

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Department of Mechanical Engineering and Material Engineering



Ph.D. Thesis

**Cartesian grid FEM (cgFEM): High performance
h-adaptive FE analysis with efficient error control.
Application to structural shape optimization**

Presented by: D. Enrique Nadal Soriano

Supervised by: Dr. D. Fco. Javier Fuenmayor Fernández

Dr. D. Juan José Ródenas García

Valencia, January, 2014

Ph.D. THESIS

**Cartesian grid FEM (cgFEM): High performance
h-adaptive FE analysis with efficient error control.
Application to structural shape optimization**

for the degree of

Doctor in Engineering and Industrial Production

presented by

D. Enrique Nadal Soriano

at the

Department of Mechanical Engineering and Material
Engineering

at Universitat Politècnica de València

Supervised by

Dr. D. Fco. Javier Fuenmayor Fernández

Dr. D. Juan José Ródenas García

Valencia, January, 2014

Ph.D THESIS

**CARTESIAN GRID FEM (cgFEM): HIGH
HIGH PERFORMANCE *H*-ADAPTIVE FE ANALYSIS WITH
EFFICIENT ERROR CONTROL. APPLICATION TO
STRUCTURAL SHAPE OPTIMIZATION**

Presented by: D. Enrique Nadal Soriano

Supervised by: Dr. D. Fco. Javier Fuenmayor Fernández

Dr. D. Juan José Ródenas García

QUALIFYING TRIBUNAL

PRESIDENT: Dr. D. _____

VOCAL: Dr. D. _____

SECRETARY: Dr. D. _____

Valencia, January, 2014

Abstract

More and more challenging designs are required everyday in today's industries. The traditional trial and error procedure commonly used for mechanical parts design is not valid any more since it slows down the design process and yields suboptimal designs. For structural components, one alternative consists in using shape optimization processes which provide optimal solutions. However, these techniques require a high computational effort and require extremely efficient and robust Finite Element (FE) programs. FE software companies are aware that their current commercial products must improve in this sense and devote considerable resources to improve their codes. In this work we propose to use the Cartesian Grid Finite Element Method, cgFEM as a tool for efficient and robust numerical analysis. The cgFEM methodology developed in this thesis uses the synergy of a variety of techniques to achieve this purpose, but the two main ingredients are the use of Cartesian FE grids independent of the geometry of the component to be analyzed and an efficient hierarchical data structure. These two features provide to the cgFEM technology the necessary requirements to increase the efficiency of the cgFEM code with respect to commercial FE codes. As indicated in [1, 2], in order to guarantee the convergence of a structural shape optimization process we need to control the error of each geometry analyzed. In this sense the cgFEM code also incorporates the appropriate error estimators. These error estimators are specifically adapted to the cgFEM framework to further increase its efficiency. This work introduces a solution recovery technique, denoted as SPR-CD, that

in combination with the Zienkiewicz and Zhu error estimator [3] provides very accurate error measures of the FE solution. Additionally, we have also developed error estimators and numerical bounds in Quantities of Interest based on the SPR-CD technique to allow for an efficient control of the quality of the numerical solution. Regarding error estimation, we also present three new upper error bounding techniques for the error in energy norm of the FE solution, based on recovery processes. Furthermore, this work also presents an error estimation procedure to control the quality of the recovered solution in stresses provided by the SPR-CD technique. Since the recovered stress field is commonly more accurate and has a higher convergence rate than the FE solution, we propose to substitute the raw FE solution by the recovered solution to decrease the computational cost of the numerical analysis. All these improvements are reflected by the numerical examples of structural shape optimization problems presented in this thesis. These numerical analysis clearly show the improved behavior of the cgFEM technology over the classical FE implementations commonly used in industry.

Resumen

Cada día diseños más complejos son requeridos por las industrias actuales. Para el diseño de nuevos componentes, los procesos tradicionales de prueba y error usados comúnmente ya no son válidos ya que ralentizan el proceso y dan lugar a diseños sub-óptimos. Para componentes estructurales, una alternativa consiste en usar procesos de optimización de forma estructural los cuales dan como resultado diseños óptimos. Sin embargo, estas técnicas requieren un alto coste computacional y también programas de Elementos Finitos (EF) extremadamente eficientes y robustos. Las compañías de programas de EF son conocedoras de que sus programas comerciales necesitan ser mejorados en este sentido y destinan importantes cantidades de recursos para mejorar sus códigos. En este trabajo proponemos usar el Método de Elementos Finitos basado en mallados Cartesianos (cgFEM) como una herramienta eficiente y robusta para el análisis numérico. La metodología cgFEM desarrollada en esta tesis usa la sinergia entre varias técnicas para lograr este propósito, cuyos dos ingredientes principales son el uso de los mallados Cartesianos de EF independientes de la geometría del componente que va a ser analizado y una eficiente estructura jerárquica de datos. Estas dos características confieren a la tecnología cgFEM de los requisitos necesarios para aumentar la eficiencia del código cgFEM con respecto a códigos comerciales. Como se indica en [1, 2], para garantizar la convergencia del proceso de optimización de forma estructural se necesita controlar el error en cada geometría analizada. En este sentido el código cgFEM también incorpora los apropiados estimadores

de error. Estos estimadores de error han sido específicamente adaptados al entorno cgFEM para aumentar su eficiencia. En esta tesis se introduce un proceso de recuperación de la solución, llamado SPR-CD, que en combinación con el estimador de error de Zienkiewicz y Zhu [3], da como resultado medidas muy precisas del error de la solución de EF. Adicionalmente, también se han desarrollado estimadores de error y cotas numéricas en Magnitudes de Interés basadas en la técnica SPR-CD para permitir un eficiente control de la calidad de la solución numérica. Respecto a la estimación de error, también se presenta un proceso de estimación de error para controlar la calidad del campo de tensiones recuperado obtenido mediante la técnica SPR-CD. Ya que el campo recuperado es por lo general más preciso y tiene un mayor orden de convergencia que la solución de EF, se propone sustituir la solución de EF por la solución recuperada para disminuir así el coste computacional del análisis numérico. Todas estas mejoras se han reflejado en esta tesis mediante ejemplos numéricos de problemas de optimización de forma estructural. Los resultados numéricos muestran claramente un mejor comportamiento de la tecnología cgFEM con respecto a implementaciones clásicas de EF comúnmente usadas en la industria.

Resum

Cada dia dissenys més complexos són requerits per les indústries actuals. Per al disseny de nous components, els processos tradicionals de prova i error usats comunament ja no són vàlids ja que ralentinzen el procés i donen lloc a dissenys subòptims. Per a components estructurals, una alternativa consisteix a usar processos d'optimització de forma estructural els quals donen com resultat dissenys òptims. No obstant això, estes tècniques requerixen un alt cost computacional i també programes d'Elements Finites (EF) extremadament eficients i robustos. Les companyies de programes d'EF són coneixedores que els seus programes comercials necessiten ser millorats en este sentit i destinen importants quantitats de recursos per a millorar els seus codis. En este treball proposem usar el Mètode d'Elements Finites basat en mallats Cartesians (cgFEM) com una ferramenta eficient i robusta per a l'anàlisi numèrica. La metodologia cgFEM desenrotllada en esta tesi usa la sinergia entre diverses tècniques per a aconseguir este propòsit, els dos ingredients principals de la qual són l'ús dels mallats Cartesians d'EF independents de la geometria del component que serà analitzat i una eficient estructura jeràrquica de dades. Estes dos característiques conferixen a la tecnologia cgFEM dels requisits necessaris per a augmentar l'eficiència del codi cgFEM respecte a codis comercials. Com s'indica en [1, 2], per a garantir la convergència del procés d'optimització de forma estructural es necessita controlar l'error en cada geometria analitzada. En este sentit el codi cgFEM també incorpora els apropiats estimadors d'error. Estos estimadors d'error han sigut específicament adaptats a l'entorn cgFEM per a

augmentar la seua eficiència. En esta tesi s'introduïx un procés de recuperació de la solució, anomenat SPR-CD, que en combinació amb l'estimador d'error de Zienkiewicz i Zhu [3], dóna com resultat mesures molt precises de l'error de la solució d'EF. Addicionalment, també s'han desenrotllat estimadors d'error i cotes numèriques en Magnituds d'Interés basades amb la tècnica SPR-CD per a permetre un eficient control de la qualitat de la solució numèrica. Respecte a l'estimació d'error, també es presenta un procés d'estimació d'error per a controlar la qualitat del camp de tensions recuperat obtingut mitjançant la tècnica SPR-CD. Ja que el camp recuperat és generalment més precís i té un major orde de convergència que la solució d'EF, es proposa substituir la solució d'EF per la solució recuperada per a disminuir així el cost computacional de l'anàlisi numèrica. Totes estes millores s'han reflectit en esta tesi per mitjà d'exemples numèrics de problemes d'optimització de forma estructural. Els resultats numèrics mostren clarament un millor comportament de la tecnologia cgFEM respecte a implementacions clàssiques d'EF comunament usades en la indústria.

Acknowledgments

This story started some time ago when I was in Padova. One day I received an email offering me the opportunity to apply for a grant to get my Ph.D. The sender of this email was Prof. Juan José Ródenas. A few days later and with the support of my family I decided to accept this life-changing challenge. That was the beginning of this new path. A difficult and long path with one of the worst global economic crises as background. But I have walked this path yet, and I did not make it through alone. The path sometimes was downhill and other times was very steep, but even in the most adverse situations there was someone to help me out when it was necessary. This Thesis is a milestone in my life. This Thesis means the end of my stage at UPV and the beginning of a new one full of opportunities and challenges.

First, I would like to thank Prof. Juan José Ródenas for aiming me to start this new path that has opened my eyes towards a world that was unknown to me. Also for his patience, kindness and comprehension in those moments when I got hopelessly stuck. He was always transmitting his energy and enthusiasm to keep going. These attributes that most of people and specially researchers are claiming for. I would also extend my deep gratitude to Prof. Fco. Javier Fuenmayor, the Head of this awesome research team, who perfectly transmits his personality into it. I am grateful for his advice in both, research and personal levels.

During this period I have made new friends among my workmates at DIMM. They have always been near to me with smiles and also with their support in the moments when the things were not so easy. Also, in those moments when the ambient permitted a more relaxing atmosphere, as such as Christmas and summer dinners. They have been an outstanding group in which everyone grows as researcher and more importantly, as person. The list of nice people at DIMM is large and I would not mind talking a bit of all of them but it is better to summarize my feelings in a single word: family. However, I believe that an special consideration should be taken to certain special people such as: Fede “El regletas”, who is always protecting us against malevolent Karpesky, Jose “Axo” who is always there with an amusing smile that makes women mad, Eva “La becaria” whose patience is even bigger than her heart, Onofre with his peculiar humor and Octavio “El Morenito” who was always ready for endless discussions about work and things that are not exactly related to work. I wish him success in the enterprise he has recently started in Colombia. I want also to remember two persons, Virginia and Rafa, who also leaved an indelible mark on me.

Along these years I have been traveling a lot and luckily I met a number of interesting people. I would like to highlight the Cardiff team, especially Prof. S.P.A. Bordas and Dr. P. Kerfriden for their support during my research stay at Cardiff University and during the collaboration that we have mantained since then. I would like to thank the help and support of Prof. P. Díez and Prof. J.P. Moitinho for their advice in research matters that have considerably increased my skills as researcher. Finally, I would like to mention two colleagues, Dr. M. Rüter and Dr. T. Gerasimov, because it has been an enormous luck to meet them.

This path I am finishing would have been impossible without the support of my friends, both in my hometown, Planes, and also in Valencia. They are now one of the pillars I lean on. They are always worrying about my problems and backing me up to overcome them. Among them, I would like to highlight Juanjo “Ronie”, Iñaki, María, “Mary”, Juny, Marcos “Pon”, Carlos, Moncia, Jordi “El Amo”, Juan “Chuanes”, Sandra “La Pona”, Raquel “Caperu”,

Marina, Alba, Jorge “Mini”, Samu, etc. I would like to specially remark the comprehension and the strong friendship shown by someone that has been quite special during these years, Irene, especially now that the situation has become strange and difficult. Thanks for your smile.

However, my most important pillar today is, without any doubt, my family. My parents, my sisters and brothers-in-law, my aunts and uncles, my cousins, my dog Bartolo, and of course my nephews and nieces which are always ready to ask for something or just annoy. I must thank them for the life they have given me, the possibility to grow up among them what I would never change for anything, despite of the times we get mad at each other. All of them have brought me up, they have built my personality, so it is fair to say that they define the man I am today. Thank you for my life. I love you all very much.

Finally, acknowledge the help of the projects DPI2010-20542 and DPI2007-66773-C02-01 from the Ministerio de Economía y Competitividad and also the financial support of the FPU program (AP2008-01086).

Because each of you are a piece of me, thank you very much.

Contents

Abstract	I
Resumen	III
Resum	V
Acknowledgments	VII
Contents	XV
List of Figures	XXIV
List of Tables	XXVII
1 Introduction and scope	1
2 On the use of Cartesian grids for the FEM	9
2.1 Introduction and review. Motivation	9
2.2 Problem Statement	13
2.2.1 Finite element discretization	15
2.3 Analysis Mesh	15
2.3.1 Virtual numbering	17
2.3.2 Structured numbering	23
2.3.3 Sequential numbering	23
2.3.4 Numbering conversion	25
2.3.5 From virtual space to real space. Nodal coordinates . .	30
2.3.6 Generation of the Analysis Mesh	32

2.3.7	Geometry-mesh intersection. Integration	34
2.3.8	Element data sharing	47
2.4	h -adaptive refinement strategies	50
2.4.1	Geometrical refinement	52
2.4.2	Solution-based refinement	52
2.5	Essential boundary conditions	56
2.6	Solver	58
2.6.1	Nested Domain Decomposition-based direct solver . . .	59
2.6.2	Iterative solver. Projections	64
2.7	Numerical Results	65
2.7.1	Evaluation of the global system of equation	67
2.7.2	NDD. Computational cost	69
2.7.3	Iterative Solvers. Projections	72
2.8	Conclusions	76
3	Error estimation and recovery procedures	79
3.1	Introduction and general review	82
3.2	Error estimators in FEM	86
3.2.1	Explicit residual error estimator	86
3.2.2	Implicit residual error estimator	91
3.2.3	Constitutive Relation Equation (CRE) based error estimators	95
3.2.4	Mixed formulation error estimators	96
3.2.5	Recovery-based error estimators. The SPR technique . .	96
3.2.6	A nearly equilibrated recovery procedure. The SPR-C technique	99
3.3	Nearly equilibrated displacement recovery procedure. The SPR-CD technique	104
3.3.1	Constraints definition	107
3.3.2	Patches cut by the boundary	108
3.3.3	Recovered stress evaluation	109
3.3.4	Singular fields	110
3.3.5	Efficiency of the recovery procedure	116
3.4	Error estimation in energy norm. Error bounding	118

3.4.1	Error estimation in energy norm	118
3.4.2	Error bounds in energy norm	119
3.5	Numerical results	124
3.5.1	Problems with exact solution	126
3.5.2	Problems without analytical solution	141
3.6	Conclusions	144
4	Goal Oriented Adaptivity	147
4.1	Introduction and motivation	147
4.2	Auxiliary problem statement. The dual problem	149
4.3	Analytical definitions of the dual problem for equilibrium en- forcement	151
4.3.1	Mean displacement in Ω_I	152
4.3.2	Mean displacement along Γ_I	153
4.3.3	Mean strain in Ω_I	153
4.3.4	Mean stress value in Ω_I	154
4.3.5	Mean tractions along Γ_I included in Γ_D	154
4.3.6	Generalized stress intensity factor	155
4.4	Error bounds in QoI	157
4.4.1	Definitions	157
4.4.2	The upper bound for GOA	159
4.4.3	The lower bound for GOA	160
4.4.4	Numerical bounding approaches for GOA	161
4.5	Numerical results	164
4.5.1	Problem 2: Thick-wall cylinder subjected to internal pressure	165
4.5.2	Problem 3: L-Shape plate	174
4.6	Conclusions	179
5	Upper error bounding techniques for the error in the energy norm	181
5.1	A fully equilibrated recovery procedure. The FER technique . .	182
5.1.1	Recovery procedure	183
5.1.2	Comments about the resolution of the system of equations	188

5.2	An error estimator combining a recovery-based procedure with equilibrated local problem resolution. The RL technique	190
5.2.1	Local problem formulation	191
5.2.2	The recovery procedure for the auxiliary stress field . .	193
5.3	A fast SPR-based upper bounding technique. The FUB technique	197
5.3.1	Numerical evaluation of the constant C_Ω	201
5.4	Numerical results	203
5.4.1	Problem 1a: 3^{rd} order solution in a 2×2 square	203
5.4.2	Problem 1c. 2×2 square without body forces	204
5.4.3	Problem 1d. 2×2 square. Thick-wall cylinder subjected to internal pressure	207
5.5	Conclusions	208
6	Error estimation in the recovered solution field	211
6.1	Introduction	211
6.2	Error norm representation for the recovered solution	213
6.3	h -adaptive refinement process	216
6.4	Numerical Examples	218
6.4.1	Accuracy of the error estimator for the recovered solution	218
6.4.2	h -adaptive process	224
6.5	Conclusions	227
7	Structural Shape Optimization	229
7.1	Introduction and motivation	229
7.2	The optimization problem	231
7.3	Numerical Results	232
7.3.1	Optimization problem 1. Cross section under pressure over internal circular shape	232
7.3.2	Optimization problem 2. Gravity dam	237
7.4	Conclusions	241
8	Conclusions	243
8.1	Future works	246
	Bibliography	247

A Displacement-based Moving Least Squares recovery with constrains. The MLS-CD technique	263
A.1 Introduction	263
A.2 The MLS-CD recovery technique	264
A.2.1 MLS recovery process	264
A.2.2 Satisfaction of the Dirichlet equation	266
A.2.3 Satisfaction of the equilibrium equation	268
A.2.4 Satisfaction of the internal equilibrium equation	272
A.2.5 Visibility	275
A.2.6 Stress splitting for singular problems	276
A.3 Concluding remarks	276

List of Figures

1.1	2D representation of the Alamillo bridge in Sevilla.	4
2.1	Reference system for the virtual numbering depending on the mesh level. In a) a 3D quadratic element is represented in red and a 3D linear element in blue. Note that the smallest quadratic element that can be represented for a given node discretization is bigger than the linear element because of the mid-side nodes.	18
2.2	2D element characterization in the virtual mesher. The element side numbering, the local node numbering and the local children numbering is indicated.	20
2.3	Children position for the 3D elements	21
2.4	Local number of the element sides and edges used to locate the neighbors of the same level elements in 3D for faces and edges.	21
2.5	Topology for the 3D elements, vertex nodes for linear elements in red and mid-side nodes for quadratic elements in blue.	22
2.6	Coordinates of the neighbors of the same level elements in 2D.	22
2.7	Oct.tree mesh structure numbered with the structured structured numbering for both, nodes and elements.	24
2.8	Set of selected elements conforming an analysis mesh. The sequential numbering is written into each element. The lower level elements are numbered first.	25
2.9	Element and node numbering transformation between the VM and the FE code.	26
2.10	Set of selected elements conforming an analysis mesh. The structured numbering is written into each element.	29

2.11	Transformation process from the virtual coordinate system to the coordinate system in the real space.	31
2.12	Difference between the Cartesian grids pile and the analysis mesh.	33
2.13	Element types according to their relation with the problem boundary: Internal elements, external elements and boundary elements.	35
2.14	Intersection and subdomain generation process in boundary elements.	37
2.15	Integration domains for different integrations schemes.	39
2.16	Integration area for linear and quadratic integrations schemes.	41
2.17	Triangular elements for the transfinite mapping in local and global reference systems.	42
2.18	Integration area for transfinite mapping.	46
2.19	<i>Vertical data sharing</i> 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 <i>reference element</i> . The calculations for boundary element colored in blue are reused. Element matrices in the finest mesh are only evaluated for white elements.	49
2.20	Comparison of two different geometries $i < j$ during an optimization process. The data for the green elements evaluated for geometry i are reused in geometry j	51
2.21	Intersection points and distances used for the evaluation of the curvature error indicator k	54
2.22	Shape function for intersection point i	58
2.23	Nested scheme 1.	60
2.24	Nested scheme 2.	61
2.25	Nested scheme K.	63
2.26	Representation of the nested arrowhead structure.	63
2.27	Problem 2, Q4u. Computational cost, time in seconds, used for meshing tasks (left) and solving tasks (right). The results are obtained for the commercial code ANSYS [®] 12.1 and for the proposed cgFEM code with the NDD technique.	68

2.28	Problem 2, Q4u. Computational cost, time in seconds, used for whole FEM analysis, considering the solution recovery process and the error estimation for a commercial code ANSYS [®] 12.1 and for the proposed code cgFEM with the NDD technique. . .	70
2.29	Problem 2, Q4u. Computational cost (time in seconds) used to solve the global system (2.35) with different reordering procedures. Reordering time also included.	71
2.30	Projection of the solution of the analysis mesh $i - 1$ into the analysis mesh i	73
2.31	Problem 2, Q4u. Number of solving iterations needed to project the initial guess from the coarser mesh ($i - 1$) to the finer one (i) and the number of iterations required to solve the system of equations.	74
2.32	Problem 2, Q4u. Number of iterations needed using as initial solution the null vector, the FEM solution and the recovered solution. All cases consider the projection of the Lagrange multipliers.	75
3.1	Scheme for the error sources during the simulation of a physical phenomena.	80
3.2	Different discretizations of a quarter of a square domain with a hole. Left, coarse discretization, right, finer element mesh. . . .	81
3.3	Local problem at element $K \in \mathcal{T}$. The traction \mathbf{t}_L represent the equilibrated traction obtained from a FE post-processing technique.	92
3.4	Local problem at element $K \in \mathcal{T}$. The tractions \mathbf{t}_L represents the equilibrated tractions obtained from a FE post-processing technique.	94
3.5	Representation of a patch of linear triangular elements. The black points indicates the nodes of the mesh and the red node is the patch assembly node. The transparent surfaces indicate the FE stress field $\boldsymbol{\sigma}^h$. The super-convergent points are indicated by blue crosses.	98

3.6	Representation of the least squares fitted polynomial surface. The pink line represents the stress value at the assembly node, the only one that is retained in the standard SPR.	99
3.7	Representation of the final recovered stress field σ^* over the problem domain. The nodal recovered values are interpolated by the FE shape functions.	100
3.8	Example of an internal patch split by an internal curve. The black circle indicates the position of the node shared between the elements of the patch. The red points indicate the position where boundary equilibrium is imposed and the green points indicate some random points where the internal equilibrium constraints are imposed.	105
3.9	Elastic solid with a V-notch	112
3.10	Possible configurations of internal patches in 2D.	117
3.11	Problem 1a. Problem model and analytical solution.	126
3.12	Problem 1b. Problem model and analytical solution.	127
3.13	Problem 2. Thick-wall cylinder subjected to internal pressure.	128
3.14	Problem 3. L-shaped domain.	129
3.15	Convergence analysis for h -uniform refinement.	131
3.16	Q4. First four meshes of the h -adaptive refinement process.	132
3.17	Q8. First four meshes of the h -adaptive refinement process.	133
3.18	Effectivity of the error estimators for h -adaptive refinement.	136
3.19	Local effectivity indicators for h -adaptive refinement.	138
3.20	Q4. SPR-CD. Local effectivity indicator D for the h -adaptive refinement process.	139
3.21	Q8. SPR-CD. Local effectivity indicator D for the h -adaptive refinement process.	140
3.22	Problem 4. Gravity dam: Model of the problem, loads and constraints.	141
3.23	Problem 5. Flywheel: Model of the problem, loads and constraints.	142
3.24	Q4. Problems without exact solution. First four meshes of the h -adaptive refinement process.	143
3.25	Q8. Problems without exact solution. First four meshes of the h -adaptive refinement process.	143

3.26	Convergence analysis for h -adaptive refinement.	145
4.1	Problem 2. Thick-wall cylinder subjected to an internal pressure. The domains of interest Ω_I and Γ_o are indicated in yellow.	166
4.2	Problem 2.a. \bar{u}_n along Γ_o . Q4h. Sequence of h -adaptive refined meshes.	167
4.3	Problem 2.a. \bar{u}_n along Γ_o . Q4h. $\lambda = 1$. Evolution of the effectivity index of the error estimation $\tilde{\vartheta}$ and the effectivity in the QoI $\tilde{\theta}^{QoI}$ for the error estimates in (4.7) and the error bounds (4.50) and (4.51) obtained with the SPR-CD technique.	169
4.4	Problem 2.b. \bar{u}_x in Ω_I . Q4h. Sequence of h -adaptive refined meshes.	169
4.5	Problem 2.b. \bar{u}_x in Ω_I . Q4h. $\lambda = 1$. Evolution of the effectivity index of the error estimation $\tilde{\vartheta}$ and the effectivity in the QoI $\tilde{\theta}^{QoI}$ for the error estimates in (4.7) and the error bounds (4.50) and (4.51) obtained with the SPR-CD technique.	171
4.6	Problem 2.c. $\bar{\sigma}_x$ in Ω_I . Q4h. Sequence of h -adaptive refined meshes.	172
4.7	Problem 2.c. $\bar{\sigma}_x$ in Ω_I . Q4h. $\lambda = 1$. Evolution of the effectivity index of the error estimation $\tilde{\vartheta}$ and the effectivity in the QoI $\tilde{\theta}^{QoI}$ for the error estimates in (4.7) and the error bounds (4.50) and (4.51) obtained with the SPR-CD technique.	173
4.8	Problem 3.a. K_I . Q4h. Cartesian meshes with h -adaptive refinement.	174
4.9	Problem 3.a. K_I . Q4h. Optimal value of λ and $\lambda = 1$. Evolution of the effectivity index $\tilde{\vartheta}$ and the effectivity in the QoI $\tilde{\theta}^{QoI}$ for the error estimates in (4.7) and the error bounds (4.50) and (4.51) obtained with the SPR-CD technique.	177
4.10	Problem 3.b. K_{II} . Q4h. Cartesian meshes with h -adaptive refinement.	178
4.11	Problem 3.b. K_{II} . Q4h. $\lambda = 1$. Evolution of the effectivity index $\tilde{\vartheta}$ and the effectivity in the QoI $\tilde{\theta}^{QoI}$ for the error estimates in (4.7) and the error bounds (4.50) and (4.51) obtained with the SPR-CD technique.	180

