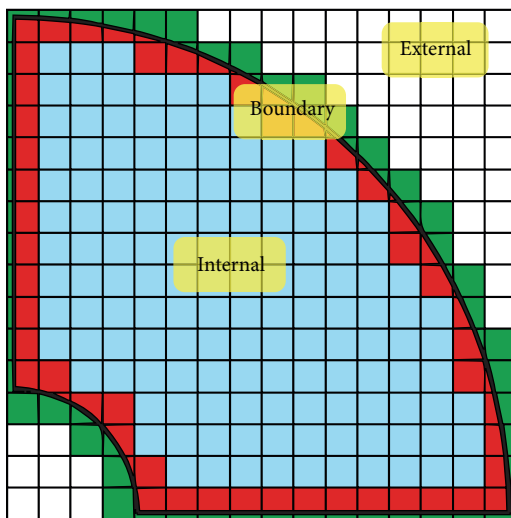


FIGURE 2: Element types according to their relation with the problem boundary: internal elements, external elements, and boundary elements.

evaluation of the stiffness matrix of these elements requires evaluating the intersection between the geometry and the sides of the elements. These intersection points will later be used in order to detect the element area situated into the domain,

- (ii) internal elements: elements fully located into the domain. These are treated as standard FE elements. These elements have the same stiffness matrix as the *reference element* as all of them are geometrically similar to this element,

- (iii) external elements: elements fully located outside of the domain and therefore not considered in the analysis.

The bibliography shows several methods to evaluate the intersection between the domain and the boundary elements and to perform the domain integrals in these elements [16, 28, 29, 52]. The process used in this work is similar to that shown in [29] and consists of three steps:

- (i) intersection of boundary with element edges: Figure 3(a) shows the intersections of the curves that define the boundary of the domain with the Cartesian Grid element edges. The colored zone represents the problem domain,
- (ii) addition of intermediate points: as shown in Figure 3(b) we will identify some extra points placed on the curves that define the boundary. The number of these extra points is related with the curvature of the boundary. This set of points together with the vertex nodes of the element are used to create a Delaunay tessellation that defines integration subdomains at each boundary element,
- (iii) selection of internal triangular subdomains: Figure 3(c) represents the integration subdomains selected to evaluate the domain integral.

For each boundary element this process generates (a) a discretization of the element for domain integration and (b) a discretization of the boundary useful for boundary

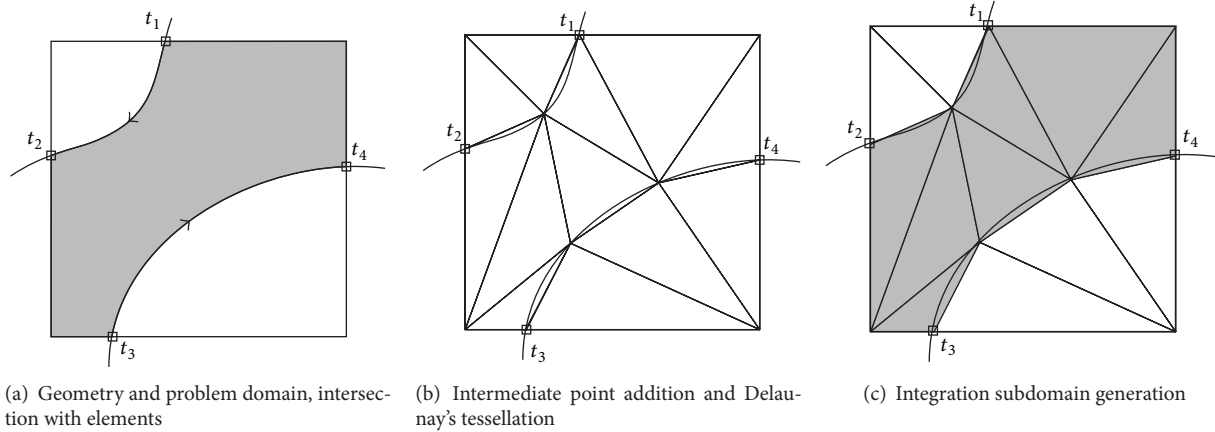


FIGURE 3: Intersection and subdomain generation process in boundary elements.

integration. Thereby, the integration in those elements will be performed using these integration subdomains by means of

$$\int_{\Omega^e \cap \Omega} f(\mathbf{x}) d\Omega = \sum_{i=1}^{nts_e} \int_{\Omega_i^e} f(\mathbf{x}) d\Omega, \quad (4)$$

where  $\Omega_e$  is the domain of element  $e$ ,  $nts_e$  is the number of triangular integration subdomains in  $e$  placed into the problem domain, and  $\Omega_i^e$  is each of the integration subdomains. The curved boundaries can be approximated with straight line segments in these triangular subdomains when Q4 elements are used. Higher order elements would require higher order approximations to the boundary to avoid modeling errors that would reduce the theoretical convergence rate of the FE analysis. Alternatively it is also possible to use the transfinite mapping techniques commonly used in  $p$ -adaptive analysis [53] to consider the exact geometry to increase the accuracy of the results. The use of this mapping increases the computation cost per subdomain but reduces the number of triangular subdomains. Our numerical experience shows that in order to maintain the theoretical convergence rate for elements of order  $p$ , at least  $p$ -order polynomials must be used to approximate the boundary.

Boundary integration will be performed using a similar procedure, discretizing the boundary  $\Gamma$  within element  $e$  into  $nls_e$  integration subdomains. The parametric definition of the boundary provides the exact boundary representation of  $\Gamma_i^e$  which is used for boundary integration instead of approximations like the straight line segments represented in Figure 3:

$$\int_{\Omega^e \cap \Gamma} f(\mathbf{x}) d\Gamma = \sum_{i=1}^{nls_e} \int_{\Gamma_i^e} f(\mathbf{x}) d\Gamma. \quad (5)$$

**3.3. Element Data Sharing.** The use of Cartesian grids together with the data structure used in the implementation of the cg-FEM methodology allows for simple and efficient information data sharing between elements that considerably reduces the total amount of calculations thus improving the computational efficiency of the FE code. This section

describes, for the internal and boundary elements, how information is shared between elements during the analysis.

**3.3.1. Internal Elements.** Reference [49] showed that, in linear elasticity, the terms used to evaluate the stiffness matrix ( $\mathbf{B}$  matrix of derivatives of shape functions and Jacobian matrix  $\mathbf{J}$ ) of geometrically similar elements are related by a constant value evaluated as a function of the scaling factor  $\lambda$  between the elements. In fact, the stiffness matrices of geometrically similar elements are simply related by a factor  $\lambda^{d-2}$  where  $d$  is the number of spatial dimensions. Therefore the evaluation of the stiffness matrices of all the internal elements of the *analysis mesh* is trivial as, for constant material properties, all these elements share the same stiffness matrix, which will be evaluated only for the *reference element* (element used to define the coarsest level, level 0) and then shared with the rest of the internal elements through the hierarchical data structure. This implies a major increase in efficiency of the generation of the system of equations to be solved by the FE code.

**3.3.2. Boundary Elements.** The hierarchical data structure used enables the use on boundary elements of the so-called *vertical data sharing* and *horizontal data sharing* described next.

*h-Adaptive Refinement Process: Vertical Data Sharing.* As previously explained, the evaluation of the stiffness matrix of all internal elements of the *analysis mesh* is trivial. As each boundary element is trimmed differently, each of these elements will require a particular evaluation of the element matrices, following the procedure exposed in Section 3.2. It could be said that the computational cost of the generation of the FE model for the analysis is a function of the number of boundary elements, that is,  $(d - 1)$ -dimensional.

In many  $h$ -adaptive FE codes the previous meshes are discarded and completely new meshes are created as the  $h$ -adaptive analysis evolves, thus preventing the reuse of information evaluated in previous meshes. In our case, the use of Cartesian grids together with the hierarchical data structure

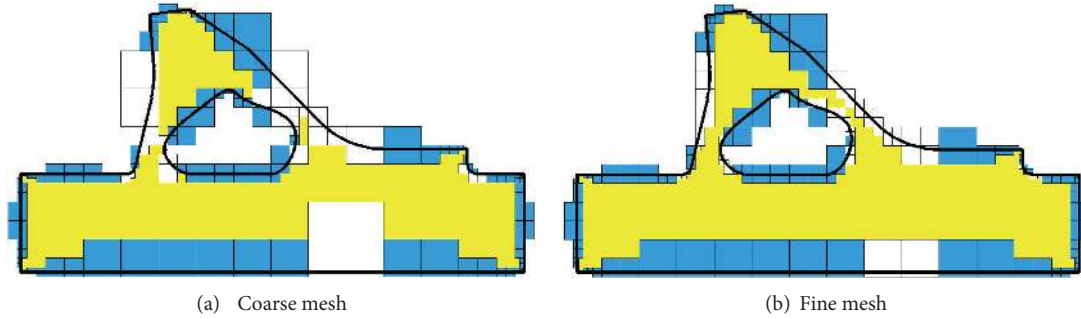


FIGURE 4: Vertical data sharing in two consecutive meshes of the  $h$ -adaptive analysis of a gravity dam. The data structure relates the stiffness matrices of internal elements (yellow) to that of the *reference element*. The calculations for boundary element colored in blue are reused. Element matrices in the finest mesh are only evaluated for white elements.

allows for reusing calculations performed in previous meshes. The hierarchical data structure provides the so-called *vertical data sharing* by means of which elements present in different meshes of the  $h$ -adaptive process will not be reevaluated for the newer meshes.

As an example, Figure 4 shows two consecutive meshes obtained during the  $h$ -adaptive analysis of a gravity dam. Note that elements colored in blue are present in both meshes. The *vertical data sharing* allows for the reuse in the finest mesh of the information evaluated in the coarsest mesh. Note that in the finest mesh new element matrices are only evaluated for white elements.

*Shape Optimization Problems: Horizontal Data Sharing.* The use of optimization technologies for structural shape optimization of mechanical components has increased its use in industry during the last years. The main drawback of these iterative technologies is their high computational cost, which prevents their widespread use. Shape optimization algorithms are composed by two different levels. The *higher level* is in charge of generating the geometries to be considered during the iterative process. On the other hand, the *lower level* is in charge of analyzing each of the geometries proposed by the higher level. A numerical method is used to obtain an approximation to the exact behavior of each geometry. In our case we will use  $cg$ -FEM in this lower level because of the benefits in computational cost obtained when evaluating each of the different geometries but also because data can be shared between different geometries through the so-called *horizontal data sharing* to further improve the overall computational efficiency of the optimization process.