5.1	Internal patch formed by 4 elements (left) and patch in contact with the boundary formed by 2 elements (right).	184
5.2	Local SdR and patch. The SdR is used to define the rotation virtual displacement	194
5.3	Problem 1a. Q8 uniform refinement. Global effectivity index θ and overall computational cost.	205
5.4	Problem 1c. Model, material and analytical solution.	205
5.5	Problem 1c. Q8 uniform refinement. Global effectivity index θ and overall computational cost.	206
5.6	Problem 1c. Q8 uniform refinement. Global effectivity index θ for the FUB _{SPR-CD} with different values of the constant C . Note that for $C = 0.1$ for the finer mesh the effectivity is $\theta = 0.99595488728704$	207
5.7	Problem 1d: The model of the problem corresponds to the green square area extracted from the thick-wall cylinder geometry which corresponds to the geometry of the problem 2.	208
5.8	Problem 1d. Q8 uniform refinement. Global effectivity index θ and overall computational cost.	209
6.1	h -adaptive refinement scheme.	213
6.2	Problem 1b. Global effectivity index during the h -adaptive refinement process.	220
6.3	Problem 1b. Q4. Local exact error of the recovered solution (left), local error estimates using \mathcal{E}_3^* estimator (right).	221
6.4	Problem 2. Global effectivity index during the h -adaptive refinement process.	222
6.5	Problem 2. Q4. Local exact error of the recovered solution (left), local error estimates using \mathcal{E}_3^* estimator (right).	223
6.6	Problem 3. Global effectivity index during the h -adaptive refinement process.	224
6.7	Problem 3. Q4. Local exact error of the recovered solution (left), local error estimates using \mathcal{E}_3^* estimator (right).	225
6.8	Problem 2. h -adaptive analysis with Q4 elements. The black line represents the prescribed relative error in energy norm (1%).	226

6.9	Problem 2. h -adaptive analysis with Q8 elements. The black line represents the prescribed relative error in energy norm (0.05%).	226
7.1	Optimization problem 1. Model and data. The optimal shape corresponds to a thick-wall cylinder under internal pressure as represented in gray.	233
7.2	Optimization problem 1. Q4. Evolution of cylinder optimization considering the error estimation of the FE solution (σ^h) or the recovered one (σ_σ^*), for different prescribed errors levels γ . <i>Left</i> : Evolution of exact error in area with respect to the optimal analytical solution. <i>Right</i> : Real error in energy norm.	235
7.3	Optimization problem 1. Q4. Evolution of the optimization process considering the error estimation of the FE solution (σ^h) or the recovered one (σ_σ^*), for different prescribed errors levels γ . Evolution of the accumulative computational cost.	235
7.4	Optimization problem 1. Q8. Evolution of cylinder optimization considering the cgFEM as a solver (red line) or ANSYS® 12.1 (blue line) for $\gamma \leq 1$. <i>Left</i> : Evolution of exact error in area with respect to the optimal analytical solution. <i>Right</i> : Real error in energy norm.	236
7.5	Optimization problem 1. Q8. Evolution of the optimization process considering the cgFEM as a solver (red line) or ANSYS® 12.1 (blue line) for prescribed relative error $\gamma \leq 1$. Evolution of the accumulative computational cost.	237
7.6	Optimization problem 2. Gravity dam. Model of the problem.	238
7.8	Optimization problem 2. Gravity dam. Detail of the parametric hole and material properties.	239
7.9	Optimization problem 2. Gravity dam. Q4. Evolution of the dam optimization considering the cgFEM as a solver. Using <i>strategy a</i> (red line) or <i>strategy b</i> (blue line) for γ variable. <i>Left</i> : Evolution of the area. <i>Right</i> : Evolution of the estimated error in energy norm.	239

7.10 Optimization problem 2. Q4. Evolution of the dam optimization considering the cgFEM as a solver. Using *strategy a* (red line) or *strategy b* (blue line) for γ variable. Evolution of the accumulative computational cost. 240

7.11 Optimization problem 2. Q8. Evolution of optimization process considering the cgFEM with *strategy b* as a solver (red line) or ANSYS® 12.1 (blue line) for variable prescribed error. *Left*: Evolution of the total area. *Right*: Estimated error in energy norm. 241

7.12 Optimization problem 2. Q8. Evolution of optimization process considering the cgFEM *strategy b* as a solver (red line) or ANSYS® 12.1 (blue line) for variable prescribed error. Evolution of the accumulative computational cost. 242

A.1 MLS support with boundary conditions applied on the nearest boundary points. 267

A.2 Satisfaction of boundary conditions. We observe two zones corresponding to the support of points A and B , Ω_A and Ω_B , with B closer to the boundary. The red line indicates the exact value at the boundary imposed (displacement or traction). We observe that when the support point gets closer to the boundary ($A \rightarrow B$), the surface (shadowed red area) represents more accurately the exact value at the boundary. When the support point is on the boundary the recovered field (stress or displacement) will represent the boundary conditions exactly. 268

A.3 Domain with re-entrant corner. 276

List of Tables

2.1	Indexing vector from sequential numbering to structured numbering for the analytical mesh represented in Figure 2.10. The first row represents the position in the indexing vector (the number of the element in the sequential numbering system) and the second row is the value (the number of the element in the structured numbering system) allocated in the corresponding position.	29
2.2	Indexing vector from structured to sequential numbering systems for the analytical mesh represented in Figure 2.10. In the first row we represent the position in the indexing vector (the number of the element in the structured numbering system) and in the second row the value (the number of the element in the sequential numbering system) that is allocated.	30
2.3	Comparison between the preliminary mesh and the geometrically adapted mesh using the curvature criterion.	53
2.4	Second and third analysis mesh obtained using the error estimation information following the 1 st meshes represented in Table 2.3	55
2.5	Problem 2, Q4u. Number of iterations needed for different initial vectors. All cases consider the projection of the Lagrange multipliers. A maximum of 10^4 iterations is allowed. The tolerance for the stopping criterion in the iterative process is indicated on each table.	78
3.1	Q4h. Values for the global effectivity θ considering h -adapted meshes.	135

3.2	Q8h. Values for the global effectivity θ considering h -adapted meshes.	135
4.1	Problem 2.a. \bar{u}_n along Γ_o . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimator $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ for the GOA.	167
4.2	Problem 2.a. \bar{u}_n along Γ_o . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ of the bounding techniques for the GOA.	168
4.3	Problem 2.b. \bar{u}_x in Ω_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimator $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ for the GOA.	170
4.4	Problem 2.b. \bar{u}_x in Ω_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ of the bounding techniques for the GOA.	170
4.5	Problem 2.c. $\bar{\sigma}_x$ in Ω_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimator $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ for the GOA.	172
4.6	Problem 2.c. $\bar{\sigma}_x$ in Ω_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ of the bounding techniques for the GOA.	173
4.7	Problem 3.a. K_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimator $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ for the GOA.	175
4.8	Problem 3.a. K_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ of the bounding techniques for the GOA.	175
4.9	Problem 3.a. K_I . Q4h. Optimal value of λ . Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ of the bounding techniques for the GOA.	176
4.10	Problem 3.b. K_{II} . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimator $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ for the GOA.	178

4.11	Problem 3.b. K_{II} . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ of the bounding techniques for the GOA.	179
5.1	Number of coefficients and constraints	189
5.2	Problem 1a. Q8 uniform refinement. Values of the global effectivity index θ for different bounding techniques and C^* for the FUB technique.	204
5.3	Problem 1c. Q8 uniform refinement. Values of the global effectivity index θ for different bounding techniques and C^* for the FUB technique.	206
5.4	Problem 1d. Q8 uniform refinement. Values of the global effectivity index θ for different bounding techniques and C^* for the FUB technique.	209
7.1	Optimization problem 1. Design variables constraints and reference solution.	234
7.2	Optimization problem 2. Gravity dam. Design variables constraints and reference solution.	240

Chapter 1

Introduction and scope

Since the early days of the Computer Aided Engineering (CAE) the Finite Element Method (FEM) has predominated over other methods of analysis and simulation of structural components. Nowadays, FEM is the most used technique, not only for the linear elasticity problem in structural analysis, but also for plasticity, electromagnetism, heat transfer, etc. The FEM is a very flexible technique that can be applied in a vast amount of engineering applications.

CAE software has been closely related with Computer Aided Design (CAD) software. Traditionally, when a part of a structure or a mechanism is being designed, first the geometry is defined with a CAD system and finally it is analyzed with CAE software in order to control its behavior under certain load situations. In case, after the simulation with the CAE software, the component does not behave as desired the user has to modify the geometry of the component to try to obtain a suitable geometry. In the majority of situations a manual or poorly automatized process is used to obtain the modified geometry. This has been the design process of components of structures for the last 50 years. The problem of this process is that it is a trial and error

procedure. The final result of this trial and error procedure strongly depends on the designer experience, preventing, in general, the evaluation of optimal designs. Another important drawback is that it requires an incredible amount of man-hours to obtain the final optimized geometry since the user has to keep on checking the results and modifying the geometry until what he/she understands as the final or, at least suitable, geometry is obtained. In a highly evolving industrial environment this process is not desired because it slows down the design process.

Nowadays high technology industries, such as the automotive industry or the aerospace industry among others, need more efficient design processes able to provide optimal solutions (and not only suitable ones) within a reasonable time, where the human intervention is restricted to the initial steps of the optimization process and non-existent during the iterative process. In order to achieve this objective, we need to couple the optimization processes with the design of components. This consists in parameterizing the geometry of the component and running an optimization process over these parameters. Theoretically, the result will be the combination of the parameters that provides the optimal geometry for the application under a set of prescribed constraints. The structural optimization processes consist of two levels, the *higher level* and *lower level*. The *higher level*, the optimization algorithm, provides the combinations of parameters that define the geometries whereas the *lower level* is in charge of numerically analyzing each of them in order to evaluate their response. This method, depending on the behavior of the higher level, could in general lead to the optimal solution, thus providing the industry the necessary tool to automatically obtain the optimal configurations for their designs. However, there are several practical problems that are not already solved. For instance, a great amount of computationally expensive analysis is required to obtain the optimal geometry of a component, making this process prohibitive for practical applications. Additionally, when using the traditional FEM as lower level, a robust and efficient meshing method for very complex geometries is required. This goal is not yet fully achieved in commercial codes. As a result of this the user needs to check each mesh before running the analysis. Finally, in order to guarantee that the optimization process converges to a suitable

geometry, we need to control the error in each analysis performed by the lower level [1, 2, 4], otherwise the "noise" introduced by the numerical errors will make the optimization process to converge to a non-optimal solution, decrease its convergence rate or even prevent convergence. This issue has not been adequately addressed in commercial codes preventing their use in optimization processes.

In this work we want to deal with and improve these three issues: efficiency of the optimization process, robustness of the FE code used in the lower level and accuracy of the FE analysis. We will present the Cartesian Grid Finite Element Method, cgFEM technique, which represents an appropriate combination of techniques to reach our objective. The main characteristic of cgFEM is that the traditional geometry-conforming mesh of the FEM method disappears. We will make the mesh used to solve the FE problem independent of the geometry of the component to be analyzed. In the cgFEM framework we have two domains, the problem domain Ω and the meshing domain Ω_E that is a square, trivial to mesh, surrounding Ω . This process avoids the tedious meshing process of the traditional FEM. Similar techniques can be found in the literature, being all of them classified under the umbrella term of Finite Elements in ambient space [5]. The results will show how under this meshing framework very complex geometries, such as the bridge represented in Figure 1.1 where the detail ratio is 1500 : 1 (relation between the problem size and the smallest detail), could be successfully represented, with good quality elements, without any user manipulation.

Once the robustness issue has been addressed, the next issue is efficiency. The efficiency in comparison with traditional FE codes can be gained in two different fronts: the generation of the FE numerical model and its resolution. The first one is building the FE numerical problem and the second one is solving it. In the cgFEM framework, to build the numerical model, we rely on the mesh structure and on the hierarchical data structure specifically implemented for this type of meshes to reduce the amount of calculations to be performed. In general, in the traditional FEM, each element of the mesh is different, having different stiffness matrix per element and integration points. Then they

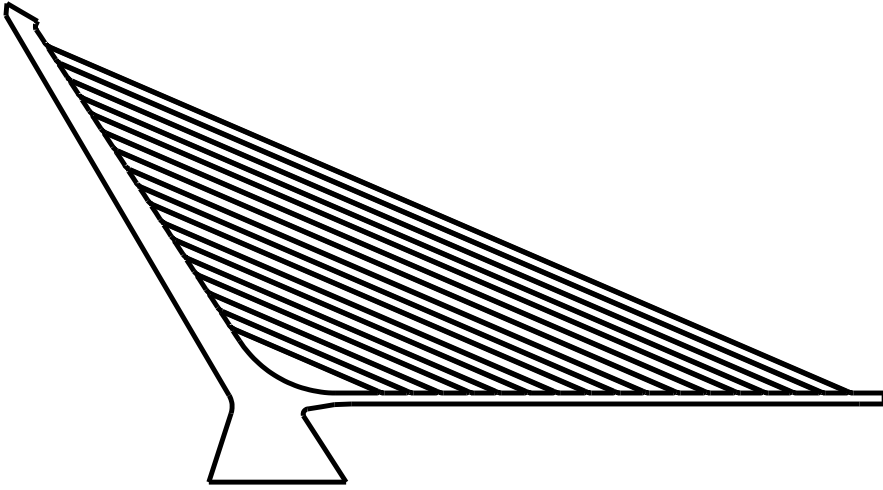


Figure 1.1: 2D representation of the Alamillo bridge in Sevilla.

have to be specifically evaluated at each element of the mesh. As it will be explained in detail in this work, in the cgFEM the mesh is formed by geometrically similar elements. The geometrically similar elements are those that have the same shape but different sizes. The integration properties and specifically the stiffness matrix of these geometrically similar elements are related with the corresponding scaling factor, then when only one is evaluated, automatically all of them can be directly evaluated. For instance, when the stiffness matrix is evaluated for one element, this matrix will be shared with the other elements, thus avoiding repeated calculations. However, there is a price we have to pay. There are some elements of the Cartesian grid that can not be treated in this way. The elements that are cut by the boundary of the problem domain $\partial\Omega$ need a special and individualized treatment. This process avoids numerous unnecessary repeated calculations improving the performance of the method in comparison with the traditional FEM.

Additionally and specifically when the cgFEM is used as the lower level of a shape optimization process, since the problem domain Ω and the meshing domain Ω_E are independent, we can choose the same meshing domain for all individuals during the optimization process. As the meshing domain

Ω_E remains the same for all individuals, the sharing information procedures could also be applied between the different geometries during the optimization process. The numerical experiments indicate that this approach considerably reduces the computational cost [6] of the whole optimization process.

Regarding the resolution of the global system of equations, we can differentiate between two cases: direct solvers and iterative solvers. When direct solvers are used, it is very interesting to have an adequate reordering of the matrix of the system of equations. In this work we have used a reordering based on the structure of the Cartesian grid, the Nested Domain Decomposition (NDD). This provides an optimal reordering for the matrix since it directly uses the topological information of the mesh. When iterative solvers are used, because of the Cartesian grid structure, efficient projection techniques have been easily implemented to project the solution of the previous mesh as the initial solution for the current mesh.

The third issue we wanted to deal with regarding to the use of FE codes in shape optimization processes was the error estimation and the h -adaptive refinement schemes. Only a few commercial codes include error estimation techniques and none of them are robust enough to work as a lower level of a shape optimization process with satisfactory results. In this work we propose a very robust solution for this problem that perfectly fits not only in the cgFEM framework but also in commercial codes, the use of a recovery type estimator of the error of the FE solution [3] based on the use of an improved version of the the Superconvergent Patch Recovery Technique (SPR) [7, 8] developed in this thesis and called SPR-CD technique. As indicated before, to guarantee that an optimization process converges to a suitable geometry we need to control the error of the numerical solution of each individual (geometry) analyzed during the optimization process. The SPR-CD technique is a displacement recovery process that provides a kinematically admissible improved displacement field and a nearly-statically admissible stress field of a high quality. This improved stress field will be used to evaluate the error in energy norm both, locally, to guide the h -adaptive refinement process, and globally. One of the main characteristics of the SPR-CD technique is that it

is extremely efficient when applied in the cgFEM framework, when compared with other similar techniques under the traditional FEM. In this work we have also extended the error estimation procedures to Goal Oriented Adaptivity in order to predict the error not only in energy norm but also in a quantity that could be of interest for practitioners, such as displacements or stresses. Moreover, in this work we also introduce some new alternatives to evaluate upper error bounds in both, energy norm and Quantities of Interest.

Another important aspect of this work is that we have introduced a new error estimation of the recovered stress field provided by the SPR-CD technique. Since we will be able to evaluate the error of the recovered solution we propose the use of the recovered solution as output of the analysis. The recovered solutions provided by the SPR-CD technique generally have higher accuracy than the raw FE solution provided by the cgFEM. In practice this means that we can solve the problem with the same accuracy in coarser meshes, considerably diminishing the computational cost of the analysis to reach the prescribed accuracy level. This is crucial in optimization processes because it reduces the total computational cost of the process which is one of the more important drawbacks.

The cgFEM framework successfully deals with the three main problems of the traditional FEM when applied as a lower level of a structural shape optimization process. Techniques similar to the cgFEM have been already implemented and presented in the bibliography. In academia, we can highlight the technique introduced in [9] which does not include error estimation procedures. Also there is a commercial code based on Cartesian grids for 3D scanned images under the name Scan&SolveTM developed by the US company In-tact Solutions, LLC. However, so far it does not include error estimation nor h -adaptive techniques. The Cartesian grid-based FE codes are a challenging area of research because of their properties, specially in linear problems, and also because of their robustness.

This thesis will deal with different aspects of the FEM. In this introduction we have presented a general overview of this work to show the motivation of the different topics of this thesis. Further detailed introduction and state of the art of each topic will be presented in the corresponding Chapter. This work is distributed as follows: in Chapter 2 we introduce the cgFEM and all the details to build the global system of equations and to solve it. In Chapter 3 we present the error estimation techniques used in the cgFEM and a brief summary of traditional ones for comparison. In Chapter 4 we introduce the Goal Oriented Analysis in combination with the SPR-CD technique that is able to provide not only error measures in Quantities of Interest but also very accurate numerical bounds. In Chapter 5 we show three new techniques to obtain upper bounds of the error in energy norm. In Chapter 6 we present an heuristic error estimation for the recovered stress solution provided by the SPR-CD technique and finally in Chapter 7 we gather all improvements together applying the cgFEM to structural shape optimization problems comparing the results with a commercial code despite of the fact that the proposed techniques have been fully implemented in Matlab[®] 2010b. The last chapter will summarize the conclusions of this work.

Chapter 2

On the use of Cartesian grids for the Finite Element Method

2.1 Introduction and review. Motivation

During the last century researches have devoted big efforts to solve Boundary Value Problems (BVP) in a wide range of disciplines, such as elasticity, fluid-dynamics, acoustics, electromagnetism, etc. Generally these problems do not have any known analytical solution, then they can only be solved numerically. One technique to solve them, that emerged around mid XXth century, is the Finite Element Method (FEM), with the initial contributions of Alexander Hrennikoff (1941) and Richard Courant (1942). The main characteristic of these two preliminary approaches was to divide the problem domain into small regions bringing out the concept of *mesh* [10]. Along the 50s and 60s several authors such as Courant, Prager, Synge, Clough and Friedrichs among others

created the basis of the FEM. In the late 60s Zienkiewicz and Cheung published one of the first books on the FEM “*The finite element method in structural and continuum mechanics: Numerical solution of problems in structural and continuum mechanics*” (1967), bringing the corresponding mathematical formalism to the method.

The mesh has become one of the main characteristic of the FEM, and obviously, its creation, is the first step to solve any BVP using this method. Nowadays the computational resources have increased and also the complexity of the problems at hand. The difficulty to generate the mesh is directly related to the complexity and the details of the geometry (the problem domain Ω). The designers need to simplify the geometry to obtain an analysis-suitable geometry. According to [11] the time spent to generate an appropriate finite element mesh is excessive requiring a great amount of man-time. According to [12], recent studies at Sandia National Laboratories revealed that the process of creating an analysis-suitable geometry and the meshing of that geometry requires about 80% of overall analysis time, whereas only 20% of overall time is devoted to the analysis itself.

The traditional FEM is based on the idea of a domain Ω which is discretized into small regions, for instance triangles or squares in 2D. Therefore the shape and the distribution of those subdomains is directly related to the geometry of the problem domain Ω . One way to decrease this 80% of overall time prior to the analysis is to make the mesh geometry-independent, for instance, meshing an auxiliary domain Ω_E . This approach can considerably reduce the time devoted to prepare an analysis-suitable model and to mesh the domain, and is especially useful in applications that would require continuous remeshings during the analysis, like structural shape optimization problems, wear modeling, etc. Thus, a natural variant of the traditional FEM appeared in 60s, which aimed to avoid or reduce to a minimum the mesh generation burden. According to [13], a wide amount of techniques have been developed since 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). These methods are mainly based on separating, or

making independent, the problem domain from the Finite Element (FE) mesh discretization. This means dealing with two different domains, the first one is the discretization domain Ω_E , where the FE discretization is applied, and the second is the problem domain Ω . The only requirement is that the discretization domain must completely embed the problem domain, $\Omega \subset \Omega_E$.

Following these ideas some variations of the FEM were developed to improve its performance. In these approaches, the discretized domain is Ω_E instead of the problem domain. Generally Ω_E is a domain with a simple geometry that can be easily meshed. The analysis methodology presented in this work is based on this idea. In our implementation for 2D problems, Ω_E is a square whose discretization into square quadrilateral elements of uniform size is trivial. These techniques have been used both, in the Finite Volume Method (FVM) and in the FEM and been applied in various fields, such as acoustics [14, 15, 16], fluid dynamics and fluid-structure interaction [17, 18], tank waves modeling [19], biomedical problems [20], convection-diffusion [21] and optimization [22, 23, 24, 25]. The present work will only focus in the FEM framework for 2D linear elastic problems.

These techniques have several names in literature, such as *Fictitious Domain* [14, 22, 21, 13, 26], *Implicit Meshing* [11], *Immersed FEM* [27], *Immersed Boundary Method* [17, 28], *Fixed Grid FEM* [29, 9], etc. They have been described in [5] under the name *Finite elements in ambient space*.

Two more techniques based on these ideas and the particularity of improving the solution by adding known information have appeared in the late 90's. These two improvements of the original FEM are the eXtended Finite Element Method (XFEM) [30, 31] developed by T. Belytschko and his group at Northwestern University (USA), and the Generalized Finite Element Method (GFEM) [32, 33] developed by I. Babuška and coworkers 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) [34] and the Level Set Method (LSM) [35] to introduce enrichment functions to represent the dis-

placement discontinuity between the crack faces, the singular fields around the crack tip and the geometrical description of the crack. 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 [32] follows a similar approach also based on 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 [32]. 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. The LSM has also been used in [36] to represent the geometry in non-conforming meshes. Other authors also combine the LSM for boundary representation with the XFEM to represent the solution gradient discontinuities into an element containing more than one material [37, 38]

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 work 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 between these methodologies and the standard FEM is the consideration of the Dirichlet boundary conditions. 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 the previous approaches were mainly interested on decoupling the geometry representation from the FE mesh where the solution is interpolated. Generally, in these techniques, the computational cost is concentrated in the elements lo-

cated along the boundary as, in the Boundary Element Method (BEM) where only the external boundary is considered for the analysis. In this framework, we can cite the work done by Simpson *et. al.* [39] and Scott *et. al.* [40] where the Isogeometric Analysis IGA [41], respectively with NURBS and T-Splines has been adapted to the BEM. Thus, the geometry and the BEM mesh are strongly coupled.

Through this Chapter we will expose the main characteristics of the cgFEM whose objective is to be used as a FE code suitable to efficiently solve the 2D linear elasticity problem.

2.2 Problem Statement

In this section we briefly present the model for the 2D linear elasticity problem. We denote the Cauchy stress as $\boldsymbol{\sigma}$, the displacement as \mathbf{u} , and the strain as $\boldsymbol{\varepsilon}$, all these fields being defined over the domain $\Omega \subset \mathbb{R}^2$, of boundary denoted by $\partial\Omega$. Prescribed tractions denoted by \mathbf{t} are imposed over the part Γ_N of the boundary, while displacements denoted by $\bar{\mathbf{u}}$ are prescribed over the complementary part Γ_D of the boundary. \mathbf{b} denotes the body load.

The elasticity problem takes the following form. We look for $(\boldsymbol{\sigma}, \mathbf{u})$ satisfying:

- statical admissibility:

$$\mathbf{L}^T \boldsymbol{\sigma} + \mathbf{b} = 0 \quad \text{in } \Omega \quad (2.1)$$

$$\mathbf{G}\boldsymbol{\sigma} = \mathbf{t} \quad \text{on } \Gamma_N \quad (2.2)$$

where \mathbf{L} is the differential operator,

$$\mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \quad (2.3)$$

and \mathbf{G} is the projection operator that projects the stress field into tractions over the boundary. The operator \mathbf{G} is the matrix form of Cauchy's law considering the unit normal $\mathbf{n} = \{n_x \ n_y\}^T$ to Γ_N such that:

$$\mathbf{G} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix} \quad (2.4)$$

- kinematic admissibility:

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_D \quad (2.5)$$

- constitutive relation:

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \mathbf{L}\mathbf{u} \quad \text{in } \Omega \quad (2.6)$$

$$\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\varepsilon}(\mathbf{u}) - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0 \quad \text{in } \Omega \quad (2.7)$$

where matrix \mathbf{D} contains the elasticity coefficients of the usual linear isotropic constitutive law relating stress and strain, and $\boldsymbol{\sigma}_0$ and $\boldsymbol{\varepsilon}_0$ are the initial stress and strain respectively.

Using the notations introduced in [42] the problem above takes the primal variational form:

$$\begin{aligned} &\text{Find } \mathbf{u} \in (V + \{\mathbf{w}\}) : \forall \mathbf{v} \in V \\ &a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{u})^T \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega = l(\mathbf{v}) = \\ &\int_{\Omega} \mathbf{b}^T \mathbf{v} d\Omega + \int_{\Gamma_N} \mathbf{t}^T \mathbf{v} d\Gamma + \int_{\Omega} \boldsymbol{\varepsilon}_0^T \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega - \int_{\Omega} \boldsymbol{\sigma}_0^T \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega \end{aligned} \quad (2.8)$$

where $V = \{\mathbf{v} \mid \mathbf{v} \in H^1(\Omega), \mathbf{v}|_{\Gamma_D} = \mathbf{0}\}^1$ and \mathbf{w} is a particular displacement field satisfying the Dirichlet boundary conditions.

¹ $H^1(\cdot)$ is the $W_2^1(\cdot)$ Sobolev space. This is a vector space equipped with a norm that is the combination of the L^2 -norms of the function itself and its first derivative. $H^1(\cdot) = W_2^1(\cdot) := \left\{ f \in L^2(\cdot) : \|f\|_{W_2^1(\cdot)} < \infty \right\}$, where $\|f\|_{W_2^1(\cdot)} := \sqrt{\sum_{|\alpha| \leq 1} \|D^\alpha f\|_{L^2(\cdot)}^2}$. $L^2(\cdot)$ is the Lebesgue space.

2.2.1 Finite element discretization

Let us introduce a classical finite element discretization scheme for the elasticity problem. The approximate displacement field \mathbf{u}^h is searched for in a space of finite dimension $(V^h + \{\mathbf{w}\}) \subset (V + \{\mathbf{w}\})$ such that V^h is spanned by locally supported finite element shape functions.

Using the Galerkin framework, the primal variational formulation (2.1-2.7) is recast in the form:

$$\begin{aligned} \text{Find } \mathbf{u}^h \in (V^h + \{\mathbf{w}\}) : \forall \mathbf{v} \in V^h \\ \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{u}^h)^T \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega = \\ \int_{\Omega} \mathbf{b}^T \mathbf{v} d\Omega + \int_{\Gamma_N} \mathbf{t}^T \mathbf{v} d\Gamma + \int_{\Omega} \boldsymbol{\varepsilon}_0^T \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega - \int_{\Omega} \boldsymbol{\sigma}_0^T \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega \end{aligned} \quad (2.9)$$

which can be solved using classical finite element technology [43]. Finally the FE stress solution is evaluated as follows:

$$\boldsymbol{\sigma}^h(\mathbf{x}) = \mathbf{D}(\boldsymbol{\varepsilon}(\mathbf{u}^h(\mathbf{x}))) - \boldsymbol{\varepsilon}_0(\mathbf{x}) + \boldsymbol{\sigma}_0(\mathbf{x}) \quad (2.10)$$

2.3 Analysis Mesh

In standard FEM implementations the mesh must conform to the outer 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 un-distorted 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 cgFEM the mesh does not need to conform to the geometry. As a result of this, the analysis mesh is easily created and the elements will remain un-distorted.

There are several approaches to decrease the meshing time in the related bibliography. They are braced under the oct-tree (quad-tree for 2D) framework. The mesh generator implemented in the cgFEM code is also based on the quad-tree approach. Meagher [44] presented an oct-tree procedure for geometry representation due to its advantages in computational cost and memory storage. Yerry and Shepherd [45] and Baehmann *et. al.* [46] presented a mesh generation based on an oct-tree. In this case, they built the oct-tree 3D mesh obtaining some cut elements along the boundaries. Then, all the elements in the mesh are spitted into triangles adding some extra nodes when necessary. Finally the nodal locations are modified to improve the element quality. Therefore the regular element shape is lost during the process. Another oct-tree implementation developed by Jackins and Tanimoto [47], similar to the one used in this work, is based on a Cartesian coordinate system to allocate the elements. Our purpose was to generate a virtual home-made quad-tree mesh generator in order to have total control over the mesh generation process and to be able to adapt it to our FE code requirements.

This Section will show how the final mesh used for the analysis will be created. A computationally efficient technique has been developed to create the h -adapted meshes that will be used in the FE analyses. We will make use of three different numbering structures:

- Virtual numbering
- Structured numbering
- Sequential numbering

These three numbering systems, their construction, and their relations will be described in Sections 2.3.1 to 2.3.3. Each of the numbering systems will be advantageous for certain processes during the analysis. To take advantage of all the properties of these systems we will create efficient numbering conversion schemes that will be described in Section 2.3.4.

2.3.1 Virtual numbering

The virtual numbering consists in an oct-tree (in the implementation this part is already prepared for 3D codes) numbering system that defines, by means of explicit formulas, the relation between the nodes and the elements. The virtual numbering is the kernel of the virtual mesher (VM) that is in charge of finding the relations between element and nodes, i.e., it is in charge of building the mesh. With the virtual numbering of elements or nodes it results extremely fast to evaluate element and nodal information, such as neighborhood, children, parents, etc. by means of explicit formulas, avoiding the use of recursive techniques, typical in standard FE codes. The virtual numbering defines the element numbering and the node numbering, both based on their position in a *virtual* coordinates system. Figure 2.1 shows several examples of meshes. The origin of the virtual coordinate system is located at the top-left corner. For the nodes, Figure 2.1a, the length of each one of the axis is $2^{mL} + 1$, where mL is the maximum level of the mesh. mL is a parameter defined by the user that is related with the size of the finest element of the mesh. It can be appreciated that the mL parameter indicates the number of division of each axis, thus defining the discretization space. A node defined by the virtual numbering is characterized by its three coordinates in the virtual reference system, $\check{i} = \{X, Y, Z\}^T$, were \check{i} is the node number in the virtual numbering.

The virtual numbering for the elements is slightly more complex. Figure 2.1 shows different element numberings for the same domain and for the same $mL = 3$, in this case. They only differ in the mesh level. For instance Figure 2.1b corresponds to level one, $L = 1$, that is 2^1 divisions per axis. In general 2^L divisions per axis are performed, see Figure 2.1c for $L = 2$ and Figure 2.1d for $L = 3$. Note that the higher level, the finer element. The virtual numbering for elements depends on the element level, then we need to add this to the corresponding virtual element numbering, $\check{K} = \{L, X, Y, Z\}^T$, where \check{K} is the element number in virtual numbering.

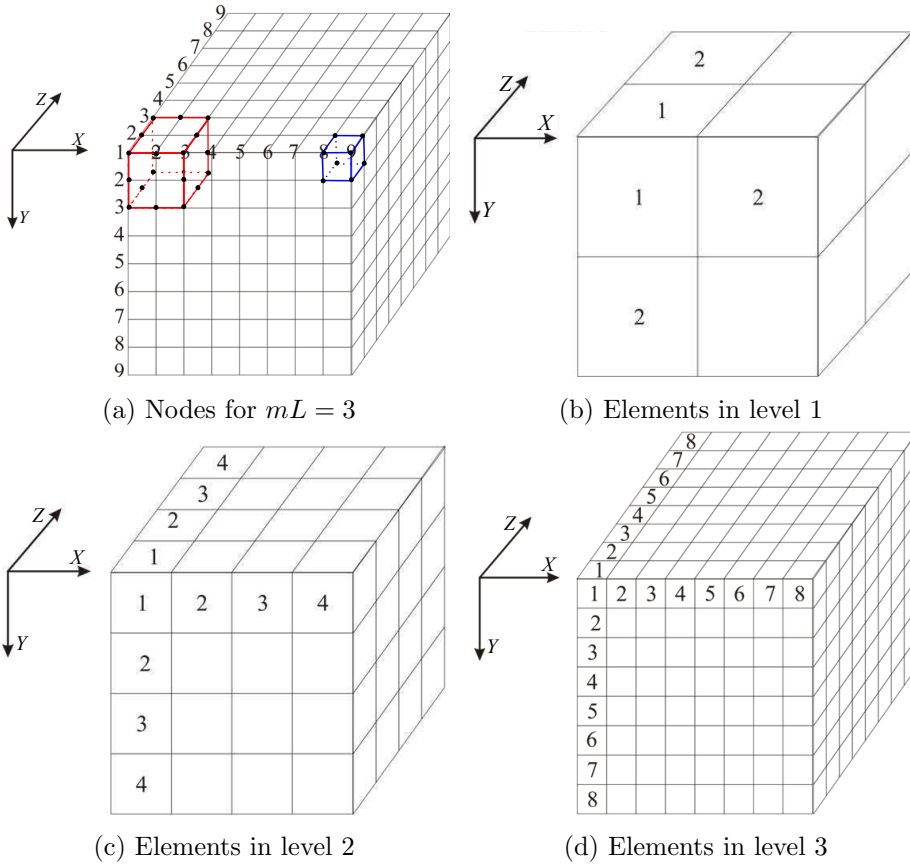


Figure 2.1: Reference system for the virtual numbering depending on the mesh level. In a) a 3D quadratic element is represented in red and a 3D linear element in blue. Note that the smallest quadratic element that can be represented for a given node discretization is bigger than the linear element because of the mid-side nodes.

The advantage of this numbering type both, for nodes and for elements, is that it permits to build a set of explicit formulas that, given the number of the element \tilde{K} or the node \tilde{i} , it can easily provide the data required for a FE analysis such as the element topology, its neighborhood, etc. Additionally, the searching element operation in which a point of the domain is contained results extremely fast as it only requires a coordinate system transformation from real coordinates to virtual coordinates. The following list describes all

data that is coded via explicit formulas that provide results in the virtual numbering system.

- *Children elements*: given an element in level L , its children are those in level $L + 1$ obtained by splitting the original element into new elements whose size (size of element's edge) is half of the original one. As the mesh refinement process will be based on element splitting, the information about the children elements will be used for the refinement. Children elements created during the refinement process will easily inherit properties from the parent element. The coordinates of the children elements, relative to the parent element coordinates, in the virtual numbering are shown in 2.2. Additionally the local element characterization, that is, the local element children numbering (1,2,3,4), the local sides numbering and the local nodes numbering are shown in Figure 2.2 for 2D and in Figures 2.3, 2.4 and 2.5 for 3D.
- *Parent element*: this is the element that embeds the children element as represented in Figure 2.2.
- *Element topology*: this indicates what nodes, in virtual numbering, are forming an element. The local position of the nodes is indicated in Figure 2.5 for 3D. For instance, in Figure 2.1b for the linear element $[1, 2, 1, 1]^T$ and $mL = 3$ the topology is:

$$\begin{bmatrix} 5 & 9 & 9 & 5 & 5 & 9 & 9 & 5 \\ 5 & 5 & 1 & 1 & 5 & 5 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \end{bmatrix}$$

- *Neighbors*: the neighbors of a given element are those of the same level (same size) sharing an edge in 2D, or a face or edge in 3D. There are 4 neighbors in 2D, one per side, and, for 3D 6 face neighbors and 12 edge neighbors. The element numbering of the neighbors in 2D is shown in Figure 2.6 and the local one in Figure 2.4 for 3D.

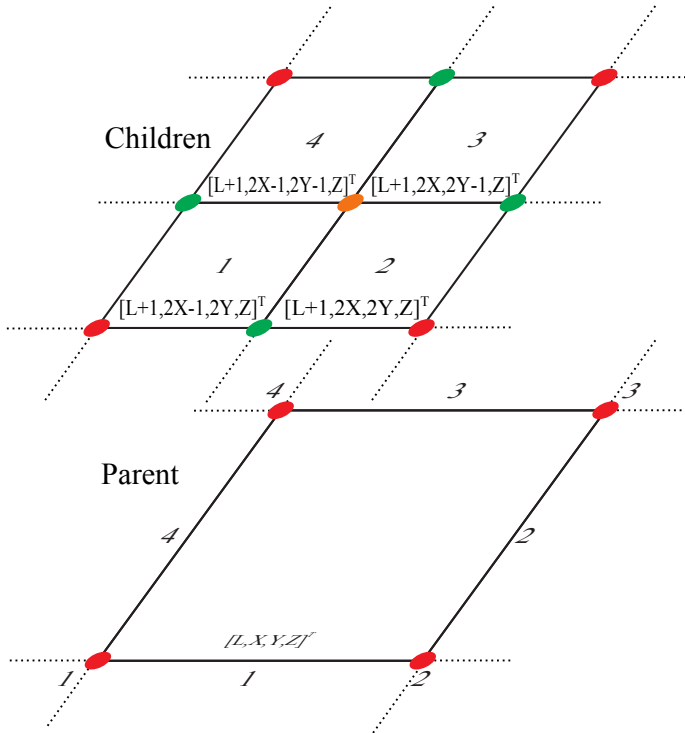


Figure 2.2: 2D element characterization in the virtual mesh. The element side numbering, the local node numbering and the local children numbering is indicated.

- *Element level*: this is directly extracted from the first component of a element virtual numbering. The level indicates the relative size of the element.
- *Node level*: this is the level of the element in which the node appears for the first time.
- *Node parents*: under a h -adaptive refinement process, one element could share an edge or a face with other of smaller size. The FE interpolation in that face or edge should be described by the element of higher size in order to enforce the continuity of the solution. The nodes defining the interpolation in that shared edge or face are the so-called *parent nodes*. Obviously this depends on the FE interpolation. For squared

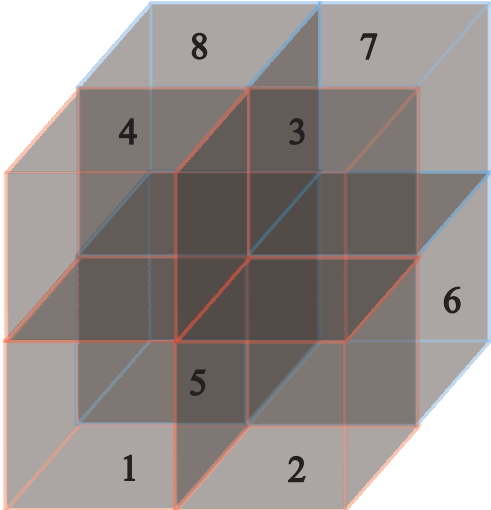


Figure 2.3: Children position for the 3D elements

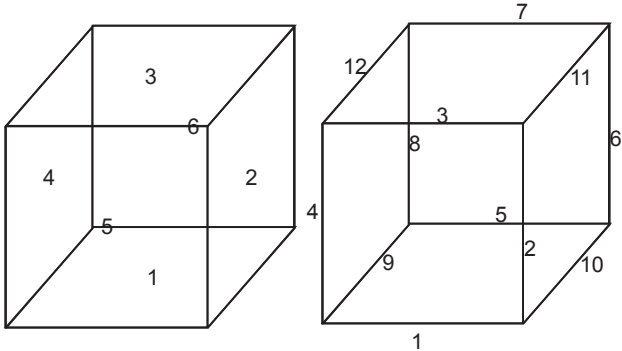


Figure 2.4: Local number of the element sides and edges used to locate the neighbors of the same level elements in 3D for faces and edges.

linear elements in Figure 2.2, green points are the children nodes and red point are the parents.

Note that the virtual mesher is able to provide 3D meshes, however, so far, the cgFEM code presented in this thesis is only able to deal with 2D prob-

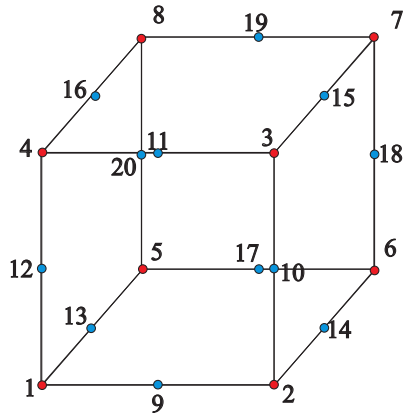


Figure 2.5: Topology for the 3D elements, vertex nodes for linear elements in red and mid-side nodes for quadratic elements in blue.

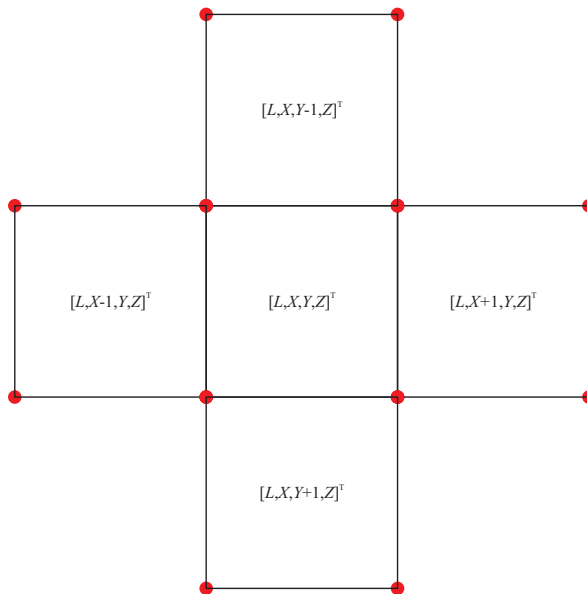


Figure 2.6: Coordinates of the neighbors of the same level elements in 2D.

lems. Hence the virtual mesher is only used for the 2D approach setting the coordinate $Z = 1$ for both, nodes and elements.

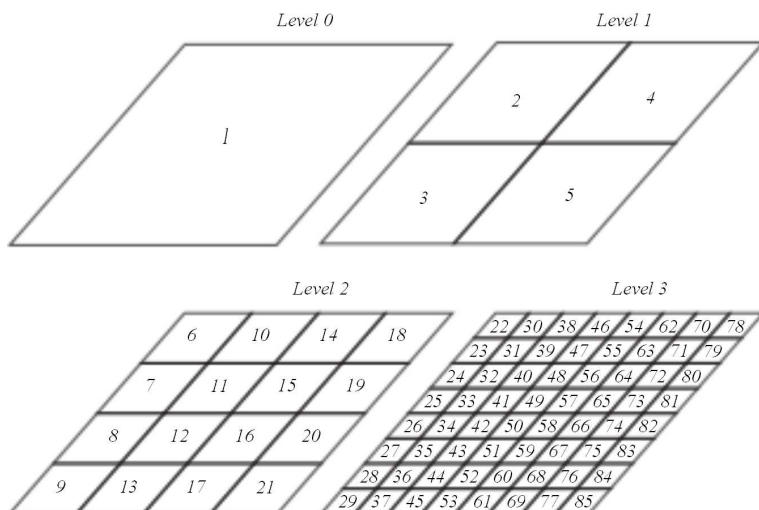
2.3.2 Structured numbering

The structured numbering condenses the virtual numbering into a single natural number. This avoids the use of a vector definition for each element or node. The structured numbering is living between the virtual mesher and the FEM code. The main purpose is to make easier the communication between the virtual mesher and the FEM code. The relation between the virtual numbering and the structured numbering depends on the dimensions of the problem (2D or 3D) and on the maximum level of the mesh mL . The structured numbering is a sequential numbering which maintains a rigid structure directly related with the oct-tree. The number of each element in structured numbering also indicates the position in the oct-tree, including the level of the element. That is, the element is completely defined.

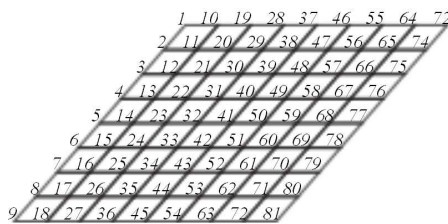
Figure 2.7 represents the scheme for the structured numbering of a given mesh. The maximum level of the mesh in the Figure is $mL = 3$. Figure 2.7a describes the element numbering in structured numbering. The characteristic feature is that independently of the level of the element the numbering is continuous even when we change from one level to the next level. The node numbers, see Figure 2.7b, only need to be identified in the finest mesh. They are numbered in a sequential manner from top-left to bottom-right as shown in the Figure.

2.3.3 Sequential numbering

The sequential numbering is only living into the FEM code itself. This numbering is related with the structured numeration via an indexing vector. In order to build the sequential numbering it is required to first build the analysis mesh. The analysis mesh consist of a set of selected elements from the quad-tree structure which will made up the mesh for the FE model. Further details of the analysis mesh construction will be presented in Section 2.3.6.



(a) Elements numbered with the structured numbering.



(b) Nodes numbered with the structured numbering for $mL = 3$.

Figure 2.7: Oct.tree mesh structure numbered with the structured structured numbering for both, nodes and elements.

<i>2</i>	<i>4</i>	<i>6</i>	<i>10</i>
<i>3</i>	<i>5</i>	<i>7</i>	<i>11</i>
<i>1</i>		<i>8</i>	<i>12</i>
		<i>9</i>	<i>13</i>

Figure 2.8: Set of selected elements conforming an analysis mesh. The sequential numbering is written into each element. The lower level elements are numbered first.

Figure 2.8 shows an example of an analysis mesh which includes the corresponding sequential numeration. The sequential numeration is obtained by sequentially numbering the elements in the analysis mesh, without gaps. This allows to obtain a compact numbering that perfectly fits in any FE code. The sequential numeration allows for the use of compact matrices (as opposite to sparse matrices) for data storage along the FE code. It also allows to simplify the code complexity.

2.3.4 Numbering conversion

We have just defined three different numbering systems. These three systems have the required properties to efficiently work into each environment, the virtual mesher or the FEM code. The appropriate numbering system will be used into each environment to take advantage of the different properties. Efficient methods to go from one numbering system to another will be defined.

The virtual numbering perfectly fits into the virtual mesher because of its structure. However, the main drawback of the virtual numbering is that each node or element respectively requires a set of three or four components for its definition. Thus, it does not fit into a FEM code. For instance, the loops and the data storage will be much more complex with the virtual numbering system. The sequential numbering is in general standard for the commercial FE codes. This numbering system perfectly works with loops and data storage, making the code faster. However the sequential numbering system can not be used in the virtual mesher in an efficient manner because no structured information is implicit the sequential numbering system. So, the sequential numbering system will be used in the FEM code and the virtual numbering system will be used in the virtual mesher.

It can be observe that we will be continuously changing from one numbering system to another, then we need an efficient procedure for this purpose. The idea is to use the third numbering system, the structured numbering system, that it is able to live in the FEM code and also in the virtual mesher. As Figure 2.9 shows, to go from the sequential numbering to the virtual numbering and vice versa, we pass thorough the structured numbering. The procedures to convert the numbering systems will now be described

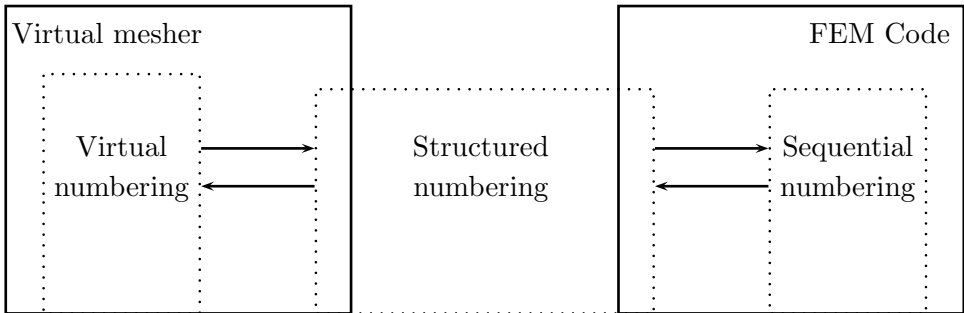


Figure 2.9: Element and node numbering transformation between the VM and the FE code.

Virtual numbering to structured numbering

The quad-tree structure is implicitly contained both, in the structured and in the virtual numbering systems. Therefore, any element or node can be easily converted from the virtual numbering to the structured numbering via an explicit formula. That is, it will change the element numbering, $\check{K} = [L, X, Y, Z]^T$, into a single number K by means of expression (2.11).

$$K = A + B + C + Y \quad (2.11)$$

where

$$\begin{aligned} A &= \sum_{i=0}^{L-1} 2^{i \cdot d} \\ B &= (X - 1) 2^L \\ C &= (Z - 1) 2^{2L} \end{aligned} \quad (2.12)$$

and d is the dimension of the problem, 2D or 3D. A similar procedure is followed for the node numbering defined as $\check{i} = \{X, Y, Z\}^T$ and converted to the sequential one using:

$$i = (Z - 1)2^{2(mL \cdot p + 1)} + (X - 1)2^{mL \cdot p + 1} + Y \quad (2.13)$$

where p is the degree of the FE interpolation.

Structured numbering to virtual numbering

Alternatively it is also possible to convert the sequential numbering into the virtual numbering system, by using function (2.14) where L is obtained bounding the sequential element number into the maximum and minimum number of each level. For instance the maximum number in level 2 ($L = 2$) is 21 and the minimum is 6, and for $L = 3$, 22 and 85, respectively (see Figure 2.7a).

That is, $K > \sum_{i=0}^{L-1} 2^{i \cdot d}$ & $K \leq \sum_{i=0}^L 2^{i \cdot d}$. Then the value of L for which both inequalities hold is the corresponding mesh level.

$$\check{K} = \begin{Bmatrix} L \\ X \\ Y \\ Z \end{Bmatrix} = \begin{Bmatrix} L \\ \text{ceil} \left(\frac{K-A-C}{2^L} \right) \\ K - A - B - C \\ \text{ceil} \left(\frac{K-A}{2^{d \cdot L}} \right) \end{Bmatrix} \quad (2.14)$$

where A , B and C are defined in (2.12). Note that we will evaluate first Z , then X and finally Y .

Finally, the virtual node numbering \check{i} is obtained with (2.15):

$$\check{i} = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{Bmatrix} \text{ceil} \left(\frac{i - (Z-1)2^{d \cdot (mL \cdot p+1)}}{2^{mL \cdot p+1}} \right) \\ i - (Z-1)2^{d \cdot (mL \cdot p+1)} - (X-1)2^{mL \cdot p+1} \\ \text{ceil} \left(\frac{i}{2^{d \cdot (mL \cdot p+1)}} \right) \end{Bmatrix} \quad (2.15)$$

where mL is the maximum level for the Virtual Mesh (defined by user) and p is the order of the element.

Sequential numbering to structured numbering

Figure 2.8 represents an example of the sequential numbering for an analysis mesh. Now we are interested in converting this sequential numbering into the structured one. Figure 2.10 shows the corresponding structured numbering.

As opposite to the previous case (structural \leftrightarrow virtual relations), in this case there is no physical or data-structure-based relation between both numbering systems. The only way to efficiently convert the sequential numbering into structured numbering is via an indexing vector built for each analysis mesh when constructed. The indexing vector in the case of the analysis mesh in Figure 2.10 that converts the sequential numbering into virtual numbering is represented in Table 2.1.

<i>6</i>	<i>10</i>	<i>14</i>	<i>18</i>
<i>7</i>	<i>11</i>	<i>15</i>	<i>19</i>
<i>3</i>		<i>16</i>	<i>20</i>
		<i>17</i>	<i>21</i>

Figure 2.10: Set of selected elements conforming an analysis mesh. The structured numbering is written into each element.

Sequential	1	2	3	4	5	6	7	8	9	10	11	12	13
Structured	3	6	7	10	11	14	15	16	17	18	19	20	21

Table 2.1: Indexing vector from sequential numbering to structured numbering for the analytical mesh represented in Figure 2.10. The first row represents the position in the indexing vector (the number of the element in the sequential numbering system) and the second row is the value (the number of the element in the structured numbering system) allocated in the corresponding position.

It can be clearly observed that the vector represented in Table 2.1 is compact. To convert the sequential numbering to the structured one, we ask for the value allocated in the position corresponding to the sequential numbering. The transformation from the sequential to the structured numbering systems only requires to read the corresponding value in the indexing vector.