The optimization problem can be formally defined as follows: given a decision space (search space)  $X$ , an objective space (objective values)  $Y$ , the objective function  $f: X \rightarrow Y$  and a set of constraints  $g_j$ :

$$\begin{aligned} & \min f(\mathbf{x}) \\ & \text{where } \mathbf{x} = \{x_i\} \quad i = 1, 2, \dots, n, \\ & \text{under } \mathbf{g}(\mathbf{x}) \leq \{g_j(\mathbf{x})\} \quad j = 1, 2, \dots, m, \\ & a_i \leq x_i \leq b_i \quad i = 1, 2, \dots, n. \end{aligned} \quad (6)$$

In the particular case of structural optimization the objective function (OF)  $f$  is, normally, the weight of the component,  $x_i$  are the design variables (e.g., coordinates of control points) that define the geometry,  $g_j$  are the constraints expressed, normally, in terms of displacements or stresses, and  $a_i$  and  $b_i$  define the side constraints.

There are different optimization algorithms that can be used in the *higher level* to create the different geometries to be analyzed during the optimization process. The benefits of the use of the  $cg$ -FEM methodology could be obtained with any optimization algorithm that would require the use of the FEM to analyze the different geometries during the iterative process. In our case, in the numerical examples, we have considered the use of the genetic algorithm (GA) proposed by Storn and Price [54]. More precisely, we use the *Differential Evolution* (DE) algorithm, version DE1.

With the traditional FEM it is almost impossible to enable the exchange of information between elements of different geometries because, in general, the elements of different meshes are different and completely unrelated as each geometry requires a different mesh conformal to the boundary. However if we use  $cg$ -FEM considering the same *Cartesian grid pile* for all the geometries to be analyzed, we will be able to relate elements used in different geometries making it possible to define a process for *horizontal data sharing*, that is, between elements of different geometries.

Note that the parametric definition of the boundary of the components to be analyzed can be subdivided into two parts:

- (i) the fixed part: this is the part of the boundary that remains fixed in all the geometries (such as the external boundary and the lower straight segment of the internal boundary of the gravity dams represented in Figure 5),
- (ii) the moving part: this is the part of the boundary that would be modified by the optimization algorithm (such as the curved part of the internal boundary of the gravity dams in Figure 5).

The *horizontal data sharing* consists in reusing the computations performed over the elements intersected by the *fixed part* of the boundary in the different geometries analyzed during the optimization process. Figure 5 shows an

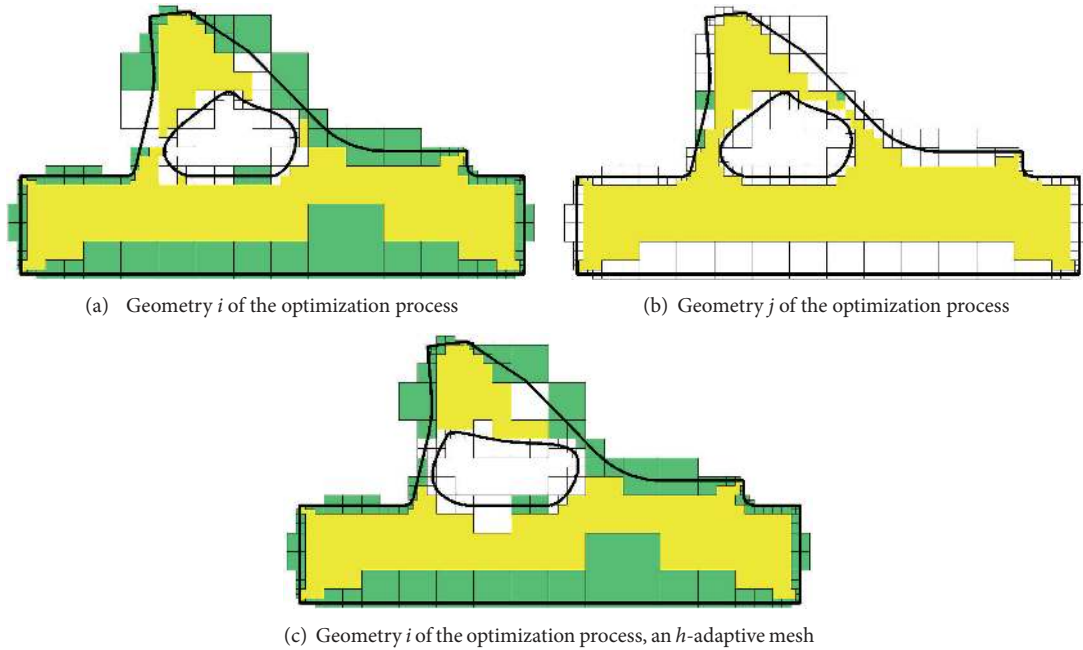


FIGURE 5: Comparison of two different geometries  $i < j$  during an optimization process. The data for the green elements evaluated for geometry  $i$ , in both meshes, are reused in geometry  $j$ .

example of this *horizontal data sharing*. This figure shows two different geometries  $i$  and  $j$  analyzed during the iterative process.  $h$ -adaptive analysis is used to obtain an accurate solution for each geometry as the low accuracy results would negatively affect the performance of the optimization process. In this case green elements represented in Figures 5(a) and 5(b) for geometry  $i$  are reused in geometry  $j$  represented in Figure 5(c). Observe that the *horizontal data sharing* implies a significant reduction of calculations as the information required for most of the boundary elements used in geometry  $j$  was already evaluated in geometry  $i$ . The only element matrices evaluated for the analysis of geometry  $j$  are those corresponding to the white elements.

#### 4. $h$ -Adaptive Refinement

The cost of FE analyses of complex structural components can be reduced by means of the use of  $h$ -adaptive techniques. These techniques can provide an adequate sizing of the elements adapted to the characteristics of the problem. The use of these techniques can provide the required accuracy in the solution with optimized FE models where the number of elements has been minimized, thus reducing the computational cost of the analysis. In some cases, like in structural shape optimization problems,  $h$ -adaptive analysis is a must because inaccurate FE results can negatively affect the behavior of the optimization algorithm [55], leading to nonoptimal solutions, reducing the convergence rate to the optimal solution or even preventing convergence.

cg-FEM implements two  $h$ -refinement procedures. As in [56], the first one is based exclusively on geometrical criteria, whereas the second one considers the quality of the solution and is based on the minimization of the error in the energy norm.

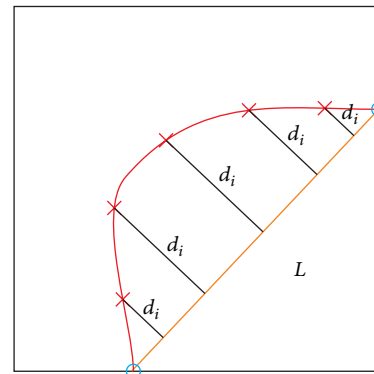


FIGURE 6: Intersection points and distances used for the evaluation of the curvature error indicator  $k$ .

**4.1. Geometrical Refinement.** The analysis starts adapting the Cartesian Grid dimensions domain  $(\Omega_E)$  to the problem domain  $\Omega$  in order to ensure that  $\Omega$  is embedded into  $\Omega_E$ . A preliminary mesh (not used for FE analysis) of uniform element size defined by the user is created as the first step of the analysis process. This preliminary mesh is then intersected with the problem domain. The first analysis mesh is then created following a refinement process based on the geometry of the domain. This procedure consists in refining the boundary elements where the curvature of the boundary is too large with respect to the element size. A simple curvature indicator is defined in (7), where, as represented in Figure 6, the values of  $d_i$  represent the distances between points over the boundary and a straight line segment of length  $L$  defined by the intersection of the boundary with the element sides.  $NIP$  is the number of those intersection points. The refinement process is repeated until the relative

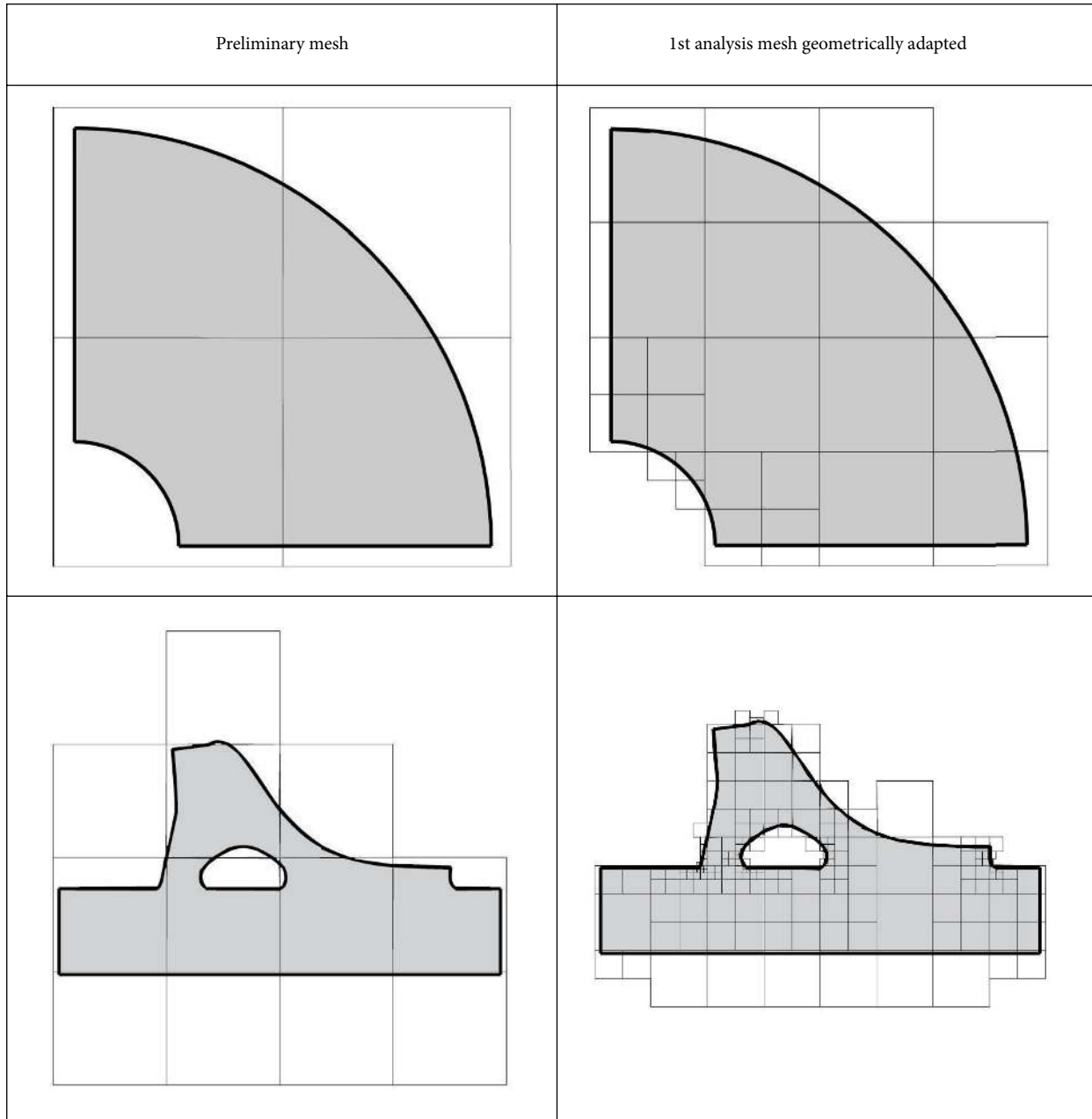


FIGURE 7: Comparison between the preliminary mesh and the geometrically adapted mesh with the curvature criterion.

curvature indicator  $k$  is small enough. The first analysis mesh is created as a result of this process. Figure 7 shows examples of preliminary meshes and first analysis meshes:

$$k = \frac{\sum_{i=1}^{NIP} |d_i|}{NIP \cdot L}. \quad (7)$$

**4.2. Solution-Based Refinement.** After the FE solution of the first analysis mesh has been obtained, new meshes are created following a refinement procedure that takes into account the quality of the FE solution. This procedure aims to minimize the error in energy norm of the solution. In order to estimate the error in energy norm, we use the Zienkiewicz and Zhu

(ZZ) error estimator (8), presented in [34], where  $\sigma^h$  is the FE stress field and  $\sigma^*$  is an improved stress field recovered from  $\sigma^h$ :

$$\|\mathbf{e}_{es}\|^2 = \int_{\Omega} (\sigma^* - \sigma^h)^T \mathbf{D}^{-1} (\sigma^* - \sigma^h) d\Omega. \quad (8)$$

Particularizing (8) to each element domain, we would obtain the estimation of the error in energy norm at element level. With that information, and using a mesh optimization criterion based on the equidistribution of the error in the elements of the mesh to be created [57], we obtain the new levels (sizes) of the elements in each zone. Examples of



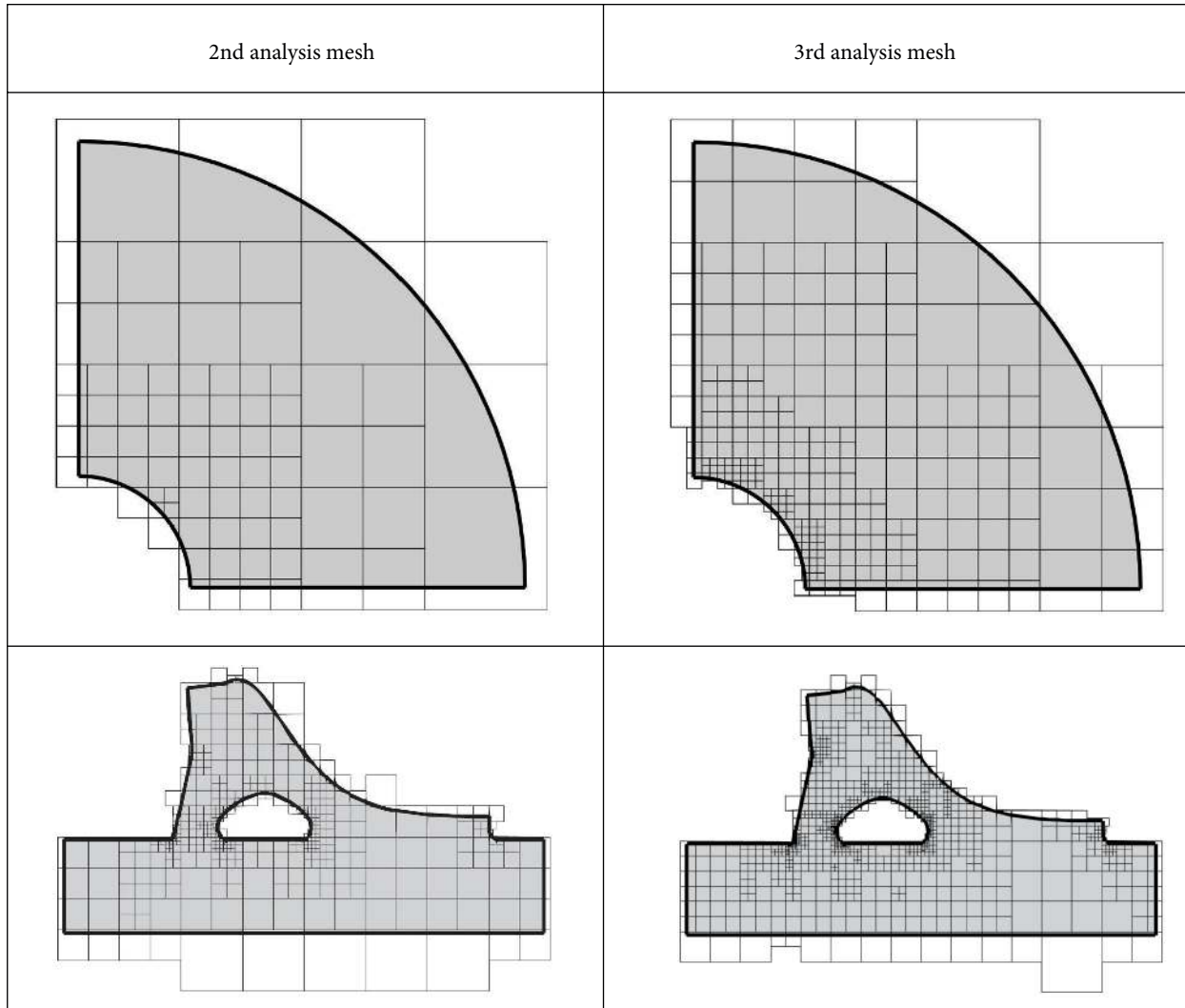


FIGURE 8: Second and third analysis mesh obtained with the error estimation information.

analysis meshes obtained by this procedure are represented in Figure 8.

The numerical results presented in Section 5 will show that, as previously exposed, the use of the cg-FEM increases the computational efficiency of the FE analyses. On the other hand, some authors have reported that the main drawback of this technique is the low accuracy along the boundaries [25, 28]. Loon et al. [52] propose to slightly modify the shape of those elements over the boundary to fit them, generating a mesh conformal to the boundary. One of the benefits of the use of the ZZ error estimator is that the process requires the evaluation of a recovered solution  $\sigma^*$ . Thus, our approach to improve the solution, not only on boundary elements but on all the elements of the analysis mesh, consists in substituting the raw FE solution (obtained by cg-FEM) by the recovered solution  $\sigma^*$  obtained during the error estimation process to guide the adaptive process. The recovery technique used by cg-FEM is based on the Superconvergent Patch Recovery

(SPR) scheme presented and developed in [35, 58–60] by Zienkiewicz and Zhu. The proposed technique is called SPR-CD and provides a recovered displacement field  $\mathbf{u}^*$  that is then used to obtain  $\sigma^*$ . A stress-based version of this technique was presented initially for the traditional FEM in [61], adapted for XFEM in [41, 42] and finally also adapted for cg-FEM in [62]. The displacement-based technique was first presented in [63]. The recovery process produces a recovered displacement field that satisfies the Dirichlet and Neumann boundary conditions and the internal equilibrium equation in each patch of elements attached to the vertex nodes. A continuous recovered field is then obtained using a partition of unity approach as described in the previous references. The recovered solution obtained with the SPR-CD technique is very accurate both along the boundary and in the interior of the domain.

The proposed SPR-CD recovery method is based on the  $L_2$  minimization at each patch of elements (elements

connected to a vertex node) of the difference between the recovered displacement field  $\mathbf{u}^*$  and the FE solution  $\mathbf{u}^h$ , according to (9):

$$\Psi = \int_{\Omega_i} (\mathbf{u}^* - \mathbf{u}^h)^2 d\Omega, \quad (9)$$

where  $\mathbf{u}_i^* = \mathbf{p}\mathbf{a}_i$  with  $i = \{x, y\}$ ,  $\mathbf{p} = \{1, x, y, x^2, \dots\}$  and  $\mathbf{a}_i$  are the vectors of unknown coefficients to be evaluated. Minimizing (9), integrating numerically, and combining the two displacement components, we end up with the linear system of equations shown in (10):

$$\sum_{\text{GP}} \mathbf{P}^T \mathbf{P} |\mathbf{J}| \omega \mathbf{a} = \sum_{\text{GP}} \mathbf{P}^T \mathbf{u}^h |\mathbf{J}| \omega \implies \mathbf{M} \mathbf{a} = \mathbf{G} \quad (10)$$

with  $\mathbf{P} = \begin{bmatrix} \mathbf{p} & 0 \\ 0 & \mathbf{p} \end{bmatrix}$ ,

where  $|\mathbf{J}|$  is the determinant of the Jacobian matrix that provides the mapping between the derivatives of the physical and reference element and  $\omega$  is the weight associate to each integration point.

The equilibrium and Dirichlet constrains are imposed via the Lagrange multipliers technique through matrix  $\mathbf{C}$  in the system of equations shown in (11) used to evaluate the unknown coefficients  $\mathbf{a}_i$ . Once  $\mathbf{a}_i$  have been evaluated, the recovered displacements  $\mathbf{u}_i^*$  can be evaluated at each patch  $i$ . The recovered stresses  $\boldsymbol{\sigma}_i^*$  will be evaluated at each patch from  $\mathbf{u}_i^*$ , yielding a solution pair  $(\mathbf{u}_i^*, \boldsymbol{\sigma}_i^*)$ , that is, both statically and kinematically admissible at each patch:

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ \boldsymbol{\theta} \end{bmatrix} \implies \mathbf{Q} \mathbf{A} = \mathbf{f}. \quad (11)$$

Once the solution  $(\mathbf{u}_i^*, \boldsymbol{\sigma}_i^*)$  is obtained at each patch, it is evaluated for the whole domain  $(\mathbf{u}^*, \boldsymbol{\sigma}^*)$ , (12), by using the *Conjoint Polynomial Enhancement* presented by Blacker and Belytschko in [37] to ensure continuity of both fields:

$$\begin{aligned} \mathbf{u}^* &= \sum_i^{nVN} N_i(x, y) \mathbf{u}_i^*(x, y), \\ \boldsymbol{\sigma}^* &= \sum_i^{nVN} N_i(x, y) \boldsymbol{\sigma}_i^*(x, y). \end{aligned} \quad (12)$$

**4.3. Efficiency of the Recovery Procedure.** A maximum difference of one level is allowed by the  $h$ -adaptive process between two adjacent elements. Because of this and the topological features of the Cartesian grid, only a reduced number of possible patch configurations can be obtained. Figure 9 represents the 19 possible configurations of internal patches for 2D. Note that internal patches are defined as the patches composed by internal elements only.