Structured numbering to sequential numbering

To pass from the structured numbering to the sequential one, we follow a similar procedure. Once the analysis mesh is built we construct the corresponding transformation matrix. The vector that transforms the structured numbering into the sequential one for the analysis mesh represented in Figure 2.10 is represented in Table 2.2:

Structured	...	3	...	6	7	...	10	11	...	14	15	16	17	18	19	20	21
Sequential	0	1	0	2	3	0	4	5	0	6	7	8	9	10	11	12	13

Table 2.2: Indexing vector from structured to sequential numbering systems for the analytical mesh represented in Figure 2.10. In the first row we represent the position in the indexing vector (the number of the element in the structured numbering system) and in the second row the value (the number of the element in the sequential numbering system) that is allocated.

Note that the indexing vector represented in Table 2.2 is sparse and in general it will have a great amount of zeros. In this case to obtain the element number in the sequential numbering system we only need to read the value allocated in the position corresponding to the number of the element in the structured numbering system. The size of this indexing vector is limiting the number of elements in the structured numbering system, and it is also limiting the maximum level of elements (the minimum element size). For instance, for 2D we can reach 22 refinement levels (4, 194, 304 elements per side) and 14 (16, 384 elements per side) for 3D.

2.3.5 From virtual space to real space. Nodal coordinates

We have described the numbering system and its transformation, however we have obtained a mesh structure that has no relation with the geometry. The final step is to convert the quad-tree (living in the VM space) to a suitable mesh structure in the real space (the space where the problem is living). As

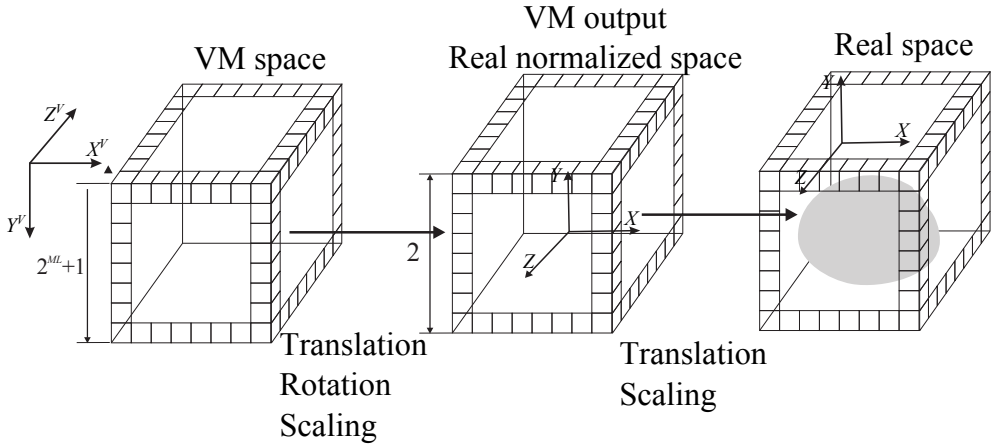


Figure 2.11: Transformation process from the virtual coordinate system to the coordinate system in the real space.

previously described, a node is described in the virtual numbering system by its virtual coordinates. To obtain the nodal coordinates in the real space it will only be necessary to change the virtual coordinate system to the real coordinate system in the real space.

Figure 2.11 represents a scheme of this process. The left hand side of the picture represents the mesh structure in the virtual space (VM space X^V, Y^V, Z^V) represented by a cube of dimensions $2^{mL} + 1$. The first step consists in converting this virtual system to a normalized one (centered in $(0, 0, 0)$ and side length 2), represented at the center of the picture, by means of a translation, rotation and scaling. This normalized system, aligned with the real system, is the standard output of the virtual mesher. The final step consists in a second translation and scaling to obtain the coordinates system in the real space, which is the one used for solving the problem at hand, shown on the right hand side of the picture in Figure 2.11.

Once the nodal coordinates are evaluated in the real coordinates system, we are able to work together with the mesh structure and the geometry of the

problem to finally build the corresponding analysis mesh as it will be described in the following Section.

2.3.6 Generation of the Analysis Mesh

cgFEM is based on the use of a sequence of uniformly refined 2D Cartesian meshes, created by the VM, where hierarchical relations between the different mesh levels have been defined. Note that the virtual mesher is prepared for 3D but the FE code in this work only works with 2D problems, so far. This sequence of meshes used to discretize Ω_E is called the *Cartesian grids pile*, see Figure 2.12a, which embeds all the problem domain Ω and 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 [48]. 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 grids pile* where all elements are geometrically similar to the element used in the coarsest level (level 0) of the *Cartesian grids pile*, called the *reference element*. The *Cartesian grids pile* is built using the virtual mesher functions previously described. One of the main benefits of the data structure is that, as it will be described in Section 2.3.8, in the linear elastic case all elements of the *Cartesian grids 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 pre-evaluation of most of the information used by the FE code, remarkably influencing on the efficiency of the code.

The first step of the analysis consists in creating the *analysis mesh* used to obtain the FE solution of the problem. This mesh is built taking a set of

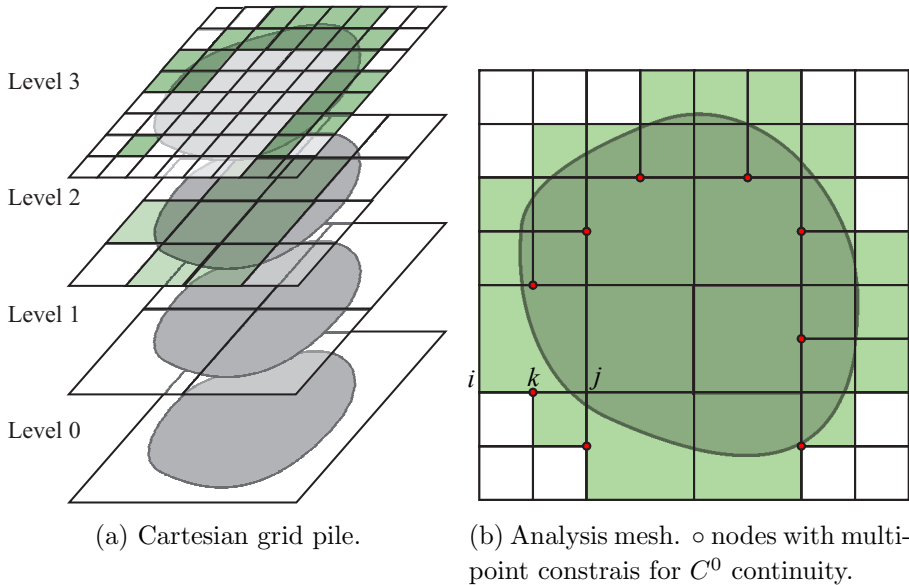


Figure 2.12: Difference between the Cartesian grids pile and the analysis mesh.

non-overlapped elements of different sizes, see Figure 2.12b, taken from the different levels of the *Cartesian grids pile*. A maximum difference of 1 refinement level is allowed between adjacent elements in the *analysis mesh*. Due to this, as the reader can observe in Figure 2.12b, the resulting h -adapted mesh is not conforming, then the required C^0 continuity of the FE solution is not guaranteed. This issue is overcome by using Multi-point constrains (MPCs) [49, 50], which enforces the C^0 continuity between adjacent elements of different levels. The lower-left corner in Figure 2.12b shows nodes i, j and k . k is a so-called hanging node, as it is present in the smaller element but not in the bigger one, thus preventing the C^0 continuity between elements. MPC equations are imposed to enforce C^0 continuity. The constraint equation reads $\mathbf{u}_k = 0.5\mathbf{u}_i + 0.5\mathbf{u}_j$, for linear elements. Note that in the data structure i and j are the k 's parent nodes previously introduced.

Figure 2.12 shows elements of different levels. The element level indicates the relative size of the element with respect to the *reference element*, that is the

level 0 element on the bottom of the *Cartesian grid pile* in 2.12a. That is, level 1 elements are obtained splitting the level 0 element sides 2^1 times. The elements of level 2 are obtained by subdivision of the elements in level 1 and are 2^2 times smaller than the reference element. The next levels are evaluated similarly.

Taking into account that all elements in the *Cartesian grid pile*, are geometrically similar, that is, they are all square elements, whose only difference is their size, it is possible to set some relation between the main element characteristics, such as the element stiffness matrix, Jacobian matrix, etc. Ródenas *et al.* define in [48] 5 characteristic properties that are related between geometrically similar elements. The parameter relating the elements is the relative elements size, ζ .

In the cgFEM framework $\zeta = 2^{-L}$. Then we could easily relate all the following properties with those of the *reference element* indicated with the subindex 0.

- *Jacobian matrix*: $\mathbf{J} = \zeta \mathbf{J}_0$.
- *Inverse Jacobian*: $\mathbf{J}^{-1} = \frac{1}{\zeta} \mathbf{J}_0^{-1}$.
- *Jacobian*: $|\mathbf{J}| = \zeta^D |\mathbf{J}|_0$, where D is the problem dimension (2 for 2D).
- *Shape function derivatives matrix*: $\mathbf{B} = \frac{1}{\zeta} \mathbf{B}_0$.
- *Stiffness matrix*: $\mathbf{k} = \zeta^{D-2} \mathbf{k}_0$, for \mathbf{D} being constant.

2.3.7 Geometry-mesh intersection. Integration

According to the relation of the elements with the boundary of the domain $\partial\Omega$, the analysis mesh is formed by three element types, see Figure 2.13:

- *Boundary elements*: elements placed along the domain boundary, $\partial\Omega$. Only a part of each of these elements remains into the problem domain Ω . The evaluation of the stiffness matrix of these elements requires to evaluate 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 placed into the domain.
- *Internal elements*: elements fully located into the domain. These are treated as standard FE elements. All these elements point to the same object in the data structure that stores the information of the *reference element*.
- *External elements*: elements fully located outside of the domain and, therefore, not considered in the analysis.

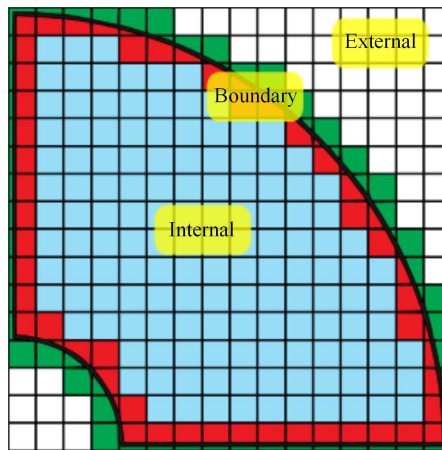


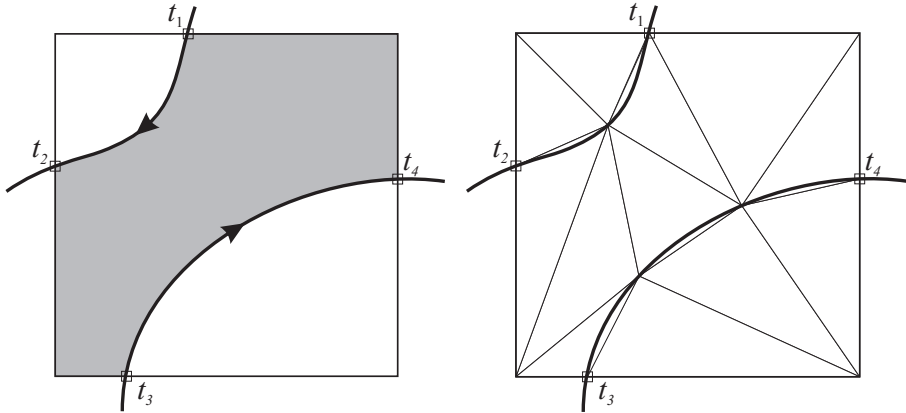
Figure 2.13: Element types according to their relation with the problem boundary: Internal elements, external elements and boundary elements.

The boundary elements are the critical elements in cgFEM. Approximations of the integration domain in these elements would lead to geometrical errors in the evaluation of the element matrices that could spoil the theoretical convergence rate of the FE analyses. An exact or at least sufficiently precise representation of the integration domain into these elements is therefore

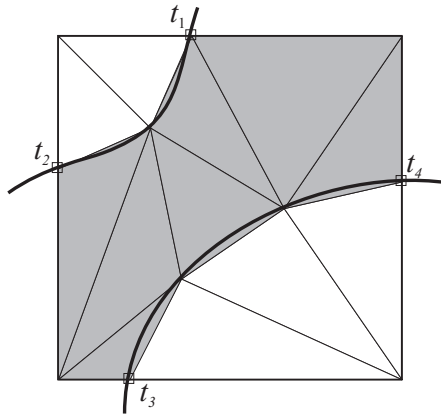
required. The bibliography shows several methods to perform the domain integrals in these elements. One method was proposed by García-Ruiz and Steven [29] where they multiply the stiffness matrix of intersected elements by the ratio of the area $|\Omega_E^e \cap \Omega|$ to the area of the element $|\Omega_E^e|$. Some authors propose to detect the nodes outside or close to the boundary of the domain and move them to the boundary [51, 18]. Daneshmand and Kazemzadeh-Parsi [9] propose to subdivide the intersected area into triangular subdomains and use them just for integration. The process used in this work is similar to that shown in [9] and consists in three steps:

- *Intersection of boundary with element edges.* Figure 2.14a shows the intersections of the curves that define the boundary of the domain with the Cartesian Grid element edges. The shaded area represents the problem domain.
- *Addition of intermediate points.* As shown in figure 2.14b we will identify some extra points placed on the curves that define the boundary. The number of these extra points is related to the curvature of the boundary. In order to increase the performance of the code the curvature is not evaluated but a quantity that is related with it. We evaluate the distance between the mid-point of the segment of the intersecting curve and an imaginary straight line between the input point and the output point. The number of additional points is proportional with this quantity. This set of points together with the vertex nodes of the element are used to create a Delaunay triangulation that defines integration subdomains at each boundary element.
- *Selection of internal triangular subdomains.* Figure 2.14c represents the integration subdomains selected to evaluate the element stiffness matrix.

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 integration. Thereby, the integration in those elements will be



(a) Geometry and problem domain, intersection with elements. (b) Intermediate point addition and Delaunay's tessellation.



(c) Integration subdomain generation.

Figure 2.14: Intersection and subdomain generation process in boundary elements.

performed using these integration subdomains. Note that in the case of multi-material problems, those triangles now considered falling outside the problem domain could be considered of a different material and also considered in the integration with different properties. Note that this would represent an homogenization of the material properties.

Contour integrals. Natural boundary conditions

The consideration of the natural boundary condition requires the evaluation of contour integrals. Contour integrals are also required along the FE analysis process to evaluate certain quantities. An example could be the value of the J -integral in problems with singularities.

As a parametric definition of the boundaries is available, the boundary integration is performed along the boundary Γ of the domain, and not along its linear discretization. In Figure 2.15a we can observe the boundary Γ intersecting an element K at intersection points I_j and I_k . We add some extra points depending on the relative curvature of the boundary. In this case we added two more points I_{m1} and I_{m2} , which divide the part of the boundary falling into K , Γ^K , into three integration sections Γ_{iS}^K , represented by different colors. NiS is the number of the integration sections. The integral of an arbitrary function $f(\mathbf{x})$, where \mathbf{x} are the coordinates of the points along the boundary, is performed by the following expression, where Ω^K is the domain of element K .

$$\int_{\Omega^K \cap \Gamma} f(\mathbf{x})d\Gamma = \sum_{i=1}^{NiS} \int_{\Gamma_{iS}^K} f(\mathbf{x})d\Gamma \quad (2.16)$$

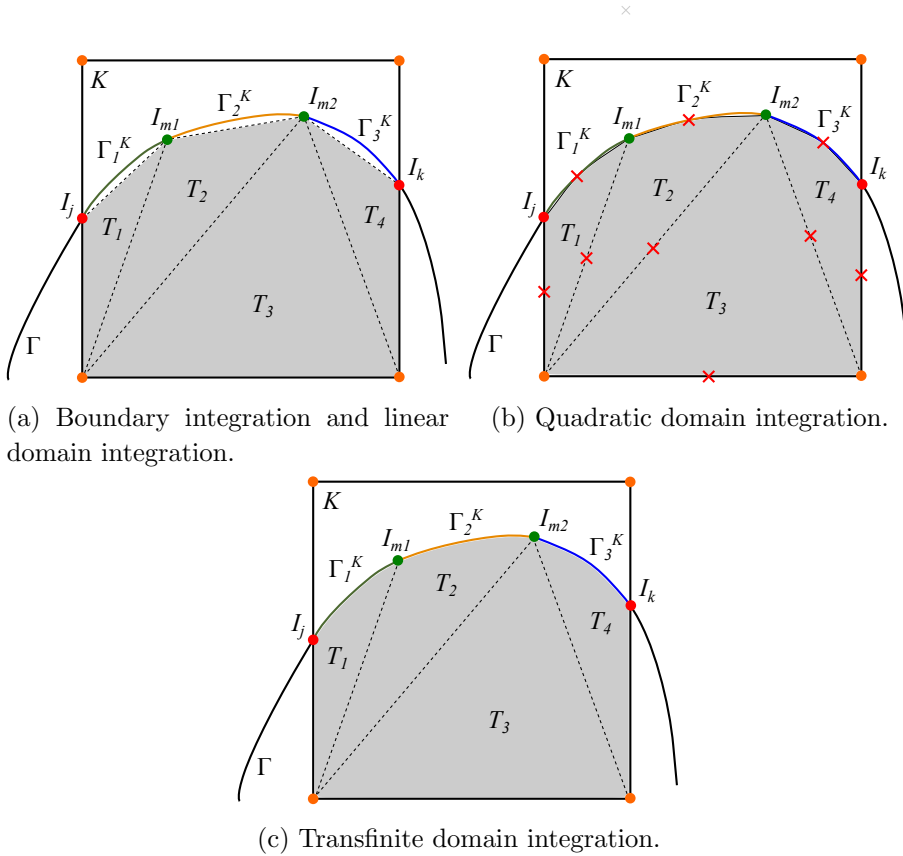


Figure 2.15: Integration domains for different integrations schemes.

Domain integration

Domain integration is crucial to achieve a good quality in the FE results. The standard FEM with an isoparametric mapping uses the same discretization to represent the geometry and the solution. As the geometrical error due to the isoparametric geometry representation converges faster than the FE solution, the convergence rate of the solution is the theoretical convergence rate associated to the discretization error. In cgFEM, we use a more sophisticated way to represent the geometry of the domain because the FE nodes are not

placed along Γ . Once the Delaunay tessellation is performed, we have some triangular integration subdomains into each boundary element. For instance, in Figures 2.15 we have 4 subdomains. The domain integrals in the boundary elements are evaluated considering these integration subdomains using (2.17).

$$\int_{\Omega^K} f(\mathbf{x})d\Omega = \sum_{i=1}^{NT} \int_{T_i} f(\mathbf{x})d\Omega \quad (2.17)$$

where NT is the number of triangular integration subdomains in K placed in $\Omega \cap \Omega^K$ and T_i is each of the integration subdomains.

The next step is the numerical integration of each one of the subdomains. We have three different approaches: a) linear approximation to the boundary, Figure 2.15a, b) quadratic approximation to the boundary, Figure 2.15b and c) transfinite mapping, Figure 2.15c. These three approaches have been ordered in terms of increasing accuracy and computational cost. Our experience shows that in order to guarantee the optimal convergence of the FE solution, at least the approximation to the boundary should be of the same order than the FE interpolation, that is a) for bilinear elements and b) for biquadratic elements.

Linear and quadratic integration The side of the triangular integration subdomain used to represent the curved boundary can be approximated by a straight line segment or by a quadratic polynomial as shown in Figure 2.16. Given an integration quadrature of a number of integration points ($NIntP$) placed at local triangle coordinates $(\xi, \eta)_j$ with weights ω_j like those represented in Figure 2.16a, we can obtain various coordinate transformations depending on the degree of approximation to the real boundary. So, the main difference between them is the location, in real coordinates, of the integration points and the Jacobian of the coordinate transformation. The integration of an arbitrary function $f(\mathbf{x})$ is performed using (2.18).

$$\int_{T_i} f(\mathbf{x})d\Omega = \sum_j^{NIntP} f(\mathbf{x}_j) |\mathbf{J}(\mathbf{x}_j)| \omega_j \quad (2.18)$$

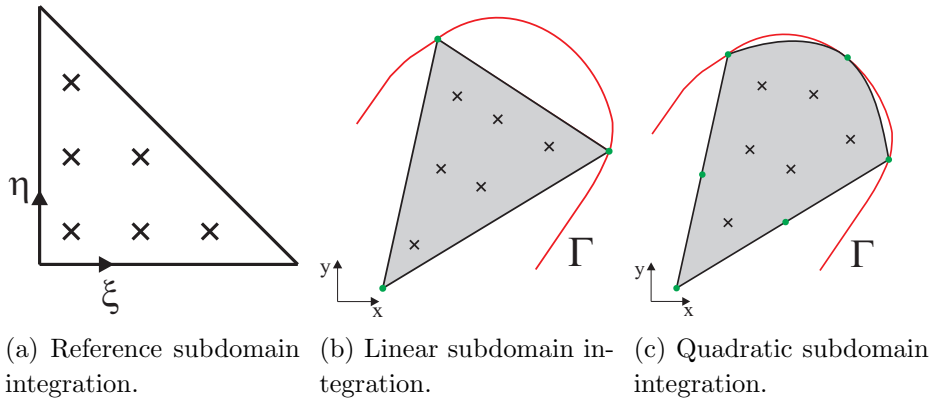


Figure 2.16: Integration area for linear and quadratic integrations schemes.

where \mathbf{x}_j are the global coordinates of the j^{th} integration point. \mathbf{x}_j is defined by the coordinates transformation $\mathbf{x}_j = \widehat{\mathbf{N}}^q(\xi, \eta)_j \widehat{\mathbf{x}}$ from local coordinates (ξ, η) to global coordinates (x, y) . Where $\widehat{\mathbf{N}}^q(\xi, \eta)_j$ is the shape functions matrix of the triangular subdomain and $\widehat{\mathbf{x}}$ are the coordinates of the points used to define the interpolation (equivalent to the nodes of the elements used in FEA), green points at Figure 2.16.

Transfinite integration Alternatively it is also possible to use the transfinite mapping technique, commonly used in p -adaptive analysis [52] to consider the exact geometry. The use of this mapping increases the computational cost per subdomain but reduces the number of triangular subdomains required for a given accuracy. In Figure 2.17a we show a reference triangle, and a generic triangle in global coordinates in Figure 2.17b defined by its vertex and the curves connecting those vertices.

Each one of the curves of the edges of the triangle in global coordinates is defined by a parametric expression $c_i(t_i)$, $i = 1, 2, 3$, according to (2.19). A parameter t_i is defined at each side of the triangle in the local reference system (2.20), where L_i are the area coordinates of the reference triangle. All this allows us to evaluate the position, in global coordinates (X, Y) , of each

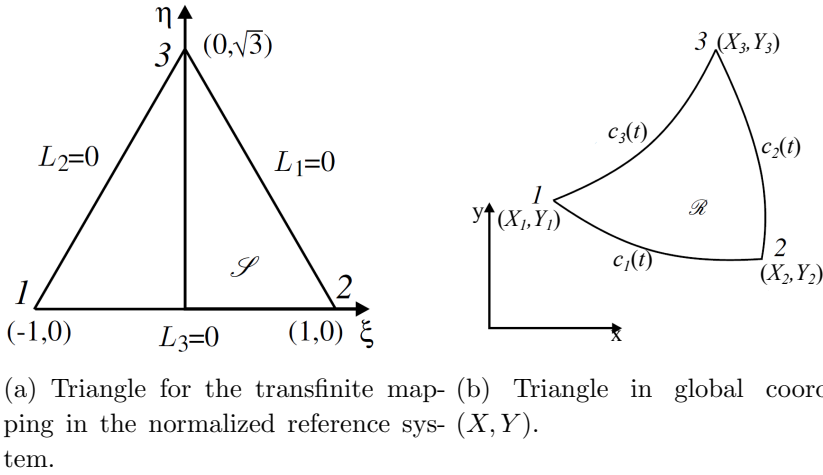


Figure 2.17: Triangular elements for the transfinite mapping in local and global reference systems.

integration point and its corresponding area, used for integration along the code.

$$c_i(t_i) = \begin{Bmatrix} x_i(t_i) \\ y_i(t_i) \end{Bmatrix} \quad (2.19)$$

$$t_1 = L_2 - L_1 \quad t_2 = L_3 - L_2 \quad t_3 = L_1 - L_3 \quad (2.20)$$

Now, following the ideas presented in [53] we obtain the coordinates mapping defined in (2.21). The Jacobian matrix (2.25) of the transfinite mapping is

obtained form (2.22,2.23,2.24).