The polynomial coefficients  $\mathbf{a}$  used to describe the recovered displacement field according to (11) are obtained for a normalized coordinate system. Then, matrix  $\mathbf{Q}$  in (11) will be exactly the same for all internal patches having the same configuration. This implies that we will only need to invert

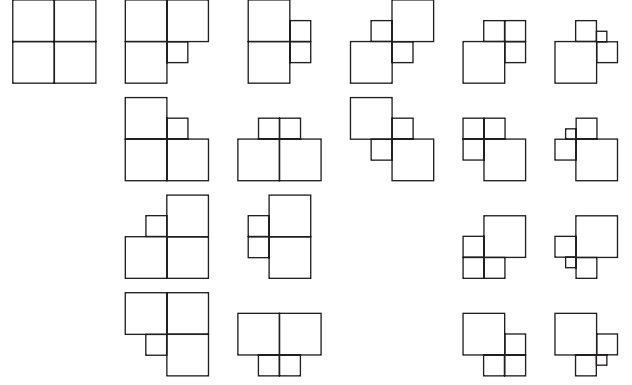


FIGURE 9: Possible configurations of internal patches in 2D.

a maximum of 19  $\mathbf{Q}$  matrices to obtain the recovered field in all the internal patches. The first step of the recovery process for the internal patches consists in codifying the configuration of each patch and classifying them according to the configurations shown in Figure 9; then the values of  $\mathbf{Q}^{-1}$  are evaluated for each of the different patch configurations. Once the  $f$  terms in (11) have been evaluated for each patch, the unknown coefficients  $\mathbf{A}$  are evaluated from  $\mathbf{A} = \mathbf{Q}^{-1} \mathbf{f}$ . This procedure considerably reduces the computational cost associated to the evaluation of internal patches. The computational cost associated to the evaluation of internal patches is negligible with respect to the cost associated to the evaluation of patches that contain boundary elements as each of these patches will have different  $\mathbf{Q}$  matrices. In practice, the computational cost of the recovery process is only depending on the number of patches along the boundary. That implies a  $(d - 1)$ -dimensional computational cost.

**4.4. Error Estimation of the Recovered Solution: Stopping Criterion.** With this novel recovery procedure we have an improved solution  $(\mathbf{u}^*, \boldsymbol{\sigma}^*)$  more accurate than the raw FE solution  $(\mathbf{u}^h, \boldsymbol{\sigma}^h)$ . The recovered solution can therefore be used as the standard output provided by cg-FEM instead of the FE solution. Then the accuracy of the recovered solution needs to be evaluated. This means that an error estimator for the recovered solution should be available. An error estimation of the recovered field was first introduced in the report [64]. This report shows that the error in energy norm of the recovered solution  $\|\mathbf{e}^*\|$  can be estimated from

$$\begin{aligned} \|\mathbf{e}^*\|^2 &= \int_{\Omega} \mathbf{e}^* s d\Omega + \int_{\Gamma_N} \mathbf{e}^* \mathbf{r} d\Gamma \approx \int_{\Omega} |\mathbf{e}_{\text{es}}^T \mathbf{s}| d\Omega \\ &+ \int_{\Gamma_N} |\mathbf{e}_{\text{es}}^T \mathbf{r}| d\Gamma = \|\mathbf{e}^*\|_{\text{es}}^2, \end{aligned} \quad (13)$$

where  $\mathbf{e}^* = \mathbf{u} - \mathbf{u}^*$  is the exact error of the recovered displacement field,  $\mathbf{e}_{\text{es}} = \mathbf{u}^* - \mathbf{u}^h$  is the estimated error of the FE displacements,  $\mathbf{s} = -\text{div}(\boldsymbol{\sigma}^*) - \mathbf{b}$  represents the lack of internal equilibrium of the recovered stresses, and  $\mathbf{r} = \boldsymbol{\Pi} \boldsymbol{\sigma}^* - \mathbf{t}$  represents the lack of equilibrium on Neumann boundaries, where  $\boldsymbol{\Pi}$  is the matrix that projects the boundary stress field into tractions over the boundary. Reference [42] showed

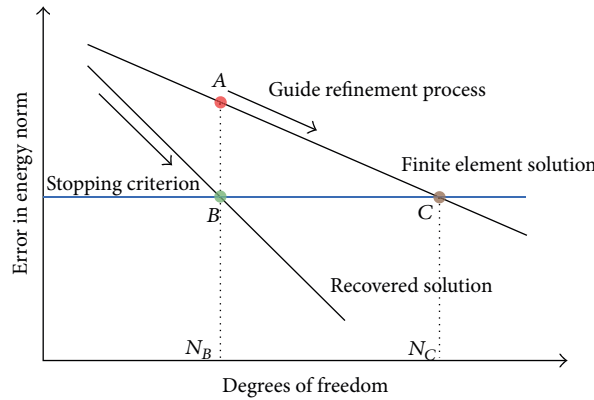


FIGURE 10:  $h$ -adaptive refinement scheme.

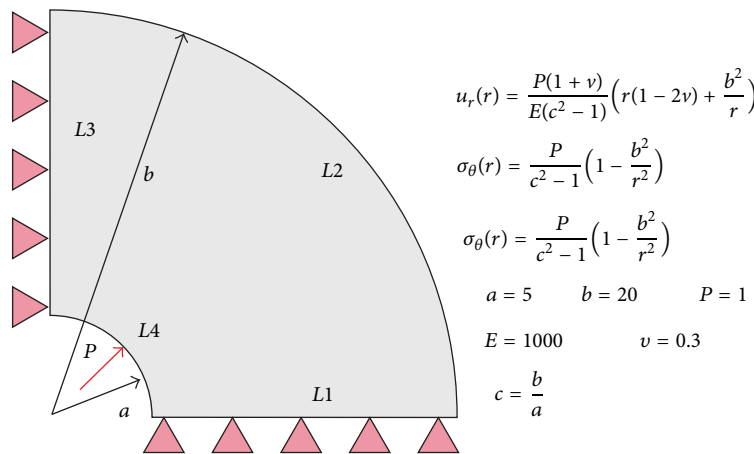


FIGURE 11: Problem 1: cylinder model and analytical solution.

that as the recovery technique produces an almost exact satisfaction of the boundary equilibrium equation ( $\mathbf{r} \approx 0$ ), then, the second integral in (13) (boundary integral) is negligible when compared with the first integral (domain integral). Therefore, only the first integral in (13) will be considered for the error estimation of the recovered solution.

Figure 10 shows a scheme of the evolution of the exact errors in energy norm of the finite element and recovered solutions. Note that in SPR-based recovery techniques the recovered solution is in practice more accurate than the FE solution and has a higher convergence rate. Therefore, the number of degrees of freedom  $N_B$  required by the recovered solution to obtain a prescribed accuracy level  $\gamma$  defined by blue horizontal line in Figure 10 is considerably smaller than that required by the FE solution ( $N_C$ ). We can thus define a highly efficient  $h$ -adaptive refinement process that considers the accuracy of the recovered solution instead of the accuracy of the FE solution. The  $h$ -adaptive refinement process of cg-FEM will, as usual, be guided by the error estimation of the FE solution using (8). The error reduction of the FE solution simultaneously produces an error reduction of the recovered solution that is estimated using (13). The refinement process will run until the estimated error of the recovered solution

reaches the prescribed accuracy level. As a resume, the error of the FE solution  $\|\mathbf{e}\|$  is used to drive the  $h$ -adaptive process whereas the error of the recovered solution  $\|\mathbf{e}^*\|$  is used to define the stopping criterion.

### 5. Numerical Examples

This section will present numerical examples to evaluate the improvements provided by cg-FEM.

**5.1. Problem 1: Cylinder under Internal Pressure.** Let us consider a thick wall cylinder subjected to an internal pressure and plain strain conditions, as described in Figure 11. The analytical solution of this problem is known and is also given in Figure 11. Only a quarter of the section has been considered in the model together with the appropriate symmetry conditions.

Figures 12 and 13 show the evolution of the errors in energy norm and the error convergence rate, respectively, for bilinear (Q4) and biquadratic (Q8) elements. In the graphs on the left of these figures we observe that the exact and estimated errors of the FE solution (blue and red lines) almost overlap. We can see that the cg-FEM results converge

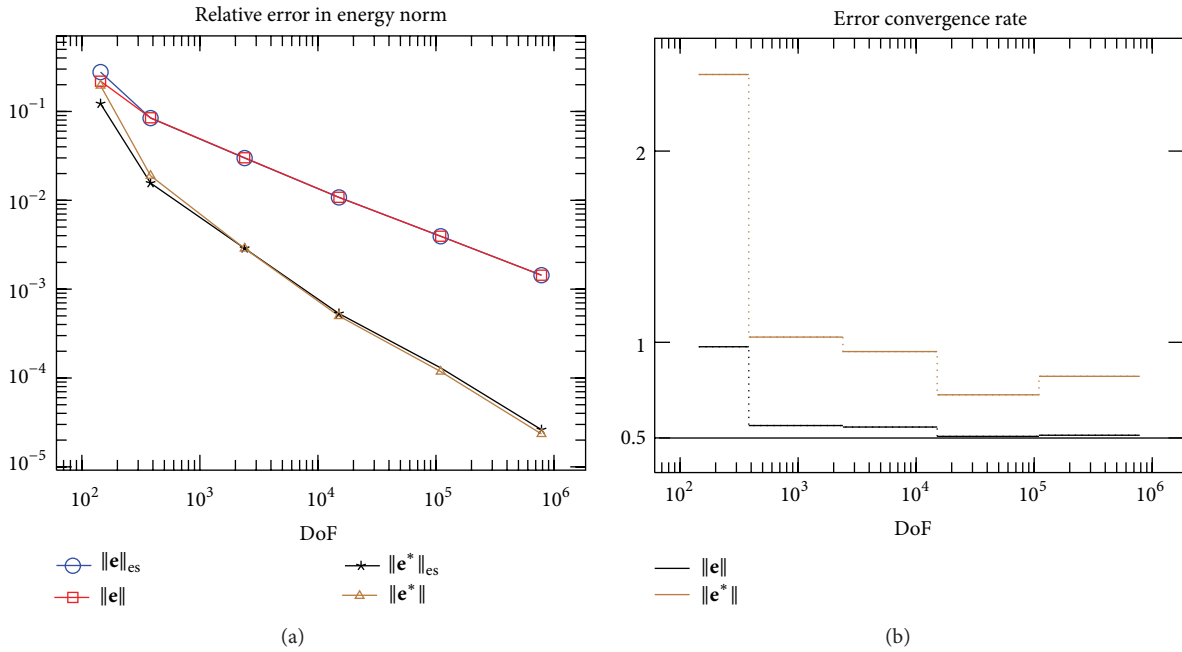


FIGURE 12: Problem 1: Cylinder Q4. Study of evolution of error in energy norm.

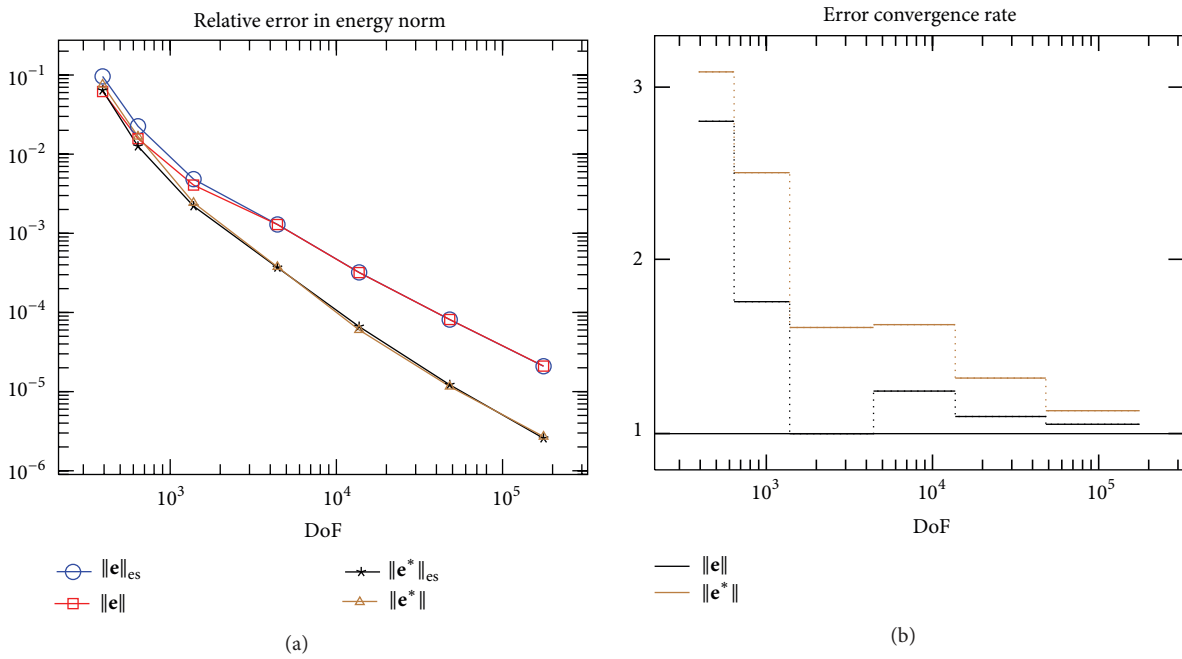


FIGURE 13: Problem 1: Cylinder Q8. Study of evolution of error in energy norm.

smoothly to the exact solution and that our recovery-based error estimator provides very accurate results. This graph also shows that the brown and black curves that represent the evolution of the exact and estimated errors of the recovered solution also overlap, clearly presenting a higher accuracy and convergence rate than the FE solution. The convergence rates of the exact error for the FE and recovered solutions are shown in the graphs on the right of the figures. The theoretical convergence rates of the exact error in energy norm of the FE

solution with respect to the number of degrees of freedom are 0.5 and 1.0 for Q4 and Q8 elements. In all cases the recovered solution has presented higher convergence rates than the FE solution.

The accuracy of the error estimation techniques for the FE and recovered solutions can be clearly evaluated by means of the use of the effectivity index  $\theta$  represented in Figure 14 for Q4 and Q8 elements and for the error estimators of the FE and recovered solutions. The effectivity index is

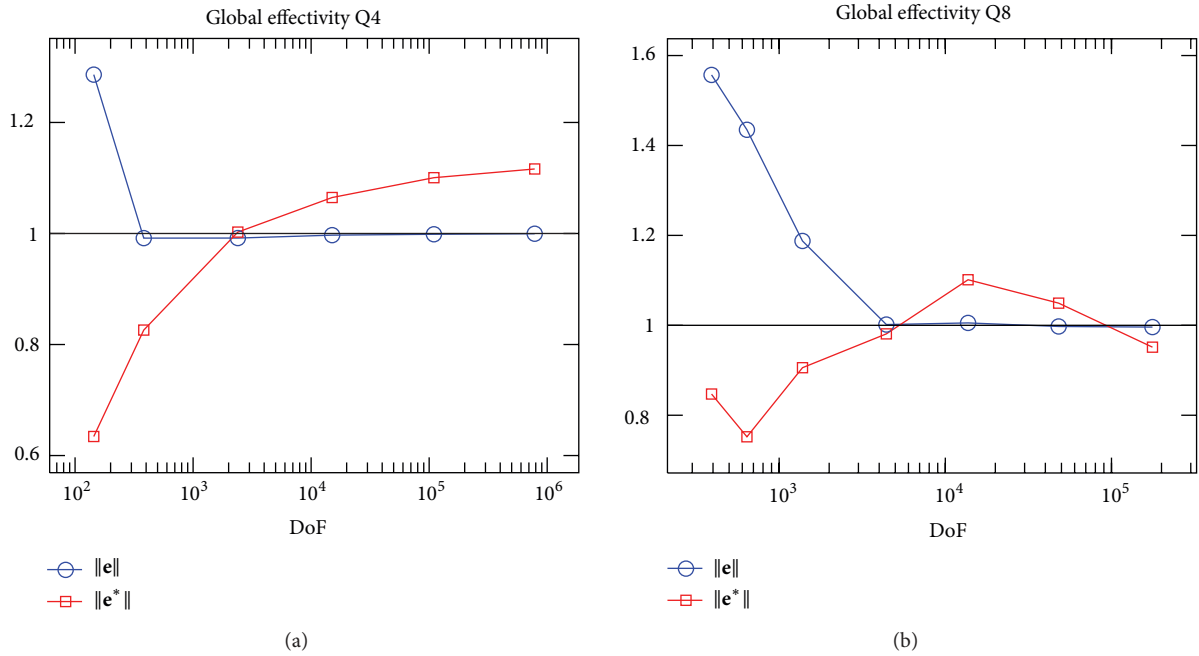


FIGURE 14: Problem 1: error estimator effectivity study.

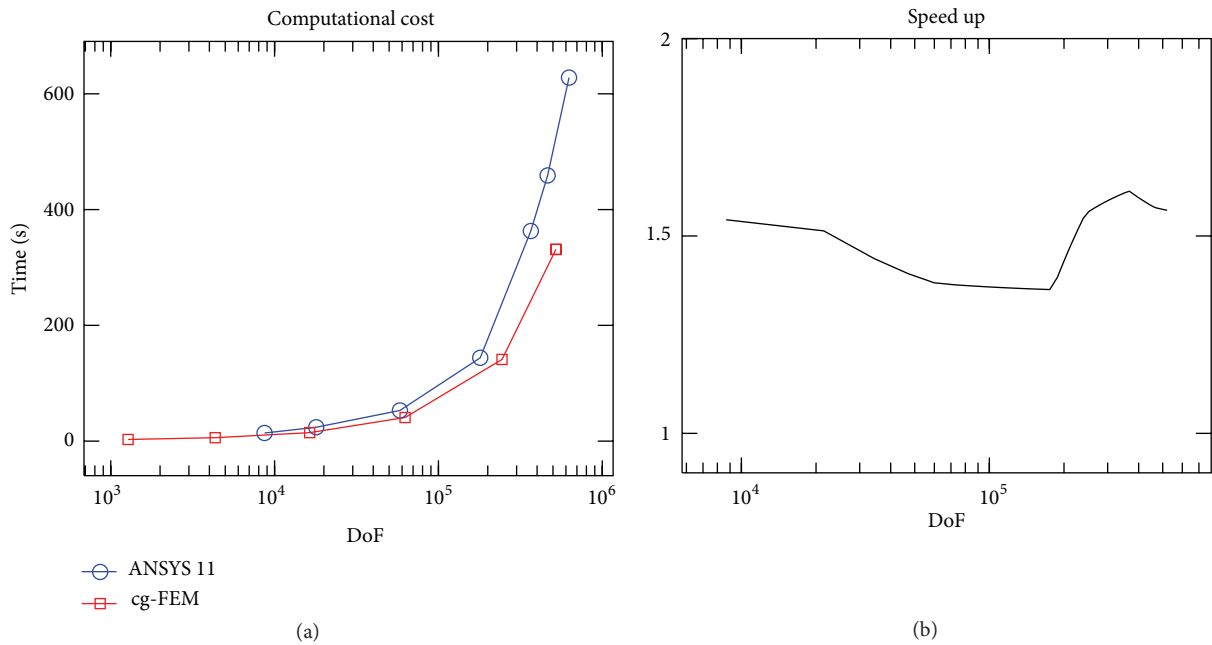


FIGURE 15: Problem 1: Cylinder Q8. Computational cost and speedup for uniform meshes.

defined as the ratio of the estimated to the exact errors in energy norm. Thus, values close to 1 would indicate accurate error estimators. This figure clearly shows that both error estimators are very accurate. Although the error estimator in energy norm of the FE solution is more accurate than that of the recovered solution, the error estimator of this last solution presents a very good behavior and can be used to effectively evaluate its accuracy.

Regarding the computational cost, we have used this problem to compare cg-FEM with the commercial code ANSYS 11 considering uniform meshes. This could not be taken as a strict comparison because the commercial code works with a more complex data structure. However, our purpose here is to show that the computational cost of the cg-FEM is comparable with that obtained by widely used commercial codes. In Figure 15 we show how the

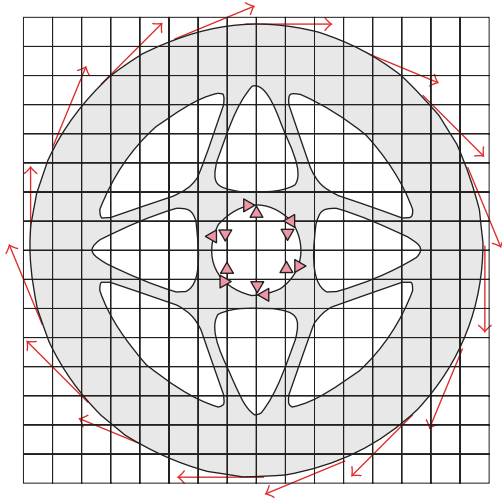


FIGURE 16: Problem 2: Flywheel: model of the problem, loads, and constrains.

computational cost is always lower for the proposed code than that for the commercial code. On the right hand side we show the speedup obtained. It is worth to mention that the cg-FEM code is fully implemented in Matlab 2010b and no subroutine has been compiled. As represented in this figure the speedup obtained with our cg-FEM implementation is about 1.5. In addition, we also have to consider that the commercial code is postprocessing the solution with a nodal averaging technique, whereas cg-FEM uses an advanced recovery process and evaluates two error estimations, one for the FE solution and other one for the recovered solution. This is a numerical evidence that the cg-FEM methodology will be highly competitive with a standard FEM compilation.

**5.2. Problem 2: Flywheel.** This problem with a more complex geometry represents a flywheel under tangential tractions along the external surface and constrained displacements in the internal surface as shown in Figure 16. The material is aluminium with elastic modulus  $E = 70$  GPa and Poisson coefficient  $\mu = 0.33$ . Plane stress conditions are considered.

Figures 17 and 18 present the set of the three first  $h$ -adapted meshes for the analysis of this component with Q4 and Q8 elements. The first mesh of each set has been obtained using the geometrical refinement criterion. The following meshes have been created with the  $h$ -adaptivity process based on the error estimation in energy norm.