$$\begin{aligned}
 x(L_1, L_2, L_3) = & L_1 X_1 + L_2 X_2 + L_3 X_3 + \\
 & + \frac{4L_1 L_2}{1 - t_1^2} \left[x_1(t_1) - \left(\frac{1 - t_1}{2} X_1 + \frac{1 + t_1}{2} X_2 \right) \right] + \\
 & + \frac{4L_2 L_3}{1 - t_3^2} \left[x_2(t_2) - \left(\frac{1 - t_2}{2} X_2 + \frac{1 + t_2}{2} X_3 \right) \right] + \\
 & + \frac{4L_3 L_1}{1 - t_3^2} \left[x_3(t_3) - \left(\frac{1 - t_3}{2} X_3 + \frac{1 + t_3}{2} X_1 \right) \right]
 \end{aligned} \tag{2.21}$$

$$\begin{aligned}
 y(L_1, L_2, L_3) = & L_1 Y_1 + L_2 Y_2 + L_3 Y_3 + \\
 & + \frac{4L_1 L_2}{1 - t_1^2} \left[y_1(t_1) - \left(\frac{1 - t_1}{2} Y_1 + \frac{1 + t_1}{2} Y_2 \right) \right] + \\
 & + \frac{4L_2 L_3}{1 - t_3^2} \left[y_2(t_2) - \left(\frac{1 - t_2}{2} Y_2 + \frac{1 + t_2}{2} Y_3 \right) \right] + \\
 & + \frac{4L_3 L_1}{1 - t_3^2} \left[y_3(t_3) - \left(\frac{1 - t_3}{2} Y_3 + \frac{1 + t_3}{2} Y_1 \right) \right]
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial x(L_1, L_2, L_3)}{\partial L_1} &= X_1 + \\
 &+ \frac{4L_2(1-t_1^2) - 8L_1L_2t_1}{(1-t_1^2)^2} \left[x_1(t_1) - \left(\frac{1-t_1}{2}X_1 + \frac{1+t_1}{2}X_2 \right) \right] + \\
 &\frac{4L_1L_2}{1-t_1^2} \left(-\frac{\partial x_1(t_1)}{\partial t_1} - \frac{X_1 - X_2}{2} \right) + \\
 &+ \frac{4L_3(1-t_3^2) - 8L_3L_3t_3}{(1-t_3^2)^2} \left[x_3(t_3) - \left(\frac{1-t_3}{2}X_3 + \frac{1+t_3}{2}X_1 \right) \right] + \\
 &\frac{4L_3L_1}{1-t_3^2} \left(\frac{\partial x_3(t_3)}{\partial t_3} - \frac{X_1 - X_3}{2} \right)
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial x(L_1, L_2, L_3)}{\partial L_2} &= X_2 + \\
 &+ \frac{4L_3(1-t_2^2) - 8L_2L_3t_2}{(1-t_2^2)^2} \left[x_2(t_2) - \left(\frac{1-t_2}{2}X_2 + \frac{1+t_2}{2}X_3 \right) \right] + \\
 &\frac{4L_2L_3}{1-t_2^2} \left(-\frac{\partial x_2(t_2)}{\partial t_2} - \frac{X_2 - X_3}{2} \right) + \\
 &+ \frac{4L_1(1-t_1^2) - 8L_1L_2t_1}{(1-t_1^2)^2} \left[x_1(t_1) - \left(\frac{1-t_1}{2}X_1 + \frac{1+t_1}{2}X_2 \right) \right] + \\
 &\frac{4L_1L_2}{1-t_1^2} \left(\frac{\partial x_1(t_1)}{\partial t_1} - \frac{X_2 - X_1}{2} \right)
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial x(L_1, L_2, L_3)}{\partial L_3} &= X_3 + \\
 &+ \frac{4L_1(1-t_3^2) - 8L_3L_1t_3}{(1-t_3^2)^2} \left[x_3(t_3) - \left(\frac{1-t_3}{2}X_3 + \frac{1+t_3}{2}X_1 \right) \right] + \\
 &\frac{4L_3L_1}{1-t_3^2} \left(-\frac{\partial x_3(t_3)}{\partial t_3} - \frac{X_3 - X_1}{2} \right) + \\
 &+ \frac{4L_1(1-t_2^2) - 8L_2L_3t_2}{(1-t_2^2)^2} \left[x_2(t_2) - \left(\frac{1-t_2}{2}X_2 + \frac{1+t_2}{2}X_3 \right) \right] + \\
 &\frac{4L_2L_3}{1-t_2^2} \left(\frac{\partial x_2(t_2)}{\partial t_2} - \frac{X_3 - X_2}{2} \right)
 \end{aligned}$$

(2.22)

$$\begin{aligned}
\frac{\partial y(L_1, L_2, L_3)}{\partial L_1} &= Y_1 + \\
&+ \frac{4L_2(1-t_1^2) - 8L_1L_2t_1}{(1-t_1^2)^2} \left[y_1(t_1) - \left(\frac{1-t_1}{2}Y_1 + \frac{1+t_1}{2}Y_2 \right) \right] + \\
&\frac{4L_1L_2}{1-t_1^2} \left(-\frac{\partial y_1(t_1)}{\partial t_1} - \frac{Y_1 - Y_2}{2} \right) + \\
&+ \frac{4L_3(1-t_3^2) - 8L_3L_3t_3}{(1-t_3^2)^2} \left[y_3(t_3) - \left(\frac{1-t_3}{2}Y_3 + \frac{1+t_3}{2}Y_1 \right) \right] + \\
&\frac{4L_3L_1}{1-t_3^2} \left(\frac{\partial y_3(t_3)}{\partial t_3} - \frac{Y_1 - Y_3}{2} \right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial y(L_1, L_2, L_3)}{\partial L_2} &= Y_2 + \\
&+ \frac{4L_3(1-t_2^2) - 8L_2L_3t_2}{(1-t_2^2)^2} \left[y_2(t_2) - \left(\frac{1-t_2}{2}Y_2 + \frac{1+t_2}{2}Y_3 \right) \right] + \\
&\frac{4L_2L_3}{1-t_2^2} \left(-\frac{\partial y_2(t_2)}{\partial t_2} - \frac{Y_2 - Y_3}{2} \right) + \\
&+ \frac{4L_1(1-t_1^2) - 8L_1L_2t_1}{(1-t_1^2)^2} \left[y_1(t_1) - \left(\frac{1-t_1}{2}Y_1 + \frac{1+t_1}{2}Y_2 \right) \right] + \\
&\frac{4L_1L_2}{1-t_1^2} \left(\frac{\partial y_1(t_1)}{\partial t_1} - \frac{Y_2 - Y_1}{2} \right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial y(L_1, L_2, L_3)}{\partial L_3} &= Y_3 + \\
&+ \frac{4L_1(1-t_3^2) - 8L_3L_1t_3}{(1-t_3^2)^2} \left[y_3(t_3) - \left(\frac{1-t_3}{2}Y_3 + \frac{1+t_3}{2}Y_1 \right) \right] + \\
&\frac{4L_3L_1}{1-t_3^2} \left(-\frac{\partial y_3(t_3)}{\partial t_3} - \frac{Y_3 - Y_1}{2} \right) + \\
&+ \frac{4L_1(1-t_2^2) - 8L_2L_3t_2}{(1-t_2^2)^2} \left[y_2(t_2) - \left(\frac{1-t_2}{2}Y_2 + \frac{1+t_2}{2}Y_3 \right) \right] + \\
&\frac{4L_2L_3}{1-t_2^2} \left(\frac{\partial y_2(t_2)}{\partial t_2} - \frac{Y_3 - Y_2}{2} \right)
\end{aligned}$$

(2.23)

$$\begin{Bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2\sqrt{3}} & -\frac{1}{2\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial L_1} \\ \frac{\partial}{\partial L_2} \\ \frac{\partial}{\partial L_3} \end{Bmatrix} \quad (2.24)$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (2.25)$$

The transfinite mapping allows us to represent the exact geometry up to numerical integration errors. A representation of the location of the integration points into a triangular subdomain is shown in Figure 2.18 for the case of transfinite mapping. The location of integration points in this Figure can be compared to the location of integration points for the two previous approximations shown in Figure 2.13. Note that as there are no approximations in the definition of the subdomain, the integrals are free from geometrical errors. Therefore, the accuracy of the integrals depends on the number of integration points. As before, for any arbitrary function $f(\mathbf{x})$ we use (2.18) where in this case the Jacobian and the global coordinates of the integration point are obtained using the expressions shown above.

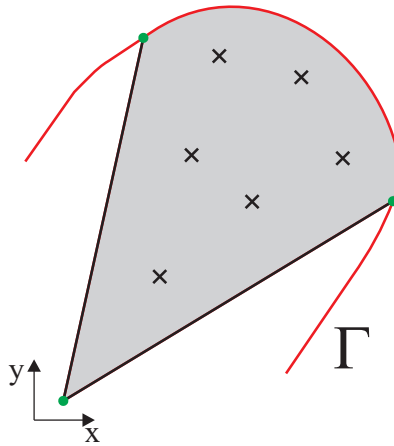


Figure 2.18: Integration area for transfinite mapping.

2.3.8 Element data sharing

The use of Cartesian grids together with the data structure used in the implementation of the cgFEM methodology allow for simple and efficient information data sharing between elements. This considerably reduces the total amount of calculations, thus improving the computational efficiency of the FE code. The shared data corresponds to information related to integration, that is: integration points, weight, element stiffness, \mathbf{B} matrix corresponding to the derivatives of the shape functions and the element nodal equivalent forces from the Neumann boundary condition. This section describes, for the internal and boundary elements the sharing information process during the analysis.

Internal elements

As shown in Section 2.3.6 the terms used to evaluate the stiffness matrix (\mathbf{B} matrix and Jacobian matrix \mathbf{J}) of geometrically similar elements are related by a constant value evaluated as a function of the scaling factor ς between the elements. In fact, the stiffness matrices of geometrically similar elements are simply related by a factor ς^{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 in the 2D case, 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 the efficiency of the generation of the system of equations to be solved by the FE code.

Boundary elements

On boundary elements the hierarchical data structure enables for the use 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 2.3.7. 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 allows 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 2.19 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 finer mesh of the information evaluated in the coarsest mesh. In the finer mesh new element matrices are only evaluated for white elements.

Structural shape optimization problems: horizontal data sharing

The structural shape optimization problem will be detailed in Chapter 7, and a brief introduction was given in Chapter 1. Let us recall that the lower level

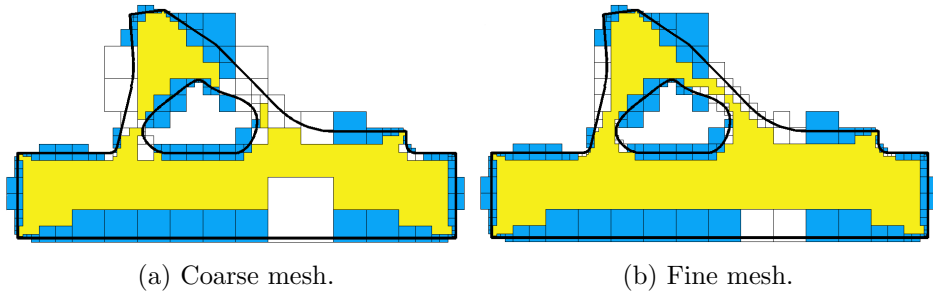


Figure 2.19: *Vertical data sharing* in two consecutive meshes of the h -adaptive analysis of a gravity dam. The data structure relates the stiffness matrices of internal elements (yellow) to that of the *reference element*. The calculations for boundary element colored in blue are reused. Element matrices in the finest mesh are only evaluated for white elements.

of the shape optimization process is in charge of analyzing each of the geometries proposed by the higher level during the iterative process. In our case we will use cgFEM 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.

With the traditional FEM it is almost impossible to enable an efficient exchange of information between elements of different geometries because, in general, the elements of different geometries are completely different and completely unrelated, as each geometry requires a different mesh conformal to the boundary. However, if we use cgFEM considering the same Cartesian grids 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*, i.e. 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:

- *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 2.20).
- *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 2.20).

The *horizontal data sharing* consist of 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 2.20 shows an 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 2.20a,b for geometry i are reused in geometry j represented in Fig. 2.20c. 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.

2.4 h -adaptive refinement strategies

The computational 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

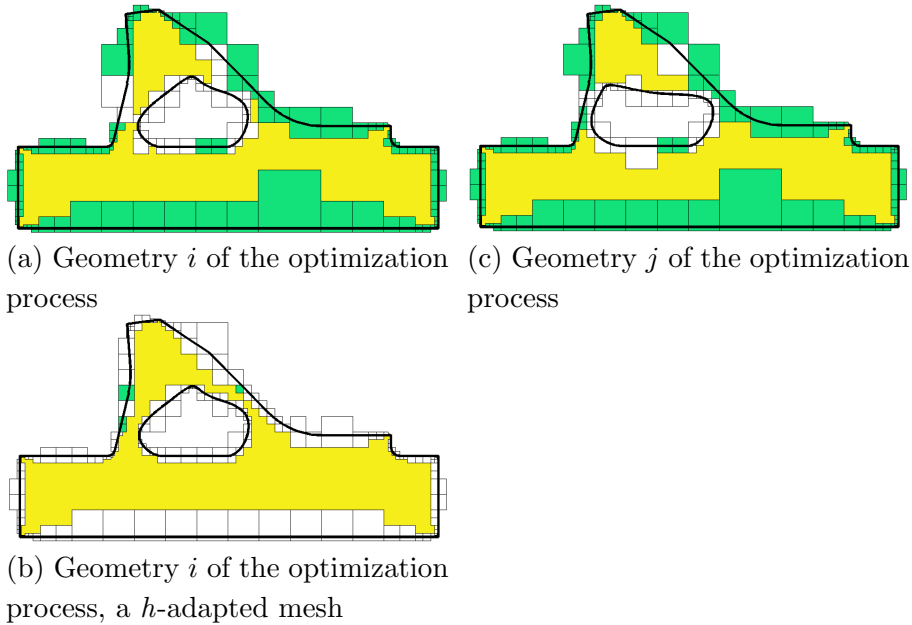


Figure 2.20: Comparison of two different geometries $i < j$ during an optimization process. The data for the green elements evaluated for geometry i are reused in geometry j .

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 [2], leading to non-optimal solutions, reducing the convergence rate to the optimal solution or even preventing convergence.

cgFEM implements two h -refinement strategies. As in [54], the first one is based 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, or, alternatively, on the error in Quantities of Interest as described in Chapter 4.

2.4.1 Geometrical refinement

The analysis starts adapting the dimensions of the Cartesian Grid domain (Ω_E) to the problem domain Ω in order to ensure that Ω is embedded into Ω_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 curvature of the boundary is too large with respect to the element size. A simple curvature indicator is defined in (2.26), where, as represented in Figure 2.21, the values of d_i represent the distances between the intersection 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 defining the curve (red crosses), obtained during the intersection process, see Figure 2.14. The refinement process is repeated until the relative curvature indicator k is smaller than a user-defined value (its default value in the cgFEM code is 0.03). The first analysis mesh is created as a result of this process. Table 2.3 shows examples of preliminary meshes and the first analysis meshes.

$$k = \frac{\sum_{i=1}^{NIP} |d_i|}{NIP \cdot L} \quad (2.26)$$

2.4.2 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. The exact error in energy norm of the solution is given by:

$$|||e|||_{\Omega}^2 := \int_{\Omega} (\boldsymbol{\sigma} - \boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} (\boldsymbol{\sigma} - \boldsymbol{\sigma}^h) \, d\Omega \quad (2.27)$$

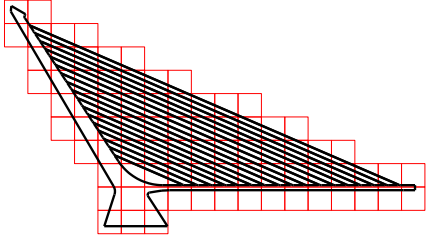
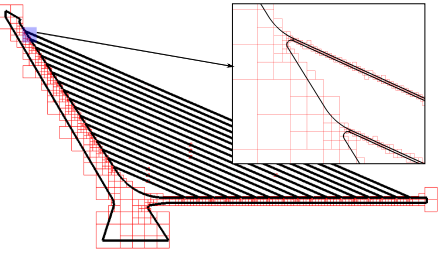
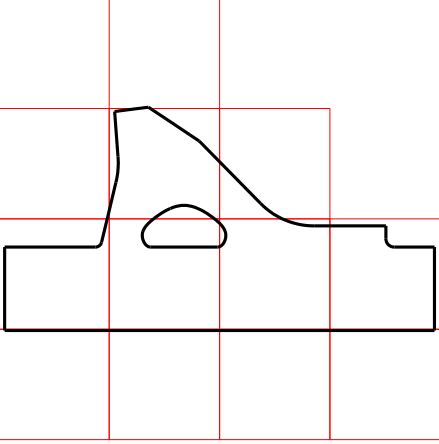
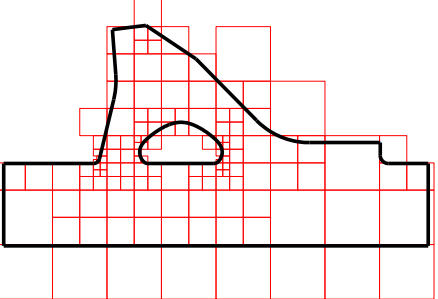
Preliminary mesh	1 st analysis mesh geometrically adapted
 A diagram showing a preliminary mesh for a curved structure. The mesh consists of a coarse grid of red squares overlaid on a black outline of the structure. The mesh is uniform in size, with large elements covering most of the area.	 A diagram showing the first analysis mesh, which is geometrically adapted. The mesh is refined (smaller elements) in regions of high curvature, such as the sharp corner and the curved surface. An inset box provides a magnified view of the curved region, showing the dense, triangular mesh elements following the curve's geometry.
 A diagram showing a preliminary mesh for a complex, multi-lobed shape. The mesh is a coarse, uniform grid of red squares overlaid on a black outline of the shape.	 A diagram showing the first analysis mesh, which is geometrically adapted. The mesh is refined (smaller elements) in regions of high curvature, such as the sharp peaks and the curved surface. The mesh is coarser in the straight, flat regions of the shape.

Table 2.3: Comparison between the preliminary mesh and the geometrically adapted mesh using the curvature criterion.

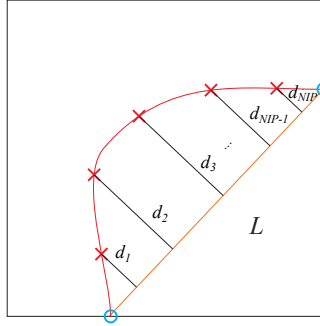


Figure 2.21: Intersection points and distances used for the evaluation of the curvature error indicator k

where $\boldsymbol{\sigma}^h$ is the FE stress field and, neglecting other error sources, $\mathbf{e} = \mathbf{u} - \mathbf{u}^h$ is the discretization error. In order to estimate the error in energy norm we use the Zienkiewicz & Zhu (ZZ) error estimator (2.28), presented in [3], where $\boldsymbol{\sigma}^*$ is an improved stress field, more accurate than $\boldsymbol{\sigma}^h$.

$$\|\mathbf{e}\|_{\Omega}^2 \approx \mathcal{E}_{ZZ}^2 := \int_{\Omega} (\boldsymbol{\sigma}^* - \boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} (\boldsymbol{\sigma}^* - \boldsymbol{\sigma}^h) \, d\Omega \quad (2.28)$$

$\boldsymbol{\sigma}^*$ could be in general any improved solution. It is easy to deduce that the quality of the estimation is directly related with the quality of the recovered stress field.

Particularizing (2.28) at 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 [55], we obtain the new levels (sizes) of the elements in each zone. Examples of analysis meshes obtained by this procedure are represented in Table 2.4.

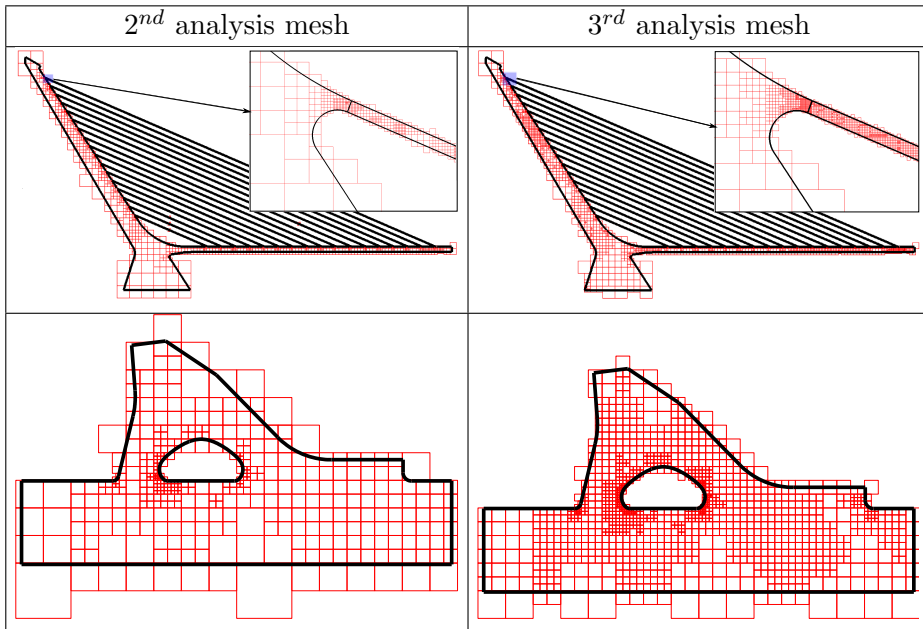


Table 2.4: Second and third analysis mesh obtained using the error estimation information following the 1^{st} meshes represented in Table 2.3

2.5 Essential boundary conditions

In the cgFEM the nodes are not placed over the boundary of the domain, then the Dirichlet boundary conditions can not be applied directly over the nodes. In this case, we follow a Lagrange multipliers approach. The proposed approach have been developed in collaboration with the Department of Mechanical Engineering and Material Engineering at Universitat Politècnica de València. In this case we do not use any stabilization procedure because for the particular case of the examples used in this thesis it is not necessary. An improvement of the proposed method including a stabilization term is recently introduced in [56]. Consider that the solution of the problem (2.8) is equivalent to the following one, where we introduce the Lagrange multipliers field $\boldsymbol{\lambda}$ to impose the Dirichlet boundary conditions:

$$\left\{ \begin{array}{ll} \text{Find } \mathbf{u} \in \mathcal{U}, \boldsymbol{\lambda} \in \mathcal{L} : & \\ a(\mathbf{u}, \mathbf{v}) + b(\boldsymbol{\lambda}, \mathbf{v}) = l(\mathbf{v}) & \forall \mathbf{v} \in \mathcal{U} \\ b(\boldsymbol{\mu}, \mathbf{u}) = b(\boldsymbol{\mu}, \bar{\mathbf{u}}) & \forall \boldsymbol{\mu} \in \mathcal{L} \\ \text{where } b(\mathbf{w}, \mathbf{v}) = \int_{\Gamma_D} \mathbf{w}^T \mathbf{v} \, d\Gamma & \end{array} \right. \quad (2.29)$$

where $\mathcal{U} = H^1(\Omega)$ and \mathcal{L} is the space of the Lagrange multipliers defined on Γ_D . The bilinear form $b(\boldsymbol{\lambda}, \mathbf{v})$ represents the virtual work of the reactions over Γ_D . Applying the FE discretization to (2.29) and considering the discrete subspaces $\mathcal{U}^h \subset \mathcal{U}$ and $\mathcal{L}^h \subset \mathcal{L}$ we obtain:

$$\left\{ \begin{array}{ll} \text{Find } \mathbf{u}^h \in \mathcal{U}^h, \boldsymbol{\lambda}^h \in \mathcal{L}^h : & \\ a(\mathbf{u}^h, \mathbf{v}) + b(\boldsymbol{\lambda}^h, \mathbf{v}) = l(\mathbf{v}) & \forall \mathbf{v} \in \mathcal{U}^h \\ b(\boldsymbol{\mu}, \mathbf{u}^h) = b(\boldsymbol{\mu}, \bar{\mathbf{u}}) & \forall \boldsymbol{\mu} \in \mathcal{L}^h \end{array} \right. \quad (2.30)$$

This discrete system (2.30) yields a linear system of equations that will give as a result the displacements at nodes and also the discrete Lagrange multipliers field:

$$\begin{bmatrix} \mathbf{K} & \mathbf{C}_D^T \\ \mathbf{C}_D & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u}^h \\ \boldsymbol{\lambda}^h \end{Bmatrix} = \begin{Bmatrix} \mathbf{F} \\ \hat{\mathbf{u}} \end{Bmatrix} \quad (2.31)$$

Stiffness matrix \mathbf{K} and force vector \mathbf{F} in (2.31) are built as in the standard FE formulation, but in this case we will have two new terms, namely $\hat{\mathbf{u}}$ and the constraints matrix \mathbf{C}_D . Now, we present a brief summary about the evaluation of the new terms related with the functional $b(\cdot, \cdot)$.

We define the discrete space for the Lagrange multiplier field as the intersection between the analysis mesh and the Dirichlet boundary, excluding from that space the initial and final points of the boundary. In Figure 2.22 we observe an example of the discretization. Then the discretized space \mathcal{L}^h for the Lagrange multipliers field is:

$$\begin{aligned}\lambda^h &= \sum M_i \lambda_i^h \\ \mu^h &= \sum M_i \mu_i^h\end{aligned}\tag{2.32}$$

where i represents each intersection point, λ_i is the value of the Lagrange multiplier at that intersection point, μ_i its variation and M_i is the shape function on the support of i , that is $\overline{\Gamma_1 \cup \Gamma_2}$, as represented in Figure 2.22. Note that the shape function M_i is constant on the extremes of the boundary. Then, for the intersection point i , we write the constraint equation C_i and the r.h.s. \hat{u}_i :

$$\begin{aligned}b(\boldsymbol{\mu}^h, \mathbf{u}^h) &\rightarrow C_i = \left(\int_{\overline{\Gamma_1 \cup \Gamma_2}} M_i \mathbf{N} \, d\Gamma \right) \tilde{\mathbf{u}}^h \\ b(\mathbf{v}, \boldsymbol{\lambda}^h) &\rightarrow C_i^T = \left(\int_{\overline{\Gamma_1 \cup \Gamma_2}} M_i \mathbf{N} \, d\Gamma \right)^T \lambda_i^h \\ b(\boldsymbol{\mu}^h, \bar{\mathbf{u}}) &\rightarrow \hat{u}_i = \int_{\overline{\Gamma_1 \cup \Gamma_2}} M_i \bar{u} \, d\Gamma\end{aligned}\tag{2.33}$$

where \mathbf{N} is a vector containing the FE shape functions involved in the element which is intersected by Γ_i such that for a single component of the displacement field $u^h = \mathbf{N} \bar{\mathbf{u}}^h$. Note that the integrals of the first two equations in 2.33 are equal, thus leading to a symmetric system matrix.

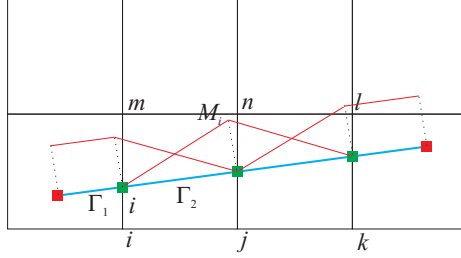


Figure 2.22: Shape function for intersection point i .

2.6 Solver

Efficiently solving the global system of the equations is one of the key aspects in a FE code to guarantee a high performance. In this thesis we have worked in two directions, the first one is based on the reordering of the stiffness matrix in order to improve direct solver speed. The second one is related to iterative solvers. In the last case we have studied projection techniques from coarser discretizations in order to obtain initial guesses for the iterative solvers.

Since cgFEM uses non-conforming meshes containing hanging nodes (see Figure 2.12b) and the Dirichlet boundary conditions are imposed via Lagrange multipliers, the linear system will have constraint equations. The constraints related with the Dirichlet boundary conditions are grouped in the matrix \mathbf{C}_D and the constraints related with the hanging nodes are grouped in the matrix \mathbf{C}_{MPC} . These last constraints enforce continuity between adjacent elements of different refinement levels using Multi-Point Constraint (MPC) via Lagrange multipliers. The global system for the analysis is:

$$\begin{bmatrix} \mathbf{K} & \mathbf{C}_D^T & \mathbf{C}_{MPC}^T \\ \mathbf{C}_D & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{MPC} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\lambda}_D \\ \boldsymbol{\lambda}_{MPC} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F} \\ \hat{\mathbf{u}} \\ \mathbf{0} \end{Bmatrix} \quad (2.34)$$

In order to decrease size of the system (2.35) we reduce it by eliminating the constraint matrix related with the hanging nodes, \mathbf{C}_{MPC} , from the system.