This problem has no analytical solution; however we can check the behavior of the cg-FEM code considering the convergence rate of the error of the FE solution [45–47]. We have to check that, in the asymptotic range, this convergence rate tends to the theoretical value (0.5 for Q4 elements and 1 for Q8). We can verify this in Figures 19 and 20 where the evolution of the estimated error and its convergence rate have been evaluated for the FE and recovered solutions. We can observe that the convergence rate of the FE results tends to theoretical values in both cases (blue lines). Furthermore, the estimated error of the recovered solution (red lines) presents

a smooth behavior and a higher (and stable) convergence rate than the FE solution along the refinement process.

### 5.3. Structural Shape Optimization

**5.3.1. Problem 3: Pipe Optimization.** To check the performance of the cg-FEM combined with the optimization software, we have chosen an optimization problem with known analytical solution. Our objective in this problem is to minimize the cross-sectional area of a tube under internal pressure  $P$  applied on the circular internal surface, with unknown external surface, where the Von Mises stresses must be below the yield stress  $S_y$ .

As shown in Figure 21, two symmetry surfaces are considered. The external surface is given by a cubic spline defined by three points. The optimal analytical solution corresponds to a circular external surface. For  $P = 0.9 \cdot 10^6$  and  $S_y = 2 \cdot 10^6$  the optimal external radius is  $R_0 = 10.67033824461$  and the corresponding area is  $A_{\text{opt}} = 69.787307715081$ .

30 individuals per generation have been considered in the genetic algorithm used for the optimization. We have performed analyses with Q4 elements for different prescribed error levels  $\gamma \leq (2.0, 5.0)\%$  to study the influence of the error in energy norm on the computational cost and the accuracy of the solution provided by the optimization process. We have considered two types of  $h$ -adaptive analysis techniques to evaluate the numerical solution of each geometry.

- (i) Strategy a: we have used the FE solution  $\sigma^h$  as the output of the analysis together with a criterion to stop the adaptive process based on  $\|\mathbf{e}\|_{\text{es}}$ , see (8).
- (ii) Strategy b: we have used the recovered solution  $\sigma^*$  and a criterion to stop the adaptive process based on the error estimator for this solution  $\|\mathbf{e}^*\|_{\text{es}}$ ; see (13).

Figure 22 presents the results of this analysis. The graph on the left shows the evolution of the exact relative error in area (evaluated from the exact known analytical solution for this problem) of the best solutions provided by the process versus the number of generations. We can observe that the optimization process will provide accurate solutions only for the most restrictive error levels. Although we used the same values for both strategies, the best results are obtained for *Strategy b* ( $\sigma^*$  curves). The reason for this could be that, given an FE solution and a recovered solution with similar global errors in energy norm, the maximum value of the Von Mises stress (used as a control parameter during the optimization process) is usually very accurate in the recovered solution as it will appear on the boundary where this solution has been forced to satisfy the boundary constraints. The graph on the right shows the evolution of the computational cost along the optimization process. It can be clearly observed that the use of *Strategy b* that takes into account the recovered solution to drive the optimization process results in a computing time considerably lower than the time required with *Strategy a*, driven by the FE solution. The speedup obtained with *Strategy b* is of about 3.3 for  $\gamma = 5.0\%$  and 3.1 for  $\gamma = 2.0\%$ . Thus, this speedup together with the considerable increase in the accuracy displayed in the graph on the left of the figure

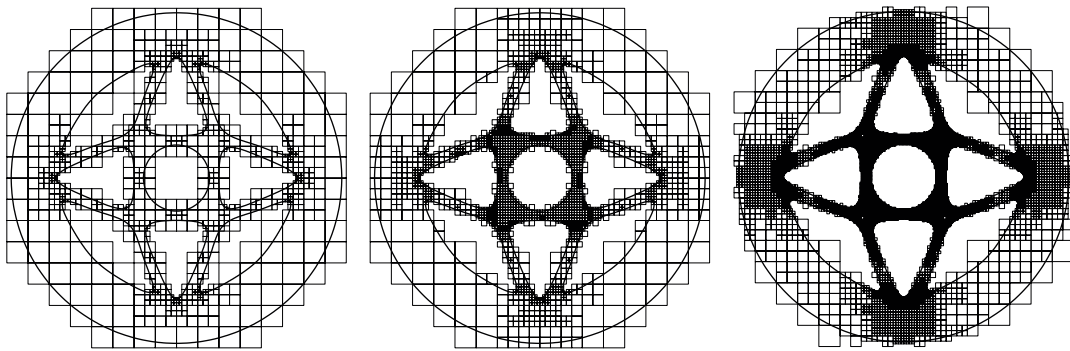


FIGURE 17: Problem 2: Flywheel Q4. Sequence of  $h$ -adapted meshes.

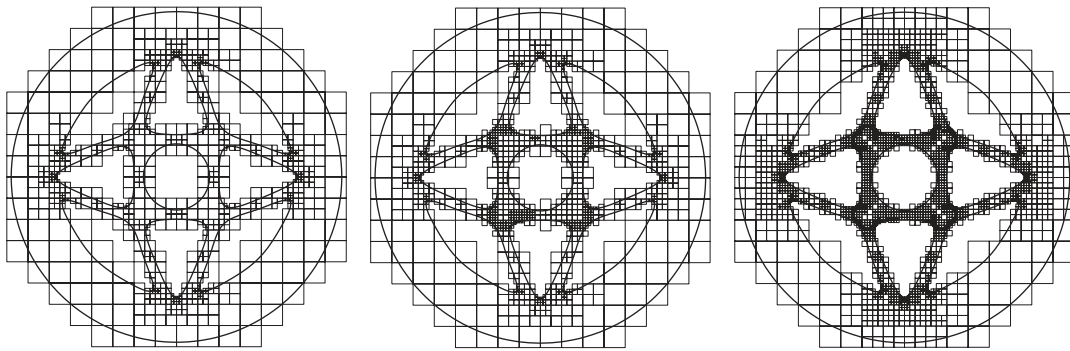


FIGURE 18: Problem 2: Flywheel Q8. Sequence of  $h$ -adapted meshes.

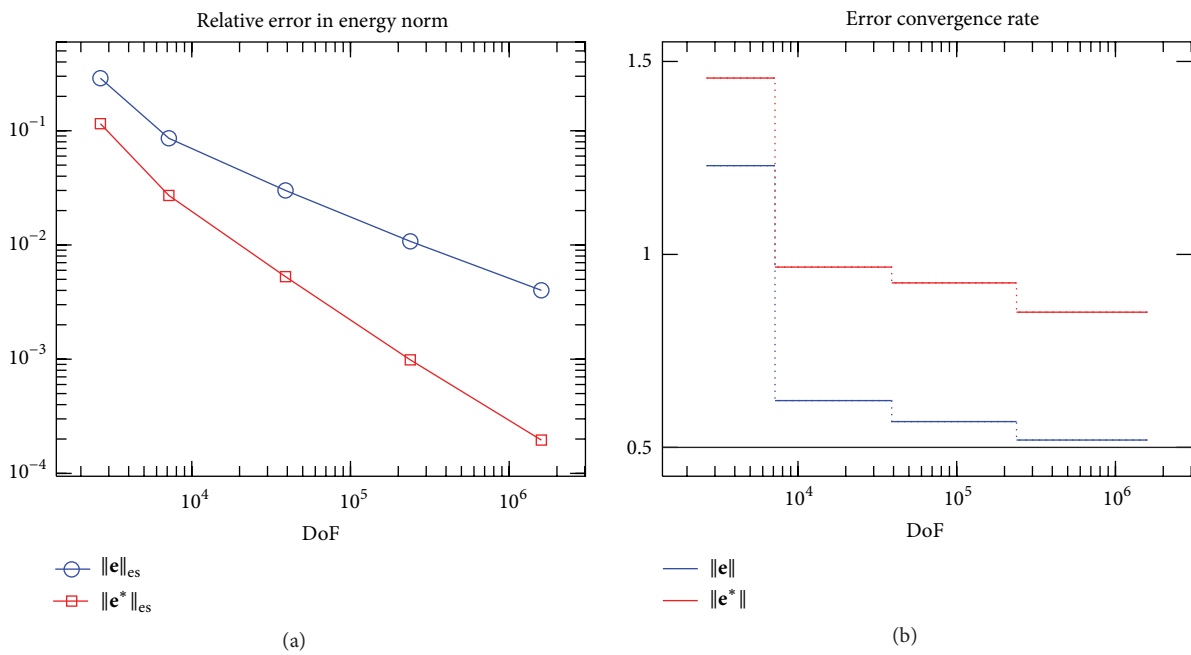


FIGURE 19: Problem 2: Flywheel Q4. Study of evolution of error in energy norm.

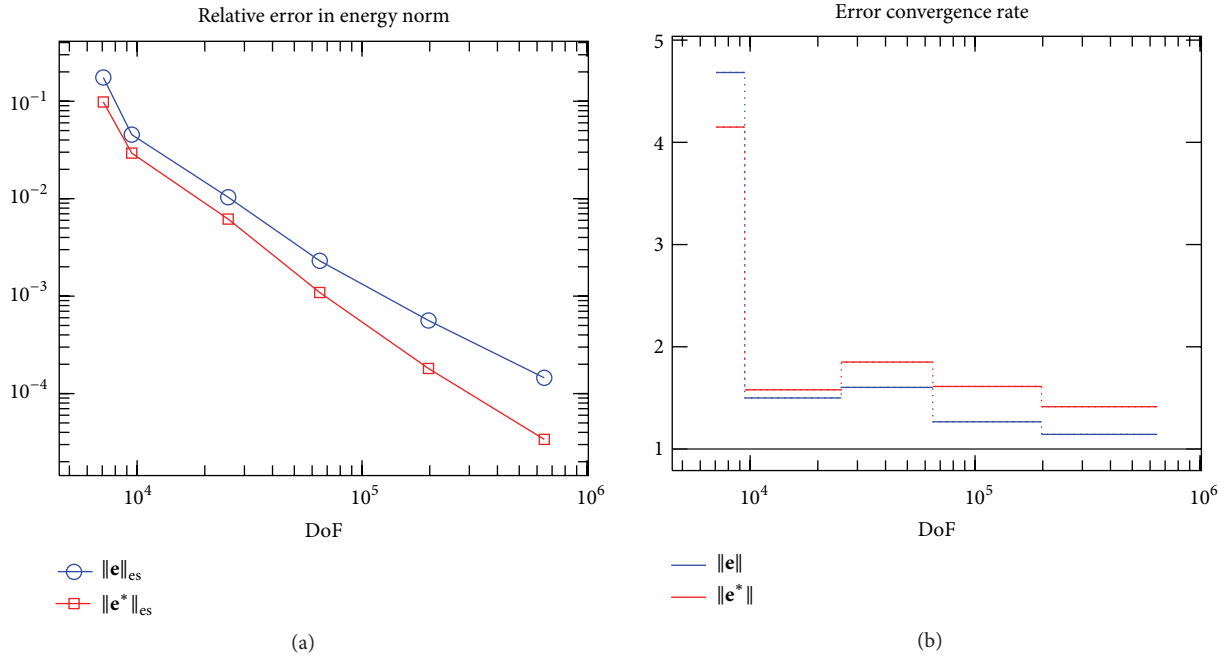


FIGURE 20: Problem 2: Flywheel Q8. Study of evolution of error in energy norm.

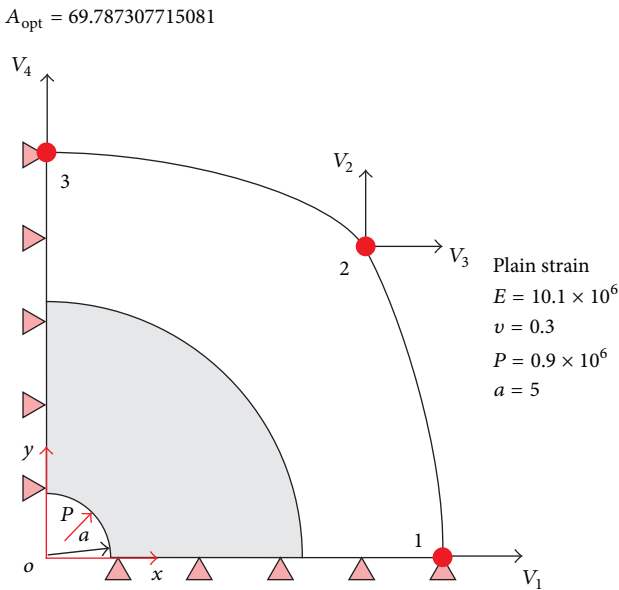


FIGURE 21: Problem 3. Pipe under internal pressure. Design variables and optimal geometry (shaded area).

clearly shows that the use of Strategy b, that is, the use of the recovered solution to drive the optimization process, produces remarkable benefits with respect to the use of the FE solution.

## 6. Conclusions

In this work, we have presented a novel numerical technique, cg-FEM, to solve Boundary Value Problems. This technique,

which has been implemented to solve 2D linear elasticity problems, is based on the use of  $h$ -adaptive Finite Element Analysis and has three main characteristics:

- (1) use of Cartesian meshes, nonconformal to the boundary of the domain, to decrease mesh burden, focusing the computational cost of the mesh generation process on the domain's boundary which must be intersected with the so-called boundary elements;
- (2) use of a hierarchical data structure that allows reusing previous computations, reducing to the minimum the total number of calculations; and
- (3) prominent use of a recovered solution ( $\sigma^*$ ,  $u^*$ ) obtained with the SPR-CD technique, for which an efficient error estimator in energy norm is available, with two main objectives:

- (a) to drive the  $h$ -adaptive analysis based on recovery-based error estimation techniques,
- (b) to replace the raw FE solution ( $\sigma^h$ ,  $u^h$ ), commonly used as the standard output of the analysis, in order to improve accuracy and reduce the size of the FE model required to reach the prescribed accuracy level.

The numerical results have shown that the FE solution of cg-FEM smoothly converges to the exact solution with (approximately) the theoretical convergence rate. Furthermore, the recovered solution, as a result of being based on the Superconvergent Patch Recovery technique, is more accurate and converges with a higher convergence rate than the FE solution. An error estimator for the recovered solution, whose accuracy is only supported by numerical tests, has been used.



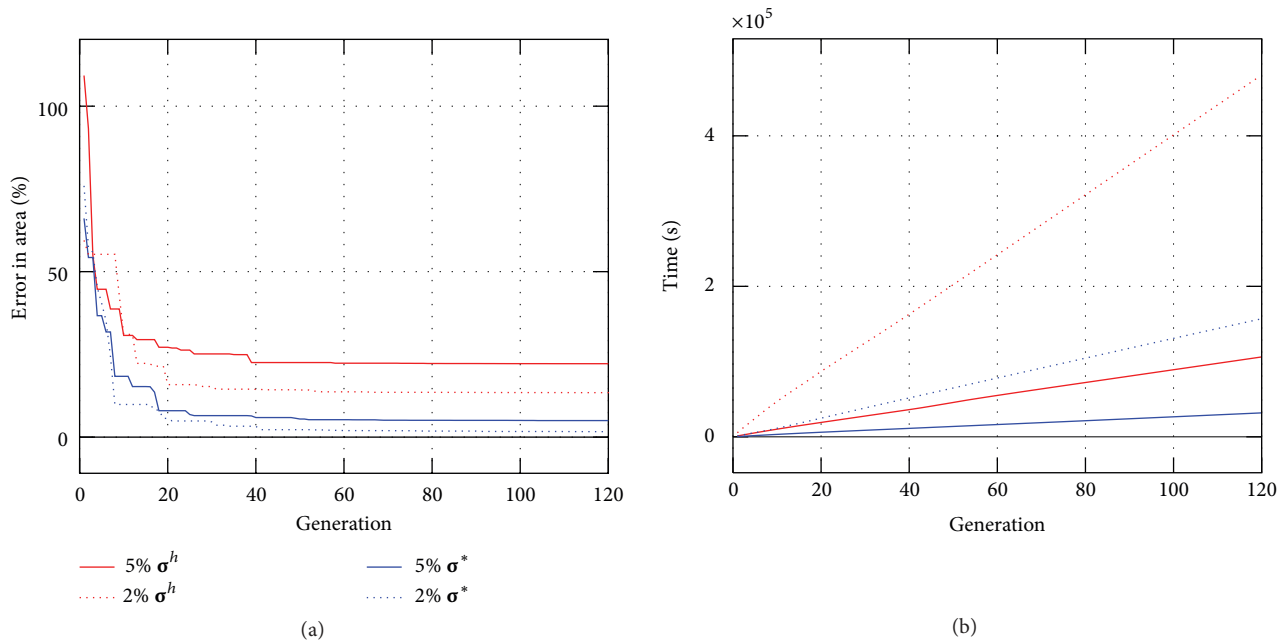


FIGURE 22: Problem 3. Evolution of pipe optimization considering the error estimation of the FE solution ( $\sigma^h$ ) or the recovered one ( $\sigma^*$ ), for different prescribed errors levels  $\gamma$ . (a) Evolution of exact error in area with respect to the optimal analytical solution, (b) accumulative computational cost of the optimization processes.

Further investigations to provide a theoretical basis for this error estimator are now in process.

The cg-FEM technology has allowed for decreasing the complexity of the problem to one less dimension in the vast majority of the routines. As a result basic numerical tests have shown that even its Matlab implementation can be computationally more efficient than commercial codes based on the traditional FEM. A 3D implementation of cg-FEM under development has already provided promising preliminary results.

The numerical results have shown how shape optimization processes can benefit from the use of cg-FEM, leading to a substantial reduction in computational cost that would help to increase the use of optimization processes in industrial applications.

## Acknowledgments

This work has been developed within the framework of research project DPI2010-20542 of the Ministerio de Economía y Competitividad (Spain). The financial support of the FPU program (AP2008-01086), the funding from Universitat Politècnica de València, and Generalitat Valenciana (PROMETEO/2012/023) are also acknowledged. The authors also thank the support of the Framework Programme 7 Initial Training Network Funding under Grant no. 289361 “Integrating Numerical Simulation and Geometric Design Technology.”

## References

- [1] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, Wiley, 1st edition, 2009.
- [2] N. Moës, J. Dolbow, and T. Belytschko, “A finite element method for crack growth without remeshing,” *International Journal for Numerical Methods in Engineering*, vol. 46, no. 1, pp. 131–150, 1999.
- [3] N. Sukumar and J. H. Prévost, “Modeling quasi-static crack growth with the extended finite element method. Part I: computer implementation,” *International Journal of Solids and Structures*, vol. 40, no. 26, pp. 7513–7537, 2003.
- [4] T. Strouboulis, K. Copps, and I. Babuška, “The generalized finite element method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 32–33, pp. 4081–4193, 2001.
- [5] T. Strouboulis, L. Zhang, and I. Babuška, “Assessment of the cost and accuracy of the generalized FEM,” *International Journal for Numerical Methods in Engineering*, vol. 69, no. 2, pp. 250–283, 2007.
- [6] J. M. Melenk and I. Babuška, “The partition of unity finite element method: basic theory and applications,” *Computer Methods in Applied Mechanics and Engineering*, vol. 139, no. 1–4, pp. 289–314, 1996.
- [7] M. Stolarska, D. L. Chopp, N. Moës, and T. Belytschko, “Modelling crack growth by level sets in the extended finite element method,” *International Journal for Numerical Methods in Engineering*, vol. 51, no. 8, pp. 943–960, 2001.
- [8] M. Moumnassi, S. Belouettar, E. Béchet, S. P. Bordas, D. Quoirin, and M. Potier-Ferry, “Finite element analysis on implicitly defined domains: an accurate representation based on arbitrary parametric surfaces,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 5–8, pp. 774–796, 2011.
- [9] W. D. Lian, G. Legrain, and P. Cartraud, “Image-based computational homogenization and localization: comparison between X-FEM/levelset and voxel-based approaches,” *Computational Mechanics*, vol. 51, no. 3, pp. 279–293, 2012.
- [10] G. Legrain, N. Chevaugeon, and K. Dréau, “High order X-FEM and levelsets for complex microstructures: uncoupling