We define the system of equations:

$$\hat{\mathbf{K}}\hat{\mathbf{U}} = \hat{\mathbf{F}} \quad (2.35)$$

where:

$$\hat{\mathbf{K}} = \begin{bmatrix} \check{\mathbf{K}} & \check{\mathbf{C}}_D^T \\ \check{\mathbf{C}}_D & \mathbf{0} \end{bmatrix} \quad \hat{\mathbf{U}} = \begin{Bmatrix} \mathbf{u} \\ \lambda_D \end{Bmatrix} \quad \hat{\mathbf{F}} = \begin{Bmatrix} \check{\mathbf{F}} \\ \hat{\mathbf{u}} \end{Bmatrix} \quad (2.36)$$

Defining \mathbf{K}^{row} and \mathbf{F}^{row} as the rows of \mathbf{K} and \mathbf{F} corresponding to the degrees of freedom constrained with a MPC, and \mathbf{K}^{col} and \mathbf{C}_D^{col} the respective columns, then we obtain:

$$\begin{aligned} \check{\mathbf{K}} &= \mathbf{K} + \mathbf{K}^{col}\mathbf{C}_{MPC} + \mathbf{C}_{MPC}^T\mathbf{K}^{row} \\ \check{\mathbf{F}} &= \mathbf{F} + \mathbf{C}_{MPC}^T\mathbf{F}^{row} \\ \check{\mathbf{C}}_D &= \mathbf{C}_D + \mathbf{C}_{MPC}\mathbf{C}_D^{col} \end{aligned} \quad (2.37)$$

Removing from (2.35) the rows and columns affected by the hanging nodes we obtain the reduced system of equations to solve:

$$\begin{bmatrix} \mathbf{K}' & \mathbf{C}'_D{}^T \\ \mathbf{C}'_D & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u}' \\ \lambda_D \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}' \\ \hat{\mathbf{u}} \end{Bmatrix} \quad (2.38)$$

Finally, the displacements associated to the degrees of freedom of the hanging nodes are evaluated as follows:

$$\mathbf{u}_{MPC} = \mathbf{C}_{MPC}\mathbf{u}' \quad (2.39)$$

2.6.1 Nested Domain Decomposition-based direct solver

When a direct solver is used to solve a system of equations such as the one presented in 2.38, a previous reordering is usually used in order to improve the performance of the solver. Usually this reordering is obtained via an optimization process which not always obtains the best reordering for the system of equations. In this Section we propose to use a Nested Domain Decomposition (NDD) reordering technique. This technique is based on the Cartesian grid

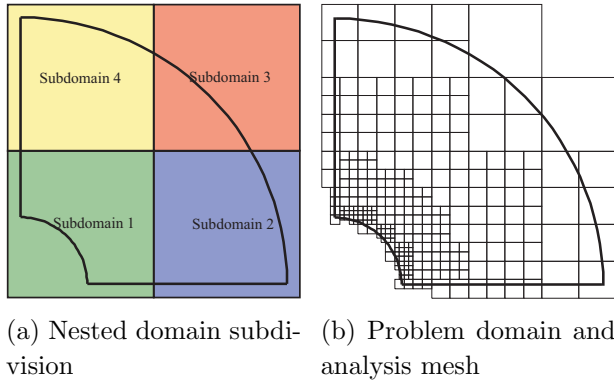


Figure 2.23: Nested scheme 1.

structure, directly related to the mesh topology. Thus, providing an optimal reordering. That will provide a considerable reduction of the computational cost associated to the resolution of the system of equations with a minimum computational cost.

The NDD technique was used in [48] and applied to a hierarchical FE code. The NDD technique consist in recursively subdividing the domain of the problem into subdomains. Then, we reorder all DoF such that all of them falling into a subdomain will be allocated together in the stiffness matrix. The nodes falling in the interface of the subdomains will not be reordered and will simply be moved to the end of the matrix. This idea can be recursively applied into each original subdomain, hence it can be called *nested*.

For the cgFEM code, we subdivide the domain Ω_E in 4 subdomains or regions as shown in Figure 2.23a. Each subdomain is represented in a different color. If we now take the *Cartesian grid pile* represented in Figure 2.12a we can identify those subdomains with the elements of level 1. Thus, the nested reordering in cgFEM will be made up by grouping the nodes according to the corresponding element in the *Cartesian grid pile*. Figure 2.23b shows an example of an analysis mesh where we are going to apply the nested reordering.

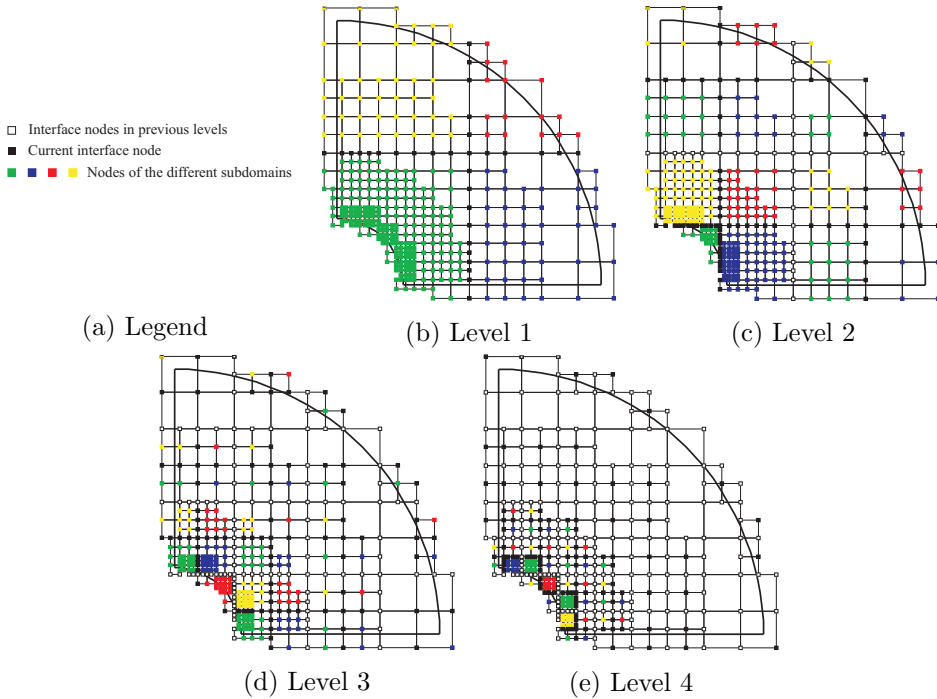


Figure 2.24: Nested scheme 2.

For instance, in Figure 2.24b we have the level 1 nested subdivision. The nodes are subdivided into 5 different regions. The colored ones indicate the nodes falling into each one of the elements of level 1. Black nodes are those falling in the interface between the level 1 elements. The stiffness matrix will be reordered grouping all nodes of the same color, see Figure 2.25b. Level 2 reordering, Figure 2.24c, indicates that each of the level 1 subdomains is again reordered in the same way. In this case we are using the 2^{nd} level of elements represented in Figure 2.12a. For instance, the green subdomain in Figure 2.24b is subdivided into 4 subdomains and the interface in black as shown in Figure 2.24c. Interfaces of previous levels are represented by white squares. The same process is followed for the next levels, using the elements of the corresponding level of the *Cartesian grid pile*.

The result of the NDD is shown in the stiffness matrices presented in Figure 2.25. As observed, the NDD produces an arrowhead type stiffness matrix. In Figure 2.25a we present the original non-reordered stiffness matrix with the constrains described in (2.35) placed at the end of the matrix. We observe the 4 subdomains corresponding to the level 1 reordering, represented with their respective colors in Figures 2.24b to 2.24e. When we increase the reordering level, reordering the nodes falling into the level 1 subdomains, we can perceive how each of those level 1 subdomains is internally reordered, leading again to an arrow-head structure into the subdomains. The recursive application of this process generates a nested arrow-head structure as shown in the evolution from Figure 2.25b. This arrow-head structure leads to a decrease in the fill-in of the factorized matrices used in the direct linear solver. For instance, in this case, the fill-in of the LU factorization of the matrix in Figure 2.25a is 12.35% while the fill-in of the corresponding factorization for the matrix in Figure 2.25e, when the NDD is used, is 8.12%.

In order to obtain a better comprehension of the NDD reordering we analyze in Figure 2.26 the matrix in Figure 2.25e. In this case we focus on the recursive reordering in one of the level 1 subdomains. For instance we study the evolution of the reordering of the level 1, the green subdomain. Zooming the green block in level 1 shows up a new image corresponding to a level 2 reordering of the green subdomain. We observe that the arrowhead structure is repeated again. Now, taking for instance the blue subdomain of the level 2 subdomain and zooming again we can observe the corresponding level 3 subdomain, again arrowhead shaped. Finally, zooming again the highlighted level 3 subdomain we obtain the level 4 blocks, again arrowhead shaped. This process will continue until all nodes are reordered. A nested arrowhead structured matrix is therefore obtained.

Note that for this implementation we have used the NDD just as a reordering procedure to further increase the standard direct solver. Its use in parallel solvers will be part of future works to improve the direct solver performance.

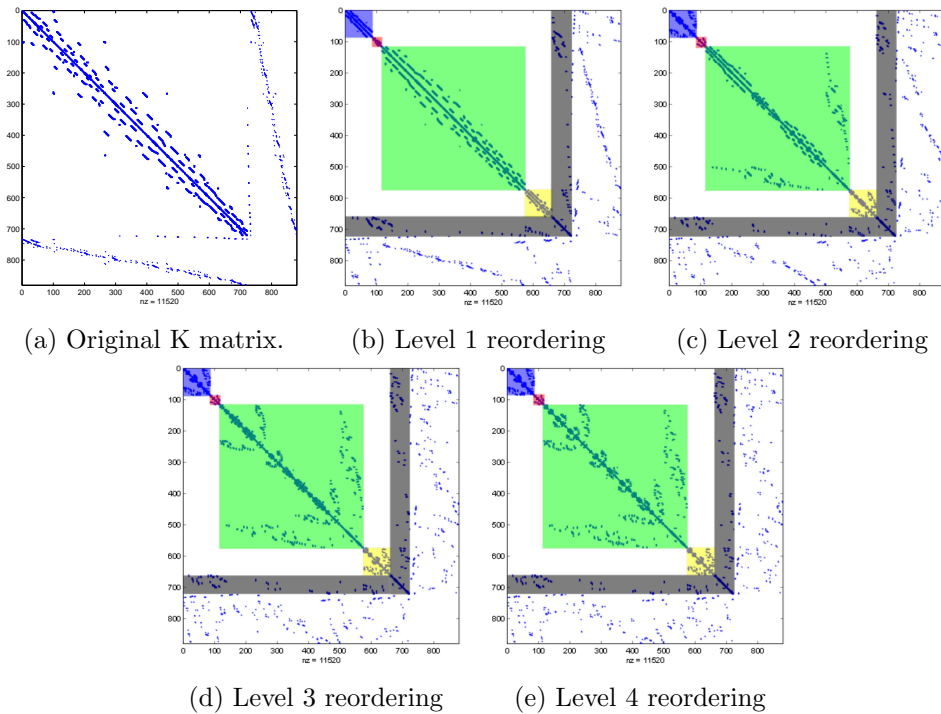


Figure 2.25: Nested scheme K.

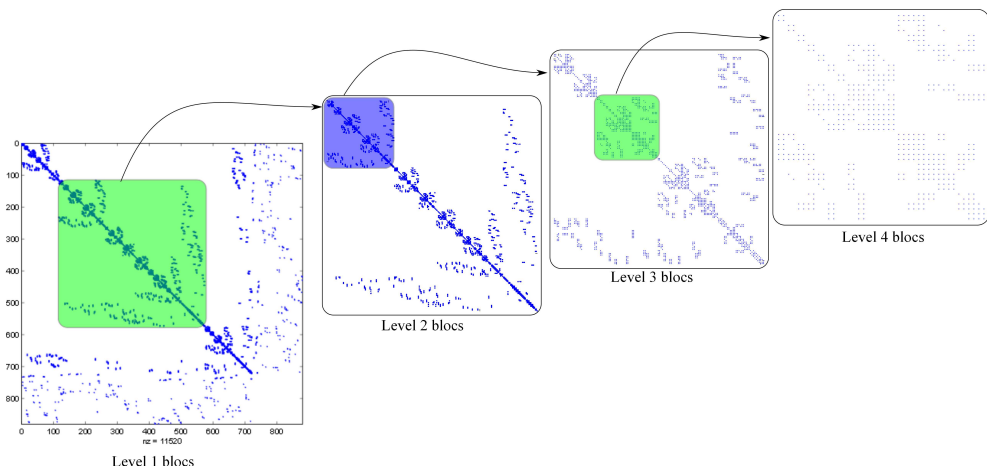


Figure 2.26: Representation of the nested arrowhead structure.

2.6.2 Iterative solver. Projections

Direct solvers are competitive for small and medium size systems of equations. However, large system of equations require the use of iterative solvers. These solvers try to iteratively approximate the exact solution in most cases using Krylov subspaces. We will consider the CGS iterative solver implemented in Matlab[®]2010b.

Among other factors, the computational cost of these methods depends on the initial guess of the solution used to initialize the iterative process. Accurate initial vectors would in general reduce the number of iterations. Note that because of the use of the cgFEM, projection techniques between different meshes are both, easily implemented and computationally very efficient.

Because of the use of the cgFEM code, the essential boundary conditions are imposed via Lagrange multipliers (2.35). In order to overcome this issue we propose the use of the Augmented Lagrange formulation. For a similar situation, in the context of contact problems, some iterative techniques have been reported in [57]. In this Section, the linear system to be solved considering the Augmented Lagrange formulation is the following:

$$\begin{bmatrix} \mathbf{K}' + \mathbf{C}'_D{}^T \mathbf{W} \mathbf{C}'_D & \mathbf{C}'_D{}^T \\ \mathbf{C}'_D & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u}' \\ \boldsymbol{\lambda}_D \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}' + \mathbf{C}'_D{}^T \mathbf{W} \mathbf{q} \\ \hat{\mathbf{u}} \end{Bmatrix} \quad (2.40)$$

where \mathbf{W} is any square symmetric positive semidefinite matrix of the same size than the number of constraints. In our implementation we define \mathbf{W} as $\mathbf{W} = \gamma \mathbf{I}$ with $\gamma > 0$. In particular, we take $\gamma = \|\mathbf{K}\|_{L_2} / \|\mathbf{C}_D\|_{L_2}^2$ in order to minimize the condition number. As preconditioner we use the one proposed in [58]:

$$\mathbf{P} = \begin{bmatrix} \text{diag}(\mathbf{K}' + \mathbf{C}'_D{}^T \mathbf{W} \mathbf{C}'_D) & \mathbf{C}'_D{}^T \\ \mathbf{C}'_D & \mathbf{0} \end{bmatrix} \quad (2.41)$$

As mentioned above, we will feed the iterative solver with an initial solution, but we have to take into account that the solution vector of the system in (2.35) has two parts: one related to the displacements solution \mathbf{u} and the

other related to the Lagrange multipliers $\boldsymbol{\lambda}$. For the first part we consider two different approaches:

- the projection of the FEM solution in mesh $i - 1$ into mesh i ,
- the projection of recovered solution \mathbf{u}_u^* evaluated in mesh $i - 1$ into mesh i . This recovered displacement field, \mathbf{u}_u^* , is obtained via a displacement recovery process (SPR-CD) considering the FE solution and information related with the boundary conditions and body forces. A full description of the SPR-CD is detailed in Section 3.3.

For the Lagrange multipliers we took into account that they represent the traction over Γ_D required to satisfy the prescribed displacements over Dirichlet boundaries. Then, we have two options for them:

- To project the FEM stress field $\boldsymbol{\sigma}^h$ from the previous mesh. The FE stress field is not continuous, then to evaluate the stress value in a point over the interface between two elements we randomly choose the value corresponding to one of the elements connected to the point of interest.
- To project the recovered stress field $\boldsymbol{\sigma}_\sigma^*$ from the previous mesh. This recovered stress field, $\boldsymbol{\sigma}_\sigma^*$ is evaluated from \mathbf{u}_u^* , see Section 3.3.

2.7 Numerical Results

The numerical results presented in this Section are mainly focused in demonstrating the efficiency of the cgFEM code to improve the computational efficiency of the main parts of the FE procedure. Two parts of the method benefits from the cgFEM code. The first one is related to the mesh generation process and the second to the resolution of the system of equations. Regarding the mesh generation process:

- The cgFEM code does not need to evaluate any mesh since it is already virtually created.
- The element stiffness matrices have to be evaluated only for the elements intersected by the boundary and not in the whole domain as in the traditional FEM. That is a $(d - 1)$ dimensional process, which leads to a considerable improvement in the computational cost.

Once system (2.35) is built, the cgFEM code can provide improvements to solve it in two different directions.

- For direct solvers, due to the hierarchy of the mesh the NDD reordering is almost costless. With that reordering method, the results show a considerable computational cost reduction in comparison with standard reordering procedures.
- For iterative solvers, projection techniques of the FE solution from one mesh to other meshes are easy to implement and computationally inexpensive. When the projection techniques are used in iterative solvers, the results shows a decrease in the number of iterations, and consequently a computational cost improvement.

For this Section, we have run several tests always using the same problem to show the behavior of the cgFEM code. The problem at hand is a hollowed cylinder under internal pressure as represented in Figure 3.13. In this case, we only model a quarter of the cross section. We have applied symmetry boundary conditions along the vertical and horizontal sides. The external side is load-free. The exact solution and material properties are shown in Figure 3.13. Uniform refinement and bilinear elements (Q4u) have been considered. For the numerical examples we have used a PC DELL PE1950 equipped with two processors Intel Xeon E5430 with 32Gb of memory. The operating system is Windows Server 2003 Enterprise x64 SP2. The cgFEM code is implemented in Matlab[®] 2010b.

2.7.1 Evaluation of the global system of equation

The main objective of this Section is to compare the computational cost involved in the evaluation of the global system of equations when cgFEM is used and to compare it with a commercial code like ANSYS® 12.1. In order to perform that comparison we divide the total cost into meshing tasks and solving tasks. Meshing tasks refer to the generation of the FE mesh. When the commercial code is used, this process consists in generating a free mesh of linear elements. When cgFEM is used, this process consist in generating the analysis mesh from the virtual mesh generator and intersecting the mesh with the geometry.

The second step generates the element stiffness matrices and the element load vector, assembles them and solves the global system of equations. The main difference in this process is that in the standard FEM we need to evaluate the element stiffness matrices for all elements in the mesh, whereas in cgFEM element stiffness matrices are only evaluated for boundary elements.

Note that the computational cost comparison between the commercial compiled code and the cgFEM code fully implemented in Matlab® 2010b presented in this Section is very basic because the main interest of the comparison is to give some numerical evidences about the improvements of the cgFEM technique in comparison with a standard FE commercial implementation.

Figure 2.27 represents the computational cost in seconds needed for the meshing tasks and for the solving tasks. We can directly observe two main details. The first one is that currently, the meshing time is considerably higher than the solving time, hence the importance on developing new techniques to improve this step of the FE analysis. Second, we observe a considerable decrease in the computational cost during the meshing tasks when the cgFEM is used.

Regarding the computational cost related to the solving tasks, we can not observe any considerable difference between both codes. For the cgFEM code we only need to evaluate the element stiffness matrices of the elements cut by the boundary, while the standard FE formulation requires the evaluation of the element stiffness matrix for all the elements in the mesh, which is considerably more expensive. This is not noticed in these results because the cgFEM code is fully implemented in Matlab[®] 2010b, considerably slower than a compiled and optimized code like those used in commercial codes.

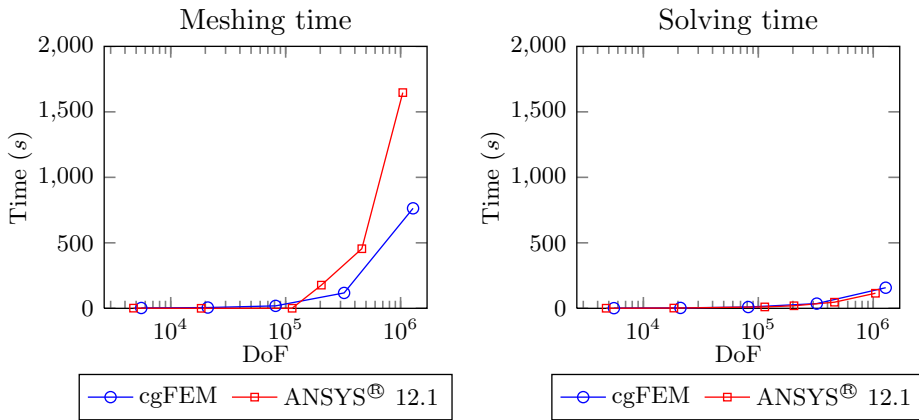


Figure 2.27: Problem 2, Q4u. Computational cost, time in seconds, used for meshing tasks (left) and solving tasks (right). The results are obtained for the commercial code ANSYS[®] 12.1 and for the proposed cgFEM code with the NDD technique.

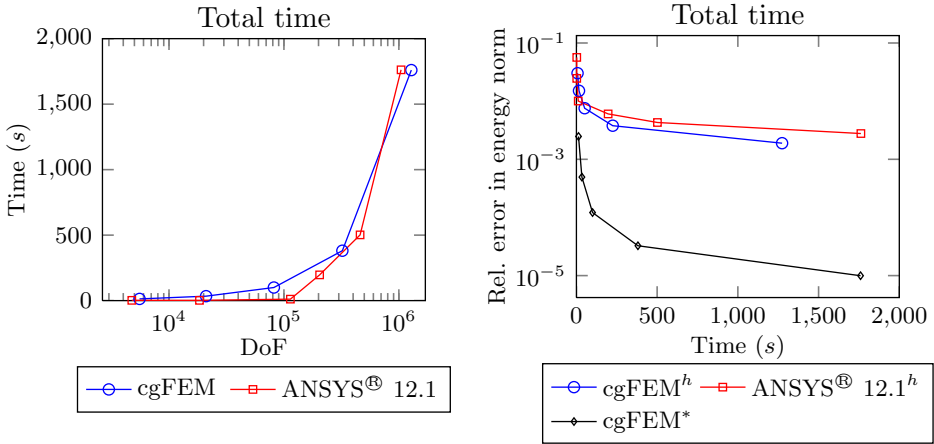
Figure 2.28a represents the total computational cost for solving the linear elasticity problem including the error estimation process in both, the commercial code and the cgFEM code. We observe that the computational cost in both codes is comparable. At this point we have to consider that the error estimation, in both codes, is based on the well-known Zienkiewicz and Zhu error estimator [3] that will be described in detail in Chapter 3. The difference between the commercial code and the cgFEM is that the commercial code just performs a simple nodal averaging to obtain the recovered stress field while the cgFEM evaluates a high quality recovered stress field based on a version of the SPR technique [7] improved to locally satisfy the internal equilibrium and

the Dirichlet and Neumann boundary conditions. In other words, the post-processing procedure used in the cgFEM is considerably more accurate than the simple nodal averaging but, at the same time, is more expensive. However the global computational cost is still comparable with that of the commercial code, despite of the fact that cgFEM is fully implemented in Matlab[®] 2010b. The motivation of using this sophisticated recovery process will be explored in detail in the following Chapters. Finally, Figure 2.28b shows a comparison between the relative error in energy norm and the computational cost of the cgFEM code using a plain nodal averaging (blue line), for error estimation of the FE solution, and the more sophisticated SPR-based recovery process (black line), which allows to evaluate the error of the recovered solution, and ANSYS[®] 12.1. We observe that the FE solution obtained with the cgFEM code requires slightly minor computational cost than the FE solution provided by ANSYS[®] 12.1 for finer meshes. However one of the most important advantages of the cgFEM code is that it is able to provide a recovered solution (see Chapter 3) and an error measure for this recovered solution (see Chapter 6), as a difference of standard commercial codes. This permits to use the recovered field as output instead of the FE solution. In this Figure we clearly show the considerably decrease in the computational cost required for a given accuracy when the recovered solution is considered (black line) with respect to the FE solutions (red and blue lines).

2.7.2 NDD. Computational cost

We utilize three different reordering strategies to evaluate the performance of the technique used to solve the linear system of equations in (2.35): Reference, Matlab[®] 2010b and NDD. We have run the tests in Matlab[®] 2010b, using the standard *backslash* solver provided in this compilation.

- *Reference*: this strategy consists in preventing the standard Matlab[®] 2010b reordering. Then, the system is solved without any previous reordering.



(a) Computational cost of the cgFEM code considering as output the recovered solution compared with the commercial code ANSYS[®] 12.1

(b) Relative error in energy norm versus computational cost for the FE solution obtained with the cgFEM code and the commercial code ANSYS[®] 12.1 and for the recovered solution provided by the cgFEM code.

Figure 2.28: Problem 2, Q4u. Computational cost, time in seconds, used for whole FEM analysis, considering the solution recovery process and the error estimation for a commercial code ANSYS[®] 12.1 and for the proposed code cgFEM with the NDD technique.

- *Matlab[®] 2010b*: in this case we allow Matlab[®] 2010b to apply the standard reordering before solving the system.
- *NDD*: in this case, we use the NDD reordering presented in Section 2.6.1. The nested reordering considers the maximum element level of each mesh.

We have used a sequence of h -uniform refined meshes for the problem presented in Figure 3.13. The evolution of time (s) required to solve the global system of equations (2.35) is presented in Figure 2.29.

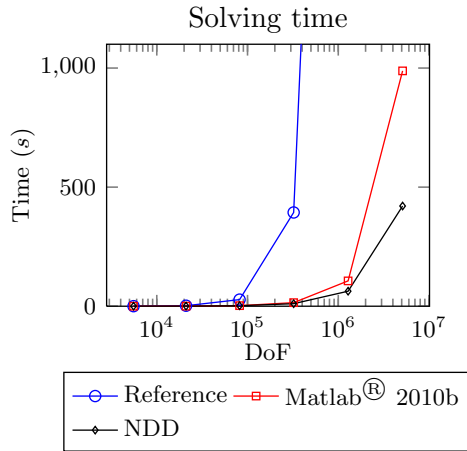


Figure 2.29: Problem 2, Q4u. Computational cost (time in seconds) used to solve the global system (2.35) with different reordering procedures. Reordering time also included.

As we can observe, when no reordering is applied (blue line) the computational cost rapidly increases. Meanwhile when a previous reordering is run (red and black lines), the memory usage and the computational cost decrease considerably, allowing for the resolution of much larger systems of equations. Furthermore, it is also observed that when the matrix is reordered with the NDD technique the computational cost decreases with respect to the use of standard reordering techniques. We observe a reduction of more than a 50% for systems with $5 \cdot 10^6$ degrees of freedom. It is also important to point out that the tendency is to increase this difference for finer meshes. The reason for this behavior is that the NDD reordering could be considered optimal, as it uses the topology of the mesh in the process. The reordering time is also included in the results.

2.7.3 Iterative Solvers. Projections

Iterative solvers require a number of iterations to get a solution with a given level of accuracy. The number of iterations depends on the quality of the preconditioner \mathbf{P} used and also on the quality of the initial vector used to initialize the process. Consider the system $\mathbf{Ax} = \mathbf{b}$, if $\mathbf{P} = \mathbf{A}^{-1}$, then any iterative solver will provide the exact solution with only one iteration: $\mathbf{x} = \mathbf{PAx} = \mathbf{Pb}$, but the computational cost to obtain \mathbf{P} will be excessive. Alternatively, if we feed the iterative solver with the solution of the system, we will directly obtain that solution with one single iteration. This means that the performance of the iterative solvers is strongly related with the pre-conditioner and with the initial guess. We are interested on decreasing the number of iterations for a given pre-conditioner by feeding the iterative solver with accurate initial solutions. In this case we are going to solve with the computationally inexpensive pre-conditioner presented in (2.41). Other pre-conditioners could also be considered.

As previously mentioned, we feed the iterative solver with an initial solution. The solution vector of the system in (2.40)² has two parts. One related with the displacements solution \mathbf{u} and the other related to the Lagrange multipliers $\boldsymbol{\lambda}$. We will only consider three possibilities for the initial vector as described below:

- $\mathbf{U}^0 = \{\mathbf{0} \ \mathbf{0}\}^T$ the null vector, for comparison.
- $\mathbf{U}^h = \left\{ \mathbf{u}_{i-1_p}^h \ \boldsymbol{\lambda}_{i-1_p}^h \right\}^T$ the projection of the FE solution \mathbf{u}_{i-1}^h and the projected traction $\mathbf{G}\boldsymbol{\sigma}_{i-1}^h$ over the Dirichlet boundaries evaluated with the FE stress field. Note that \mathbf{G} is the operator that projects the stress field to the traction along a given boundary, see (2.4).

²Note that the convergence of the iterative solver is affected by the nature of the system of equations

- $\mathbf{U}^* = \left\{ \mathbf{u}_{i-1p}^* \lambda_{i-1p}^* \right\}^T$ the projection of the recovered displacement solution \mathbf{u}_{i-1}^* over the new mesh i and the projection of the recovered traction $\mathbf{G}\boldsymbol{\sigma}_{\sigma i-1}^*$ over the Dirichlet boundaries.

As shown in Figure 2.30, the identification of the position of the DoFs of the analysis mesh i (current mesh) into the previous analysis mesh $i-1$ is very simple due to the hierarchical data structure used by the cgFEM. In Figure 2.31 we present the computational cost of the projection technique measured in equivalent number of iterations needed to project the solution from the coarser mesh $i-1$ to the finer one i . The equivalent number of iterations required to project the solution is measured by comparing the computational cost of the projecting tasks with the computational cost of the iterative solver in the current mesh i . Results show that the importance of the projection decreases for finer meshes, that are the ones of interest, in both situations. Then, the cost associated with the projection becomes small in comparison with the cost of the iterative solver as shown in the Figure.

- Nodes in mesh $i-1$.
- Nodes in mesh i already present in $i-1$ whose values are directly obtained from nodes in $i-1$.
- Nodes in i whose projected solution is interpolated using the standard partition of the unity of the FEM in their parent elements of the mesh $i-1$.

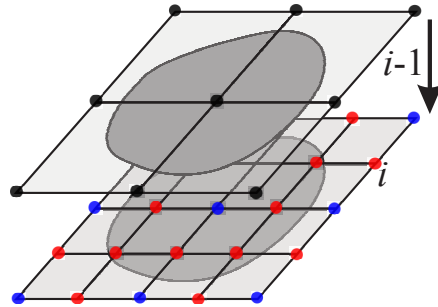


Figure 2.30: Projection of the solution of the analysis mesh $i-1$ into the analysis mesh i .

Results in Table 2.5 shows the results for the iterative solver with different stopping tolerances (10^{-6} , 10^{-8} and 10^{-9}). All the results are obtained with a constraint of 10^4 iterations. In the three tables we observe that there is a range in which the results obtained with the projection of the recovered solution \mathbf{U}^* improves with respect to the use of the projection of the raw FE solution \mathbf{U}^h as initial vector. This is because the recovered solution has a higher convergence

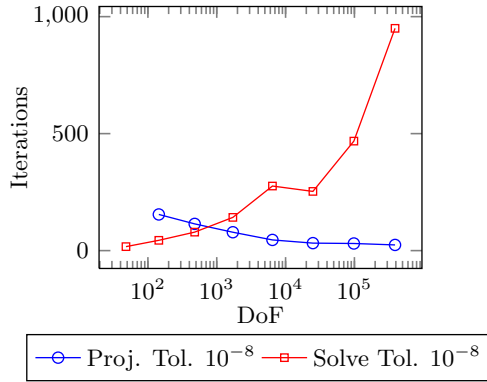


Figure 2.31: Problem 2, Q4u. Number of solving iterations needed to project the initial guess from the coarser mesh ($i - 1$) to the finer one (i) and the number of iterations required to solve the system of equations.

rate, hence the initial vector provided by the recovered solution is closer to the solution of the system of the equations in this range. However, in the first table (tolerance 10^{-6}) we clearly observe that for the last two meshes the projection of the FE solution \mathbf{U}^h improves with respect to the projection of the recovered field \mathbf{U}^* . In this case, the recovered solution has become too much accurate (because of the higher convergence rate) and it differs from the solution of the system of equations. The same behavior is expected in finer meshes for tolerances 10^{-8} and 10^{-9} .

Dashes in Table 2.5 indicate that the iterative solver could not reach a solution. The error encountered in all of these situations is “*one of the scalar quantities calculated during CGS became too small or too large to continue computing.*”. This description is obtained from the help of Matlab[®] 2010b. This indicates that the algorithm can not continue because a numerical error. It is clear from the results shown that the robustness of the iterative solver considerably improves with the use of the recovered displacement solution.

Figure 2.32 represents the results shown in the Table 2.5. We observe that after a certain number of DoF, depending on the tolerance, the use of the projected

recovered solution \mathbf{U}^* as initial guess is clearly a better choice than the use of the projected FE solution \mathbf{U}^h . However this effect disappears for finer meshes. Then there is a range where the recovered solutions provides better results. However, we suggest the use of \mathbf{U}^* since it considerably improves the robustness of the iterative solver.

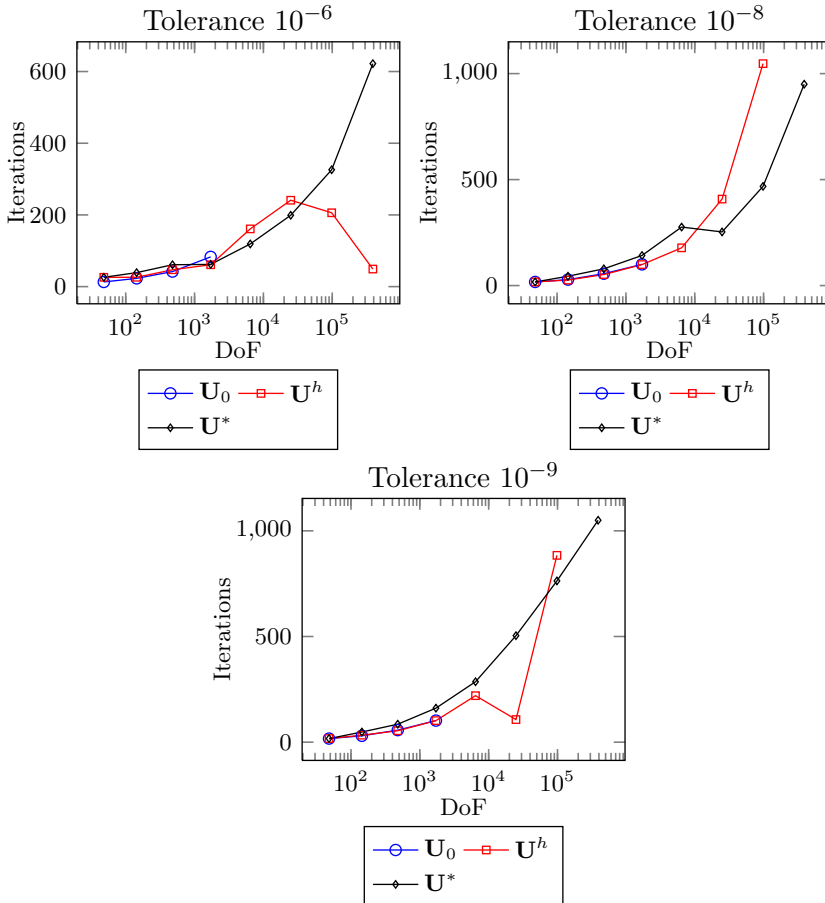


Figure 2.32: Problem 2, Q4u. Number of iterations needed using as initial solution the null vector, the FEM solution and the recovered solution. All cases consider the projection of the Lagrange multipliers.

2.8 Conclusions

In this Chapter we have presented the cgFEM code used to solve the linear elasticity problem. The cgFEM technology has several parts such as: the virtual mesher, the geometry-mesh intersection, the integration procedures, etc. described during this Chapter. All these parts have been optimized in order to make the cgFEM code competitive with commercial codes even if the cgFEM is fully implemented in Matlab[®] 2010b. As the results show, there is an important improvement in the first steps of the Finite Element analysis, i.e. the mesh generation and the creation of the element stiffness matrices. This is one of the key features of the cgFEM that reduces the number of calculations as a result of the hierarchical data structure used with the nested Cartesian grids. The information sharing procedures between elements of different meshes for a given geometry or even between different geometries play an important role in this aspect. These information sharing procedures can be easily applied under the structural shape optimization framework, providing a considerable improvement in the computational cost of the optimization process [6], as shown in Chapter 7.

We have obtained an important improvement when the NDD is used. This technique permits to reorder the global system of equation in an optimal manner. This reordering comes out naturally within the cgFEM framework because of the hierarchy of the mesh. The NDD reordering consumes a small computational cost shown in the numerical tests. As future work, and continuing with the NDD reordering, the resulting system could be easily adapted to parallel solvers because the arrow-head shape matrix resulting after the reordering can be easily divided into small subsystems suitable for parallelization.

In the case of iterative solvers the initial guess is important for their performance. Usually this initial guess is the projection into current mesh of previous solutions. Under the cgFEM framework, projection techniques between different meshes are highly efficient, improving the performance of the iterative

solvers. Under some circumstances, that require further studies, the projection of the recovered solution, \mathbf{U}^* , represents an improvement with respect to the projection of \mathbf{U}^h . However, the main characteristic of the projection of \mathbf{U}^* is its higher robustness with respect to the used of \mathbf{U}^h .

Table 2.5: Problem 2, Q4u. Number of iterations needed for different initial vectors. All cases consider the projection of the Lagrange multipliers. A maximum of 10^4 iterations is allowed. The tolerance for the stopping criterion in the iterative process is indicated on each table.

Tolerance 10^{-6}			
NDoF	N Iter \mathbf{U}_0	N Iter \mathbf{U}^h	N Iter \mathbf{U}^*
48	13	26	26
144	23	26	39
478	42	47	61
1,714	83	61	62
6,444	–	161	119
24,944	–	241	199
98,168	–	206	326
389,348	–	49	622

Tolerance 10^{-8}			
NDoF	N Iter \mathbf{U}_0	N Iter \mathbf{U}^h	N Iter \mathbf{U}^*
48	17	17	17
144	28	26	44
478	56	52	79
1,714	100	99	142
6,444	–	178	276
24,944	–	408	253
98,168	–	1,047	468
389,348	–	–	950

Tolerance 10^{-9}			
NDoF	N Iter \mathbf{U}_0	N Iter \mathbf{U}^h	N Iter \mathbf{U}^*
48	17	17	17
144	30	33	48
478	57	54	85
1,714	102	102	161
6,444	–	220	286
24,944	–	107	504
98,168	–	884	763
389,348	–	–	1,050

Chapter 3

Error estimation and recovery procedures

Chapter 2 described how to build the FE model from the geometrical model and also how to obtain the FE displacement solution. However, the FE solution is only an approximation to the exact solution, which is usually unknown. It is important to know what kind of error it is introduced during the numerical analysis. Analyzing the procedure followed to obtain the numerical solution we find out several sources of error. In Figure 3.1 we present a simplified scheme where we can observe the different sources of error during the simulation of a physical system.

During the idealization process, we try to obtain a mathematical model from the physical phenomena. That is, we condense the behavior of the phenomena into a mathematical expression. Sometimes, the mathematical model, that is, the differential equation, does not accurately describe the physics of the phenomena. For instance sometimes in the analysis at the crack tip in a mechanical component, we consider linear material behavior when we know that there exists some local plastic deformation. This is an usual practice

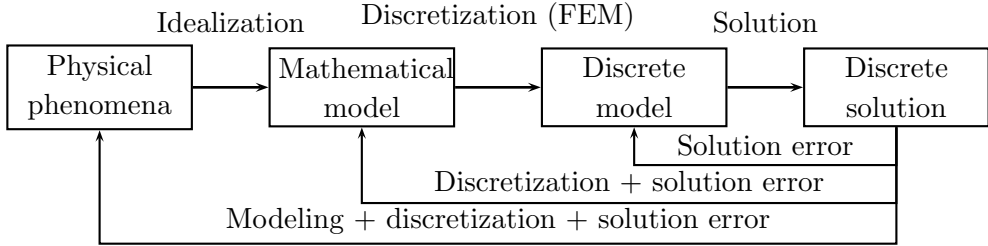


Figure 3.1: Scheme for the error sources during the simulation of a physical phenomena.

that George Box summarized in the sentence “*All models are wrong, some are useful*”.

Generally the geometry of the component to analyze, represented by a CAD model, is very complex, including small details (features). It is a common practice to eliminate these small details by means of the “defeaturing” process before applying the discretization process. The discretization process consists in converting the mathematical model in a discrete one. For a given degree of interpolation functions this error depends on the element size and will show up as two different kinds of errors:

- The geometrical error with respect to the CAD model. In this case the volume (area in 2D) of the actual CAD model is discretized into small regions, called elements, which generally cannot exactly fit the CAD model. For instance, Figure 3.2 shows two different meshes for the same domain. It can be observed that the geometrical error in the finer mesh has decreased.
- The solution discretization error. One of the main features of every FE method is that it tries to fit the exact solution by a set of polynomial functions locally defined, in the support of each node. Thus, in general as in the case of the geometry, it will not be possible to exactly represent

the analytical solution of the differential equation. We are only able to obtain a local element-wise polynomial approximation. The discretization error is, omitting other error sources, the difference between the exact solution of the mathematical model, \mathbf{u} , and its FE approximation, \mathbf{u}^h , the solution of the discrete problem.

Finally, when the discrete model is obtained, we need to evaluate a solution. During the solution process other types of error can show up, such as round off error, the errors associated to the numerical accuracy of the computer, etc. and also the prescribed tolerance when iterative solvers are used.

Generally, analysts consider that the error due to the FE discretization is small. However experience tells us that this is not always true. A clear example of this is the catastrophic collapse of the Sleipner offshore platform in 1991, with a cost of \$700 million [59], which sunk because the accuracy during the simulation was not enough. In this Chapter we will consider the discretization error as the main source of error, then all others will be considered as negligible.

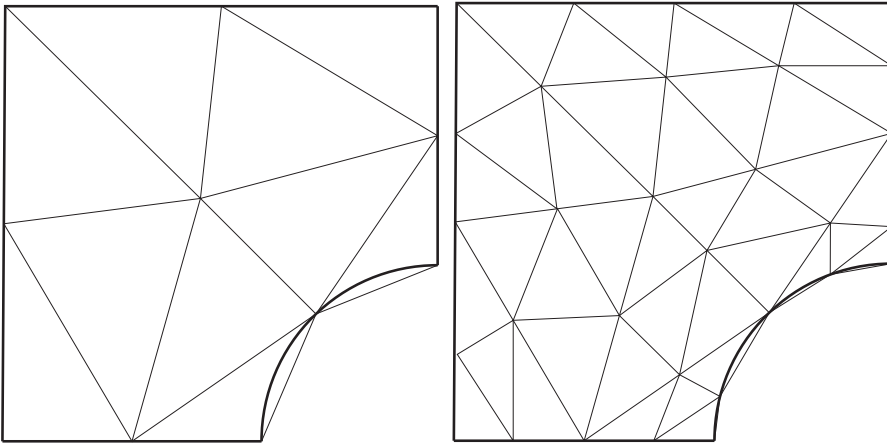


Figure 3.2: Different discretizations of a quarter of a square domain with a hole. Left, coarse discretization, right, finer element mesh.

3.1 Introduction and general review

FEM is a powerful method for a vast type of engineering problems, however it is only able to provide an approximated solution. Therefore, some *error* level has to be accounted for defining the safety factors during the design process of mechanical parts. During several years several types of error estimators techniques have been developed. It is important to classify them and to know their general properties. In general we can define three kinds of error estimators according to their convergence to the exact error [42]. The global effectivity index θ (which indicates the relation between the estimate and the true error) is an indicator of their convergence:

- *Asymptotically exact*: when we increase the richness of the discrete solution space, the estimated error gets closer (from above or below, or even oscillating) to the true one, then $\theta \rightarrow 1$ when $N \rightarrow \infty$.
- *Asymptotically effective*: when we increase the richness of the discrete solution space, the estimated error gets higher values than the true one, therefore $\theta \geq 1$ when $N \rightarrow \infty$.
- *Asymptotically "useless"*, as defined during the ECCOMAS 2012 congress during the oral presentation of [60]. When we increase the richness of the discrete solution space, the estimated error gets lower values than the true one, then $\theta \leq 1$ when $N \rightarrow \infty$.

As a consequence of this description, it is easy to observe that the best error estimator would be one which is both *asymptotically effective* and *asymptotically exact*.

Another way to classify them is according to the procedure used to obtain the estimates. Traditionally, there are four major branches in the error estimator field:

1. The *residual-based error estimators*.
2. The recovery type error estimators.
3. The Constitutive Relation Error-based error estimators.
4. Dual analysis.

The first branch, the *residual-based error estimators*, were introduced by Babuška and Rheinboldt [61]. This group may be subdivided into two more groups, the *explicit* and the *implicit* error estimators.

The explicit error estimators are based on the evaluation of the error by using the strong form of the residual equation [62]. Traditionally, they provide an upper bound up to a constant. In that sense, a great effort has been done during the last years by Stein's group at Hanover to develop a procedure to evaluate that constant [63].

The second subgroup, corresponding to implicit error estimators, is based on solving local problems by using the weak form of the residual equation. Local FE problems are solved in order to assess the global error of the FE solution. In this case, we do not need any constant to get an upper bound of the true error. This group is mostly used by mathematicians due to its well-known mathematical properties. In general, this method only guarantees the upper bound property when the local problems are solved minimizing their complementary energy. However, researchers usually solve local displacement-based problems in a richer space obtaining enough accuracy to maintain the upper bound property. Work in this area has been done for example by the groups of Oden [64] and Díez [65, 66] among others.

These two types of error estimators are *asymptotically effective*. That means they always yield upper bounds of the true error; however, the results could be rather conservative.

The second major branch, recovery type error estimators, is based on the use of the Zienkiewicz and Zhu (ZZ) error estimator [3]. In this case, these techniques were traditionally unable to provide upper error bounds. The key idea behind these techniques was to obtain a continuous recovered stress field and use it as a better approximation to the exact stress field than the raw FE solution. This recovered stress field can be used to obtain an estimation of the error in energy norm. The error evaluation is obtained by comparing the FE solution (compatible) with the recovered solution, not necessary equilibrated, but continuous, obtained with a recovery procedure such as, the Superconvergent Patch Recovery (SPR) technique [7, 8]. Reference [67] showed that if the recovered field used for the estimation is obtained with the SPR technique the error estimator is *asymptotically exact*. This kind of error estimators are robust, easy to implement, and are used in some commercial codes.

The publication of the original SPR technique was followed by several works aimed to improve its quality, see for example [68, 69, 70]. Ródenas *et al.* proposed to add constraints to impose local equilibrium and local compatibility to the recovered solution in the FEM framework [71] bringing up the SPR-C technique that was also adapted to the eXtended Finite Element Method (XFEM) framework [72]. The recovered field obtained with the SPR-C technique has a high accuracy and locally (at *patch* level) fulfills the equilibrium and compatibility equations. Díez *et al.* [73] presented a methodology to obtain computable upper bounds of the error in energy norm considering the quasi-equilibrated stress recovered field. This technique allowed us to obtain the first procedure to get practical upper error bounds for FEM and XFEM based on recovery techniques [73, 74] instead of using the traditional residual based error estimators [64]. We have to note that with SPR-based methods the upper bound property is not directly guaranteed because the recovered solution is not fully equilibrated. The upper bound property is obtained in these methods by adding correction terms for which only an estimation is available.

Regarding to the use of the SPR technique with Cartesian Grids, reference [11] indicates: ”*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 [75, 76, 77, 78, 79] and some on the SPR technique [80, 74], 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 [13, 11, 28, 29], the main drawback of the use of the techniques under the large umbrella of *finite elements in ambient space*, such as the cgFEM, 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 [71, 73, 80, 74] have been specially adapted both: i) to be used with the Zienkiewicz & Zhu error estimator [3] that will guide the h -adaptive refinement process to improve the quality of the solution, and ii) to neutralize the possible lack of accuracy along the boundaries in the cgFEM framework providing an enhanced solution (for which an error estimator will be presented in Chapter 6) that will be used instead of the FE solution.

In the third branch of error estimators we can place the Constitutive Relation Error (CRE) introduced by Ladevèze and Leguillon [81] and followed by several contributions for many applications, see for example [82, 83, 84, 85]. The CRE consist in evaluating a statically admissible stress field and compare it with a kinematically admissible stress field, directly providing upper error bounds of the error in energy norm. In general this kinematically admissible solution is the FE solution. For the statically admissible solution, a local problem is solved, which is built with the use of the strong prolongation condition.

Finally the fourth branch of error estimators is related with the concept of dual analysis, that makes use of two solutions. One of them is compatible whereas the other one is an equilibrated solution. This type of error estimators can also be used to directly obtain upper error bounds. Some of these error estimators

solve two global problems in parallel [86] whereas others post-process the FE solution [81, 87, 88]. The main characteristic of these error bounding techniques is that the error is evaluated by comparing the two solutions, which are complementary in nature, and whose errors are orthogonal, see [89]. Note that the CRE approach could also be included in this group since the final idea is to compare kinematically admissible solutions with statically admissible ones.

In this Chapter we will consider the FE solution of linear elasticity problems, where the Zienkiewicz and Zhu [3] (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 FE model by means of h -adaptive procedures. In this case, we have developed a SPR-type displacement-based recovery technique (SPR-CD) which we will use in the ZZ error estimator. The recovered stress field σ_σ^* is obtained as the continuous part of the stresses derived from a recovered displacement field \mathbf{u}_u^* , obtained with the SPR-CD technique which enforces static and kinematic admissibility constraints at patch level. The SPR-CD technique, considers the displacement and stress splitting into “singular” and “smooth” part, as in the SPR-CX technique [80] where stress splitting was also considered. Finally, during the recovery process we obtain a recovered pair $(\mathbf{u}_u^*, \sigma_\sigma^*)$ that will be used as output for the FE analysis. The subindex σ and u stands for the quasi- statical admissibility and kinematical admissibility of the recovered fields, respectively.

3.2 Error estimators in FEM

3.2.1 Explicit residual error estimator

This type of error estimators were initially introduced by Babuška and Miller [90] will be briefly described here. First we will introduce the main tools we used with his technique. The following well-known mathematical inequalities

will serve to obtain the error estimates in this method. Note that these will be useful when we introduce the explicit error estimator for the recovered solution in Chapter 6, following similar ideas.

- The Cauchy-Schwarz inequality:

$$\int_{\Omega} |\mathbf{u}^T \mathbf{v}| \, d\Omega \leq \|\mathbf{u}\|_{L^2(\Omega)} \|\mathbf{v}\|_{L^2(\Omega)} \quad \forall \mathbf{u}, \mathbf{v} \in L^2(\Omega) \quad (3.1)$$

- The Poincaré inequality, where the constant is defined by Chua and Wheeden [91]:

$$\left\| \mathbf{v} - \Upsilon^h \mathbf{v} \right\|_{L^2(\Omega)} \leq \frac{\text{diam}(\Omega)}{\pi} |\mathbf{v}|_{H^1(\Omega)} \quad \forall \mathbf{v} \in H^1(\Omega) \quad (3.2)$$

$$\|\mathbf{v}\|_{L^2(\Omega)} \leq \frac{\text{diam}(\Omega)}{\pi} |\mathbf{v}|_{H^1(\Omega)} \quad \forall \mathbf{v} \in V \quad (3.3)$$

where Υ^h is a projector that projects any vector in V to V^h , the discretized space.

- The Korn inequality:

$$|\mathbf{v}|_{H^1(\Omega)} \leq C \|\boldsymbol{\varepsilon}(\mathbf{v})\|_{L^2(\Omega)} \quad (3.4)$$

where C is a positive constant.

- The relation between the $L^2(\Omega)$ -norm of the strain field $\boldsymbol{\varepsilon}(\mathbf{v})$ and the energy norm [62]:

$$\|\boldsymbol{\varepsilon}(\mathbf{v})\|_{L^2(\Omega)} \leq \frac{1}{\sqrt{2\mu}} \|\mathbf{v}\|_{\Omega} \quad \forall \mathbf{v} \in V \quad (3.5)$$

where $\mu = \frac{E}{2(1+\nu)}$ is the Lamé constant with Young's modulus E and Poisson's ratio ν .