- geometry and approximation,” *Computer Methods in Applied Mechanics and Engineering*, vol. 241–244, pp. 172–189, 2012.
- [11] E. Burman and P. Hansbo, “Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 41–44, pp. 2680–2686, 2010.
  - [12] E. Heikkola, Y. A. Kuznetsov, and K. N. Lipnikov, “Fictitious domain methods for the numerical solution of three-dimensional acoustic scattering problems,” *Journal of Computational Acoustics*, vol. 7, no. 3, pp. 161–193, 1999.
  - [13] U. Hetmaniuk and C. Farhat, “A finite element-based fictitious domain decomposition method for the fast solution of partially axisymmetric sound-hard acoustic scattering problems,” *Finite Elements in Analysis and Design*, vol. 39, no. 8, pp. 707–725, 2003.
  - [14] C. Farhat and U. Hetmaniuk, “A fictitious domain decomposition method for the solution of partially axisymmetric acoustic scattering problems. Part I: Dirichlet boundary conditions,” *International Journal for Numerical Methods in Engineering*, vol. 54, no. 9, pp. 1309–1332, 2002.
  - [15] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy, “An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries,” *Journal of Computational Physics*, vol. 156, no. 2, pp. 209–240, 1999.
  - [16] A. Jahangirian and Y. Shoraka, “Adaptive unstructured grid generation for engineering computation of aerodynamic flows,” *Mathematics and Computers in Simulation*, vol. 78, no. 5–6, pp. 627–644, 2008.
  - [17] T. Preusser, M. Rumpf, and L. Schwen, “Finite element simulation of bone microstructures,” in *Proceedings of the 14th FE Workshop*, Ulm, Germany, July 2007.
  - [18] C. M. Silva Santos and D. M. Greaves, “Using hierarchical Cartesian grids with multigrid acceleration,” *International Journal for Numerical Methods in Engineering*, vol. 69, no. 8, pp. 1755–1774, 2007.
  - [19] I. Ramière, P. Angot, and M. Belliard, “Fictitious domain methods to solve convection-diffusion problems with general boundary conditions,” in *Proceedings of the 17th Computational Fluid Dynamics Conference AIAA*, Toronto, Canada, 2005.
  - [20] G. DeRose Jr. and A. R. Díaz, “Solving three-dimensional layout optimization problems using fixed scale wavelets,” *Computational Mechanics*, vol. 25, no. 2, pp. 274–285, 2000.
  - [21] G. Jang, Y. Kim, and K. Choy, “Remesh-free shape optimization using the wavelet-Galerkin method,” *International Journal of Solids and Structures*, vol. 41, no. 22–23, pp. 6465–6483, 2004.
  - [22] R. A. E. Mäkinen, T. Rossi, and J. Toivanen, “A moving mesh fictitious domain approach for shape optimization problems,” *Mathematical Modelling and Numerical Analysis*, vol. 34, no. 1, pp. 31–45, 2000.
  - [23] M. Victoria, O. M. Querin, and P. Martí, “Topology design for multiple loading conditions of continuum structures using isolines and isosurfaces,” *Finite Elements in Analysis and Design*, vol. 46, no. 3, pp. 229–237, 2010.
  - [24] L. Parussini and V. Pediroda, “Fictitious domain approach with  $hp$ -finite element approximation for incompressible fluid flow,” *Journal of Computational Physics*, vol. 228, no. 10, pp. 3891–3910, 2009.
  - [25] J. Bishop, “Rapid stress analysis of geometrically complex domains using implicit meshing,” *Computational Mechanics*, vol. 30, no. 5–6, pp. 460–478, 2003.
  - [26] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu, “Immersed finite element method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 21–22, pp. 2051–2067, 2004.
  - [27] A. M. Roma, C. S. Peskin, and M. J. Berger, “An adaptive version of the immersed boundary method,” *Journal of Computational Physics*, vol. 153, no. 2, pp. 509–534, 1999.
  - [28] M. García-Ruiz and G. Steven, “Fixed grid finite elements in elasticity problems,” *Engineering Computations*, vol. 16, no. 2, pp. 145–164, 1999.
  - [29] F. Daneshmand and M. J. Kazemzadeh-Parsi, “Static and dynamic analysis of 2D and 3D elastic solids using the modified FGFEM,” *Finite Elements in Analysis and Design*, vol. 45, no. 11, pp. 755–765, 2009.
  - [30] S. P. A. Bordas, T. Rabczuk, N. X. Hung et al., “Strain smoothing in FEM and XFEM,” *Computers & Structures*, vol. 88, no. 23–24, pp. 1419–1443, 2010.
  - [31] R. N. Simpson, S. P. A. Bordas, J. Trevelyan, and T. Rabczuk, “A two-dimensional isogeometric boundary element method for elastostatic analysis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 209–212, pp. 87–100, 2012.
  - [32] M. Scott, R. Simpson, J. Evans et al., “Isogeometric boundary element analysis using unstructured T-splines,” *Computer Methods in Applied Mechanics and Engineering*, vol. 254, pp. 197–221, 2013.
  - [33] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs, “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39–41, pp. 4135–4195, 2005.
  - [34] 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*, vol. 24, no. 2, pp. 337–357, 1987.
  - [35] O. C. Zienkiewicz and J. Z. Zhu, “The superconvergent patch recovery and a posteriori error estimates. Part I: the recovery technique,” *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, pp. 1331–1364, 1992.
  - [36] N. E. Wiberg, F. Abdulwahab, and S. Ziukas, “Enhanced superconvergent patch recovery incorporating equilibrium and boundary conditions,” *International Journal for Numerical Methods in Engineering*, vol. 37, no. 20, pp. 3417–3440, 1994.
  - [37] T. Blacker and T. Belytschko, “Superconvergent patch recovery with equilibrium and conjoint interpolant enhancements,” *International Journal for Numerical Methods in Engineering*, vol. 37, no. 3, pp. 517–536, 1994.
  - [38] A. C. A. Ramsay and E. A. W. Maunder, “Effective error estimation from continuous, boundary admissible estimated stress fields,” *Computers & Structures*, vol. 61, no. 2, pp. 331–343, 1996.
  - [39] J. J. Ródenas, M. Tur, F. J. Fuenmayor, and A. Vercher, “Improvement of the superconvergent patch recovery technique by the use of constraint equations: the SPR-C technique,” *International Journal for Numerical Methods in Engineering*, vol. 70, no. 6, pp. 705–727, 2007.
  - [40] P. Díez, J. J. Ródenas, and O. C. Zienkiewicz, “Equilibrated patch recovery error estimates: simple and accurate upper bounds of the error,” *International Journal for Numerical Methods in Engineering*, vol. 69, no. 10, pp. 2075–2098, 2007.
  - [41] J. J. Ródenas, E. Giner, F. J. Fuenmayor, and O. A. González-Estrada, “Accurate recovery-type error estimation for linear elastic fracture mechanics in FEM and XFEM based on a singular+smooth field splitting,” in *Proceedings of the International Conference on Adaptive Modeling and Simulation (ADMOS '07)*, pp. 202–205, International Center for Numerical Methods in Engineering (CIMNE), 2007.
  - [42] J. J. Ródenas, O. A. González-Estrada, P. Díez, and F. J. Fuenmayor, “Accurate recovery-based upper error bounds for

- the extended finite element framework,” *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 37–40, pp. 2607–2621, 2010.
- [43] M. Ainsworth and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, John Wiley & Sons, Chichester, UK, 2000.
- [44] Q. Z. Xiao and B. L. Karihaloo, “Improving the accuracy of XFEM crack tip fields using higher order quadrature and statically admissible stress recovery,” *International Journal for Numerical Methods in Engineering*, vol. 66, no. 9, pp. 1378–1410, 2006.
- [45] S. P. A. Bordas and M. Duflot, “Derivative recovery and a posteriori error estimate for extended finite elements,” *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 35–36, pp. 3381–3399, 2007.
- [46] S. P. A. Bordas, M. Duflot, and P. Le, “A simple error estimator for extended finite elements,” *Communications in Numerical Methods in Engineering*, vol. 24, no. 11, pp. 961–971, 2008.
- [47] M. Duflot and S. P. A. Bordas, “A posteriori error estimation for extended finite elements by an extended global recovery,” *International Journal for Numerical Methods in Engineering*, vol. 76, no. 8, pp. 1123–1138, 2008.
- [48] J. J. Ródenas, O. A. González-Estrada, J. E. Tarancón, and F. J. Fuenmayor, “A recovery-type error estimator for the extended finite element method based on singular+smooth stress field splitting,” *International Journal for Numerical Methods in Engineering*, vol. 76, no. 4, pp. 545–571, 2008.
- [49] J. J. Ródenas, J. E. Tarancón, J. Albelda, A. Roda, and F. J. Fuenmayor, “Hierarchical properties in elements obtained by subdivision: a hierarchical h-adaptivity program,” in *Adaptive Modeling and Simulation 2005*, P. Díez and N. E. Wiberg, Eds., 2005.
- [50] J. F. Abel and M. S. Shephard, “An algorithm for multipoint constraints in finite element analysis,” *International Journal for Numerical Methods in Engineering*, vol. 14, no. 3, pp. 464–467, 1979.
- [51] C. Farhat, C. Lacour, and D. Rixen, “Incorporation of linear multipoint constraints in substructure based iterative solvers. Part I: a numerically scalable algorithm,” *International Journal for Numerical Methods in Engineering*, vol. 43, no. 6, pp. 997–1016, 1998.
- [52] R. Loon, P. Anderson, J. Hart, and F. Baaijens, “A combined fictitious domain/adaptive meshing method for fluid-structure interaction in heart valves,” *International Journal for Numerical Methods in Fluids*, vol. 46, no. 5, pp. 533–544, 2004.
- [53] B. Guo and I. Babuška, “The  $h$ - $p$  version of the finite element method,” *Computational Mechanics*, vol. 1, no. 1, pp. 21–41, 1986.
- [54] R. Storn and K. Price, “Differential evolution—a simple and efficient adaptive scheme for global optimisation over continuous spaces,” Tech. Rep. TR-95-012, International Computer Science Institute, Berkeley, Calif, USA, 1995.
- [55] J. J. Ródenas, G. Bugada, J. Albelda, and E. Oñate, “On the need for the use of error-controlled finite element analyses in structural shape optimization processes,” *International Journal for Numerical Methods in Engineering*, vol. 87, no. 11, pp. 1105–1126, 2011.
- [56] R. Löhner, J. Cebra, M. Castro et al., “Adaptive embedded unstructured grid methods,” in *Mecánica computacional*, G. Buscaglia, E. Dari, and O. Zamonsky, Eds., vol. 23, Bariloche, Argentina, 2004.
- [57] F. J. Fuenmayor and J. L. Oliver, “Criteria to achieve nearly optimal meshes in the  $h$ -adaptive finite element method,” *International Journal for Numerical Methods in Engineering*, vol. 39, no. 23, pp. 4039–4061, 1996.
- [58] O. C. Zienkiewicz and J. Z. Zhu, “The superconvergent patch recovery and a posteriori error estimates. Part 2: error estimates and adaptivity,” *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, pp. 1365–1382, 1992.
- [59] O. C. Zienkiewicz, J. Z. Zhu, and J. Wu, “Superconvergent patch recovery techniques—some further tests,” *Communications in Numerical Methods in Engineering*, vol. 9, no. 3, pp. 251–258, 1993.
- [60] O. C. Zienkiewicz and J. Z. Zhu, “Superconvergence and the superconvergent patch recovery,” *Finite Elements in Analysis and Design*, vol. 19, no. 1–2, pp. 11–23, 1995.
- [61] J. J. Ródenas, O. A. González-Estrada, P. Díez, and F. J. Fuenmayor, “Upper bounds of the error in the extended finite element method by using an equilibrated-stress patch recovery technique,” in *Proceedings of the International Conference on Adaptive Modeling and Simulation (ADMOS '07)*, pp. 210–213, International Center for Numerical Methods in Engineering (CIMNE), 2007.
- [62] E. Nadal, S. Bordas, J. J. Ródenas, J. E. Tarancón, and M. Tur, “Accurate stress recovery for the two-dimensional fixed grid finite element method,” in *Proceedings of the 10th International Conference on Computational Structures Technology*, pp. 1–20, 2010.
- [63] E. Nadal, O. A. González-Estrada, J. J. Ródenas, S. P. A. Bordas, and F. J. Fuenmayor, “Error estimation and error bounding in energy norm based on a displacement recovery technique,” in *Proceedings of the 6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS '12)*, Ecomas, Vienna, Austria, 2012.
- [64] E. Nadal, O. González-Estrada, J. Ródenas, and F. Fuenmayor, “Report: Error estimation of recovered solution in FE analysis,” 1–31, November 2012.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