Let us define the residual equation, that is, the error in energy norm between both solutions, the exact one (living in V) and the discretized one (living in V^h).

$$R(\mathbf{v}) := a(\mathbf{e}, \mathbf{v}) = l(\mathbf{v}) - a(\mathbf{u}^h, \mathbf{v}) \quad \forall \mathbf{v} \in V \quad (3.6)$$

Solving (3.6) in $V^h \subset V$ would be one possibility to evaluate the error in the FE solution. Discretizing (3.6):

$$R(\mathbf{v}^h) = l(\mathbf{v}^h) - a(\mathbf{u}^h, \mathbf{v}^h) \quad \forall \mathbf{v}^h \in V^h \quad (3.7)$$

$$R(\mathbf{v}^h) = a(\mathbf{u}, \mathbf{v}^h) - a(\mathbf{u}^h, \mathbf{v}^h) = a(\mathbf{e}, \mathbf{v}^h) \quad (3.8)$$

where $a(\mathbf{e}, \mathbf{v}^h) = 0 \quad \forall \mathbf{v}^h \in V^h$, because of the Galerkin orthogonality, then $R(\mathbf{v}^h) = 0$ and the solution to that problem is trivial. This means that we need a richer space than V^h for solving (3.6) and to obtain an error measure, but the computational effort would be unaffordable since it would require to solve a global problem in a richer space.

Let us investigate some alternatives. Consider the regularly discretized mesh \mathcal{T} , formed by bilinear or bicuadratic squared elements, $\cup K = \mathcal{T}$, where K indicates each element. Let \mathcal{N} indicate the set of nodes of the mesh. As indicated in [62], it is possible to define a quasi-interpolation operator $\Upsilon^h : V \rightarrow V^h$, such that for every $\mathbf{v} \in V$ we can obtain $\Upsilon^h \mathbf{v} = \mathbf{v}^h \in V^h$. Then, we can rewrite (3.7) as:

$$0 = l(\Upsilon^h \mathbf{v}) - a(\mathbf{u}^h, \Upsilon^h \mathbf{v}) \quad \forall \mathbf{v} \in V \quad (3.9)$$

Now we introduce an example of the quasi-interpolation operator. Consider a node $i \in \mathcal{N}$. We define the “local” quantity related with node i as:

$$\mathbf{v}_i(\mathbf{v}) := \frac{1}{|\Omega_i^*|} \int_{|\Omega_i^*|} \mathbf{v} \, d\Omega \quad (3.10)$$

where Ω_i^* could be any area associated to the node i , for instance the patch of elements connected to the node i . The quasi-interpolation operator is defined as follows:

$$\Upsilon^h \mathbf{v} := \sum_i^{\mathcal{N}} N_i \mathbf{v}_i(\mathbf{v}) \quad (3.11)$$

Regarding to the residual equation (3.6), subtracting (3.9) and discretizing we rewrite (3.6) as follows:

$$\begin{aligned}
 R(\mathbf{v}) = & \sum_{K \in \mathcal{T}} \left(\int_{\Omega^K} \mathbf{b}^T(\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Omega + \int_{\Gamma_N \cap \partial K} \mathbf{t}^T(\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Gamma \right) \\
 & - \sum_{K \in \mathcal{T}} \int_{\Omega^K} \boldsymbol{\sigma}(\mathbf{u}^h)^T \boldsymbol{\varepsilon}(\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Omega \quad \forall \mathbf{v} \in V \quad (3.12)
 \end{aligned}$$

and integrating the second summation by parts,

$$\begin{aligned}
 R(\mathbf{v}) = & \sum_{K \in \mathcal{T}} \left(\int_{\Omega^K} \mathbf{b}^T(\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Omega + \int_{\Gamma_N \cap \partial \Omega^K} \mathbf{t}^T(\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Gamma \right) \\
 & + \sum_{K \in \mathcal{T}} \left(\int_{\Omega^K} (\mathbf{L}^T \boldsymbol{\sigma}(\mathbf{u}^h))^T (\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Omega - \oint_{\partial \Omega^K} (\mathbf{G} \boldsymbol{\sigma}(\mathbf{u}^h))^T (\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Gamma \right) \\
 & \quad \forall \mathbf{v} \in V \quad (3.13)
 \end{aligned}$$

Regrouping terms, it is possible to write the following expression:

$$\begin{aligned}
 R(\mathbf{v}) = & \sum_{K \in \mathcal{T}} \int_{\Omega^K} (\mathbf{b} + \mathbf{L}^T \boldsymbol{\sigma}(\mathbf{u}^h))^T (\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Omega + \\
 & \int_{\Gamma_N \cap \partial \Omega^K} (\mathbf{t} - \mathbf{G} \boldsymbol{\sigma}(\mathbf{u}^h))^T (\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Gamma - \int_{\partial \Omega^K \setminus \Gamma_N} (\mathbf{G} \boldsymbol{\sigma}(\mathbf{u}^h))^T (\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Gamma \\
 & \quad \forall \mathbf{v} \in V \quad (3.14)
 \end{aligned}$$

Let us analyze these terms at element level. Let be E_l the element side and J and K two adjacent elements. Let us define $\mathbf{r}_{E_l}^h$ as:

$$\mathbf{r}_{E_l}^h := - \begin{cases} \frac{1}{2} \mathbf{G}_{E_l}(\boldsymbol{\sigma}(\mathbf{u}^h)_K - \boldsymbol{\sigma}(\mathbf{u}^h)_J) & \text{if } E_l = \bar{K} \cap \bar{J} \\ \mathbf{t} - \mathbf{G}_{E_l} \boldsymbol{\sigma}(\mathbf{u}^h) & \text{if } E_l \subset \Gamma_N \\ \mathbf{0} & \text{if } E_l \subset \Gamma_D \end{cases} \quad (3.15)$$

where \mathbf{G}_{E_l} is the stress field outward projector over each side E_l of element K . Note that the outward projector of the element J of the common edge

with K is just $-\mathbf{G}_{E_l}$. Then, (3.14) could be condensed as follows:

$$R(\mathbf{v}) = \sum_{K \in \mathcal{T}} \left(\int_{\Omega^K} (\mathbf{b} + \mathbf{L}^T \boldsymbol{\sigma}(\mathbf{u}^h))^T (\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Omega - \sum_{l=1}^L \int_{E_l \subset \partial\Omega^K} (\mathbf{r}_{E_l}^h)^T (\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Gamma \right) \quad \forall \mathbf{v} \in V \quad (3.16)$$

If we take these integrals at element level, we could bound them by using the Cauchy-Schwarz inequality. Renaming $-\mathbf{s}^h = \mathbf{b} + \mathbf{L}^T \boldsymbol{\sigma}(\mathbf{u}^h)$, the lack of internal equilibrium and $\mathbf{r}_{E_l}^h$ the lack boundary equilibrium of the FE solution.

$$\begin{aligned} - \int_{\Omega^K} (\mathbf{s}^h)^T (\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Omega &\leq \left\| \mathbf{s}^h \right\|_{L^2(K)} \left\| \mathbf{v} - \Upsilon^h \mathbf{v} \right\|_{L^2(K)} \\ - \int_{E_l \subset \partial\Omega^K} (\mathbf{r}_{E_l}^h)^T (\mathbf{v} - \Upsilon^h \mathbf{v}) \, d\Gamma &\leq \left\| \mathbf{r}_{E_l}^h \right\|_{L^2(E_l)} \left\| \mathbf{v} - \Upsilon^h \mathbf{v} \right\|_{L^2(E_l)} \end{aligned} \quad (3.17)$$

The final step consists in bounding the unknown terms $\left\| \mathbf{v} - \Upsilon^h \mathbf{v} \right\|_{L^2(K)}$ and $\left\| \mathbf{v} - \Upsilon^h \mathbf{v} \right\|_{L^2(E_l)}$. From [62] the following inequalities could be proven:

$$\begin{aligned} \left\| \mathbf{v} - \Upsilon^h \mathbf{v} \right\|_{L^2(K)} &\leq C_1 h_K \left\| \boldsymbol{\varepsilon}(\mathbf{v}) \right\|_{L^2(K^*)} \\ \left\| \mathbf{v} - \Upsilon^h \mathbf{v} \right\|_{L^2(E_l)} &\leq C_2 h_K^{\frac{1}{2}} \left\| \boldsymbol{\varepsilon}(\mathbf{v}) \right\|_{L^2(K^*)} \end{aligned} \quad (3.18)$$

Using the inequalities in (3.18) and defining $C' = \max(C_1, C_2)$ we can rewrite (3.16) as follows:

$$\begin{aligned} R(\mathbf{v}) &\leq \sum_{K \in \mathcal{T}} C' h_K \left\| \mathbf{s}^h \right\|_{L^2(K)} \left\| \boldsymbol{\varepsilon}(\mathbf{v}) \right\|_{L^2(K^*)} \\ &\quad + \sum_{K \in \mathcal{T}} C' h_K^{\frac{1}{2}} \sum_l^L \left\| \mathbf{r}_{E_l}^h \right\|_{L^2(E_l)} \left\| \boldsymbol{\varepsilon}(\mathbf{v}) \right\|_{L^2(K^*)} \end{aligned} \quad (3.19)$$

extracting C' as common factor and applying the Hölder inequality, $\sum_{j=1}^n a_j b_j \leq \left\{ \sum_{j=1}^n a_j^p \sum_{j=1}^n b_j^p \right\}^{\frac{1}{p}}$.

$$\begin{aligned}
 R(\mathbf{v}) \leq C' & \left\{ \sum_{K \in \mathcal{T}} \|\boldsymbol{\varepsilon}(\mathbf{v})\|_{L^2(\Omega^*)}^2 \right\}^{\frac{1}{2}} \left(\left\{ \sum_{K \in \mathcal{T}} h_K^2 \|\mathbf{s}^h\|_{L^2(\Omega^K)}^2 \right\}^{\frac{1}{2}} \right. \\
 & \left. + \left\{ \sum_{K \in \mathcal{T}} h_K \sum_l^L \|\mathbf{r}_{E_l}^h\|_{L^2(E_l)}^2 \right\}^{\frac{1}{2}} \right)
 \end{aligned} \tag{3.20}$$

applying again the Hölder inequality, in this case with $n = 2$:

$$R(\mathbf{v}) \leq C \|\boldsymbol{\varepsilon}(\mathbf{v})\|_{L^2(\Omega)} \left\{ \sum_{K \in \mathcal{T}} \left(h_K^2 \|\mathbf{s}^h\|_{L^2(\Omega^K)}^2 + h_K \sum_l^L \|\mathbf{r}_{E_l}^h\|_{L^2(E_l)}^2 \right) \right\}^{\frac{1}{2}} \tag{3.21}$$

where $C = \sqrt{2}C'$. Finally considering (3.5) we obtain the explicit error estimator:

$$R(\mathbf{v}) \leq \frac{C}{\sqrt{2\mu}} \|\mathbf{v}\|_{\Omega} \left\{ \sum_{K \in \mathcal{T}} \left(h_K^2 \|\mathbf{s}^h\|_{L^2(\Omega^K)}^2 + h_K \sum_{l=1}^L \|\mathbf{r}_{E_l}^h\|_{L^2(E_l)}^2 \right) \right\}^{\frac{1}{2}} \tag{3.22}$$

Now, substituting \mathbf{v} by \mathbf{e} we obtain an estimate for the discretization error in energy norm, up to a constant C , depending only on the mesh type.

3.2.2 Implicit residual error estimator

The implicit residual error estimators are also based on the residual problem (3.6). The difference from the previous method is that in this case we estimate the weak form of the residual. In the previous case the error was estimated by directly measuring the energy due to the lack of equilibrium of the FE solution. The previous method depends on an unknown constant C whereas

¹Note that the Hölder inequality when $p = 2$ coincides with the Cauchy-Schwartz inequality

the implicit residual error estimator is constant free. This error estimator is based on solving the residual equation (3.23) into each element $K \in \mathcal{T}$. Thus the global error is obtained by summing up all element contributions.

$$R(\mathbf{v}) := a(\mathbf{e}, \mathbf{v}) = \sum_{K \in \mathcal{T}} a_K(\mathbf{e}, \mathbf{v}) \quad \forall \mathbf{v} \in V \quad (3.23)$$

where $a_K(\mathbf{e}, \mathbf{v})$ represents the bilinear form extended only over the area of a single element K . In Figure 3.3 we represent the local problem over the element $K \in \mathcal{T}$ where \mathbf{t}_L are the tractions that define the local problem. These tractions should be in equilibrium with the other applied loads. There are several ways to obtain the solution of these local problems at each element. According to some authors [43], the most robust formulation to solve these local problems is the one based on the equilibrated residual over the element.

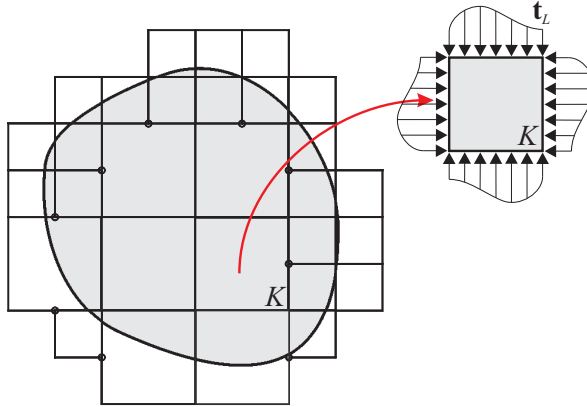


Figure 3.3: Local problem at element $K \in \mathcal{T}$. The traction \mathbf{t}_L represent the equilibrated traction obtained from a FE post-processing technique.

In order to give a better comprehension of these local problems we decompose (3.23) into the following integrals:

$$\begin{aligned} a_K(\mathbf{e}, \mathbf{v}) = & - \int_{\Omega^K} (\boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} \boldsymbol{\sigma}(\mathbf{v}) \, d\Omega + \int_{\Omega^K} \mathbf{b}^T \mathbf{v} \, d\Omega \\ & + \int_{\partial\Omega^K \setminus \Gamma_N} \mathbf{t}_L^T \mathbf{v} \, d\Gamma + \int_{\partial\Omega^K \cup \Gamma_N} (\mathbf{G}\boldsymbol{\sigma}(\mathbf{u}))^T \mathbf{v} \, d\Gamma \quad \forall \mathbf{v} \in V \end{aligned} \quad (3.24)$$

However, the traction field \mathbf{t}_L over $\partial K \setminus \Gamma_N$ are not know. In order to be able to solve these local problems, we need to evaluate an auxiliary traction field $\tilde{\mathbf{t}}_L$ from the available data, that is the FE solution and the problem loads. Furthermore, $\tilde{\mathbf{t}}_L$ must be in equilibrium with the other loads of the local problem (body forces, Neumann boundary conditions, etc.) to ensure the solvability of the local problems. Different methods [81, 83, 92] have been used to evaluate auxiliary traction fields leading to different implicit residual type error estimators. Other approach, not requiring an equilibrated traction field, was introduced by Bank and Weiser [93] and followed by Díez *et. al.* [65]. This last approach evaluates a consistent traction field, that is a continuous traction field (a nodal averaging could be used). Then, to ensure the solvability of the local problems, they restrict the set of admissible functions, in the local problem, eliminating the kernel of the l.h.s of equation (3.24) from the local interpolation space. Following the most common approach, introducing the auxiliary tractions $\tilde{\mathbf{t}}_L$, the local problem is reformulated as follows:

$$\begin{aligned}
 a_K(\boldsymbol{\phi}, \mathbf{v}) = & - \int_{\Omega^K} (\boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} \boldsymbol{\sigma}(\mathbf{v}) \, d\Omega + \int_{\Omega^K} \mathbf{b}^T \mathbf{v} \, d\Omega \\
 & + \int_{\partial\Omega^K \setminus \Gamma_N} \tilde{\mathbf{t}}_L^T \mathbf{v} \, d\Gamma + \int_{\partial\Omega^K \cap \Gamma_N} (\mathbf{G}\boldsymbol{\sigma}(\mathbf{u}))^T \mathbf{v} \, d\Gamma \quad \forall \mathbf{v} \in V
 \end{aligned} \tag{3.25}$$

where $\boldsymbol{\phi}$ is the solution of the local problem (3.25) living in $V^b(\Omega)$ so-called the broken space, richer than V . Summing up all these local problem satisfies that $a(\boldsymbol{\phi}, \mathbf{v}) = a(\mathbf{e}, \mathbf{v}) \, \forall \mathbf{v} \in V$ because the global FE problem is self-equilibrated for a given discretization. Considering the particular case when $\mathbf{v} = \mathbf{e}$ we can write:

$$\begin{aligned}
 0 \leq a(\boldsymbol{\phi} - \mathbf{e}, \boldsymbol{\phi} - \mathbf{e}) & = a(\boldsymbol{\phi}, \boldsymbol{\phi}) - 2a(\boldsymbol{\phi}, \mathbf{e}) + a(\mathbf{e}, \mathbf{e}) \\
 |||\mathbf{e}|||_\Omega & \leq |||\boldsymbol{\phi}|||_\Omega
 \end{aligned} \tag{3.26}$$

proving the upper bound property.

Unfortunately, the local problem (3.25), in general, can not be solved exactly. So we need to use a numerical method to obtain a solution. If we are interested in obtaining upper bounds, we should solve this problem with the stress-based

FEM, however this is cumbersome and it is not used in practice. Some authors [94, 95] use the standard displacement based FEM with a richer space, for instance $p + 3$ [95] where p is the interpolation degree of the FE solution \mathbf{u}^h .

Flux free implicit methods

The evaluation of the equilibrated traction of the local problems is cumbersome and requires a high computational effort. Parés *et. al.* [96] introduced a new variant of the local problems, using the partition of unity approach. In this case the local problem is extended to a star or patch of elements surrounding a node, as shown in Figure 3.4. The shape functions are partition of unity so we can rewrite (3.23) as follows:

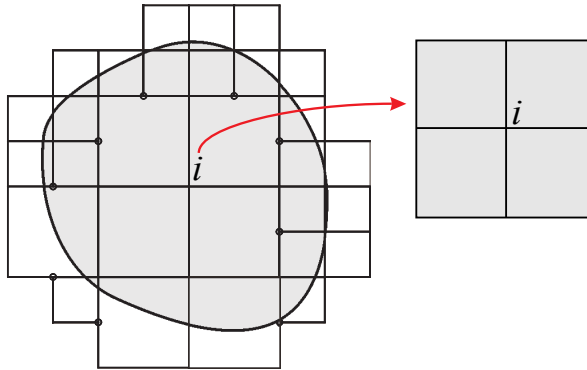


Figure 3.4: Local problem at element $K \in \mathcal{T}$. The tractions \mathbf{t}_L represents the equilibrated tractions obtained from a FE post-processing technique.

$$R(\mathbf{v}) = R\left(\sum_{i \in \mathcal{N}} N_i^1 \mathbf{v}\right) = \sum_{i \in \mathcal{N}} R(N_i^1 \mathbf{v}) \quad \forall \mathbf{v} \in V \quad (3.27)$$

where N_i^1 is the linear (or bilinear) shape function associated to the node i of the mesh. As you can perceive the global residual $R(\mathbf{v})$ could be evaluated by summing up all local contributions evaluated at each patch i , $R(N_i^1 \mathbf{v})$. Because of the use of the partition of the unity, the traction along the bound-

ary of the patch vanishes, then the evaluation of the traction filed along the boundary of the local problem (3.28) is no longer needed.

$$a_i(\bar{\phi}^i, \mathbf{v}) = R(N_i^1(\mathbf{v} - \mathbf{\Upsilon}^h \mathbf{v})) \quad \forall \mathbf{v} \in V \quad (3.28)$$

Finally, the global error could be evaluated by adding all contributions from the local problems.

3.2.3 Constitutive Relation Equation (CRE) based error estimators

Error estimators based on the violation of the constitutive law were introduced by Ladevèze *et. al* [81]. The authors use the so-called constitutive relation error $\mathcal{E}_{CRE}^2 = \bar{a}(\check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}^h, \check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}^h)$ that provides a guaranteed upper bound of the exact error $\|\mathbf{e}\|_{\Omega}$ [82], where $\bar{a}(\cdot, \cdot)$ is the stress representation of the standard FE bilinear form, $\check{\boldsymbol{\sigma}}$ is a statically admissible stress field and $\boldsymbol{\sigma}^h$ is a kinematically admissible field. Not that in this case we have followed the most common approach that consist in using the FE stress solution as the kinematically admissible stress field. In this case, the *recovered* stress field $\check{\boldsymbol{\sigma}}$ is obtained making use, at each element, of the strong prolongation condition which allows to obtain a set of equilibrated traction over the edges of each element $\check{\mathbf{t}}_L$. Once these traction have been obtained, a local high order FE problem has to be solved for each element to obtain the stress field $\check{\boldsymbol{\sigma}}$ in the bulk of the element. The error is evaluated using the following expression, similar to the ZZ-type error estimator:

$$\mathcal{E}_{CRE}^2 = \int_{\Omega} (\check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} (\check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}^h) \, d\Omega \quad (3.29)$$

Note that this method is also similar to the standard implicit residual methods. The main difference is that the implicit residual methods directly provide the error at each element while in the CRE method provides a statically admissible stress field at each element that will be used to evaluate the local error.

3.2.4 Mixed formulation error estimators

The well-known Prager-Synge inequality states that, in terms of energy, the difference between a kinematically stress field and a statically stress field is an upper bound of the true error in energy of those solutions. The FE problem provides a stress field that is kinematically admissible, so we only need to obtain a statically admissible stress field to compute an upper bound of the FE discretization error. However to obtain a statically admissible stress field we should solve a stress-based FE problem, minimizing the complementary energy. This would be as expensive as solving the displacement-based FE problem.

Another option to obtain the statically admissible stress field could be obtained by locally post processing the displacement-based FE solution, as in the CRE method, or by using a mixed formulation for the global problem. This second method, introduced by [97], is usually the most accurate [82] because minimizes both the deformation energy and the complementary energy, however requires a higher computational effort. In this case, the system solves both formulations together, the displacement-based and stress-based formulations. Some authors [86, 89, 88] maintain that despite the higher computational cost, these mixed formulations, which directly provide kinematically and statically admissible solutions, have important advantages in quality of the solution and in the possibility to directly obtain very accurate error bounds in energy norm.

3.2.5 Recovery-based error estimators. The SPR technique

Recovery-based error estimators are based on the Zienkiewicz and Zhu (ZZ) error estimator (2.28) [3], presented in Section 2.4. In this case, instead of trying to achieve the upper error bound property (*asymptotically effective*) the method only focuses on the quality of the estimate (*asymptotically exact*). There are several recovery procedures in literature from which we can high-

light the *nodal averaging* technique which is the simplest one. The recovered stress field provided by this technique is obtained by a nodal representation of the stress field. The value assigned at each node is obtained as an average of the stress value of the elements connected to the node. The FE shape functions are used to interpolate the recovered stress field. This technique is extremely simple and allows for fast calculations and provides acceptable results for linear elements. However, for quadratic elements, it does not provide good error estimations since the effectivity index does not converge ($\theta \rightarrow 1$) thus this procedure is *asymptotically "useless"*. The Super-convergent Patch Recovery (SPR) technique introduced by Zienkiewicz and Zhu [7, 8] provides a recovered stress field of a better quality providing *asymptotically effective* error estimates for practical situations.

The SPR technique is widely used to obtain the improved stress field $\boldsymbol{\sigma}^*$ used in the ZZ error estimator (2.28). References [98, 99, 100] show that the SPR is the most robust technique used for error estimation on problems with smooth solutions approximated on patch-wise uniform grids, for linear or quadratic elements. This technique, first defines a patch of elements \mathcal{P}^i , that is a set of elements sharing a vertex node $i \in \mathcal{N}$, this node is also called the patch assembly node, see Figure 3.5.

A polynomial surface, as shown in Figure 3.6, per component (3.30) (of the same degree as the FE interpolation) is fitted to the FE stress values at the super-convergent points of the patch by using a least square approach:

$$\hat{\sigma}_k^*(\mathbf{x}) = \mathbf{p}(\mathbf{x})\mathbf{a}_k \quad k = xx, yy, xy \quad (3.30)$$

where $\mathbf{p}(\mathbf{x}) = \{1, x, y\}$ for the linear case, $\mathbf{p}(\mathbf{x}) = \{1, x, y, x^2, xy, y^2\}$ for the quadratic case, and \mathbf{a}_k are the corresponding coefficients for each stress component. In this case, each component k of the stress field could be recovered independently by minimizing the following functional:

$$\Phi_{SPR} = \sum_{gp}^{NGP} (\mathbf{p}(\mathbf{x}_{gp})\mathbf{a}_k - \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}^h(\mathbf{x}_{gp}))|_k)^2 \quad (3.31)$$

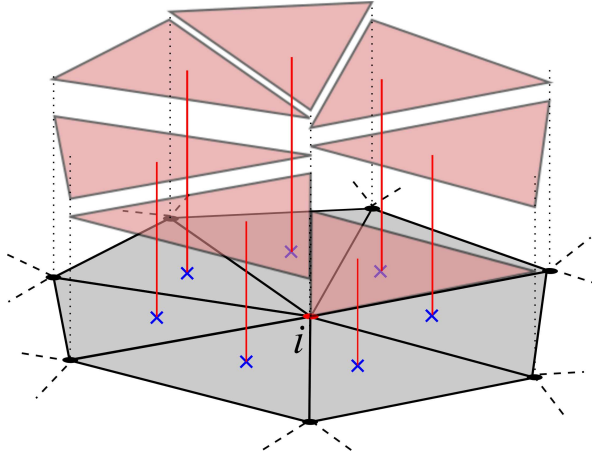


Figure 3.5: Representation of a patch of linear triangular elements. The black points indicates the nodes of the mesh and the red node is the patch assembly node. The transparent surfaces indicate the FE stress field σ^h . The super-convergent points are indicated by blue crosses.

yielding a linear system of equations per component $\mathbf{M}\mathbf{a}_k = \mathbf{H}_k$, where NGP indicates the number of integration (sample) points and:

$$\begin{aligned} \mathbf{M} &= \sum_l^{NGP} \mathbf{p}^T(\mathbf{x}_l)\mathbf{p}(\mathbf{x}_l) \\ \mathbf{H}_k &= \sum_l^{NGP} \mathbf{p}^T(\mathbf{x}_l)\mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}^h(\mathbf{x}_l))|_k. \end{aligned} \quad (3.32)$$

The recovered stresses $\hat{\boldsymbol{\sigma}}_i^*$ at each node are obtained particularizing these surfaces at the assembly node. Finally, following the same process at each assembly node of the mesh we end up with a nodal stress representation. Note that this process is computationally efficient as it only requires to solve small systems of equations to obtain the recovered field. The nodal values into each element are interpolated using the FE shape functions N_i , according to (3.33), see Figure 3.7.

$$\boldsymbol{\sigma}^*(x, y)_{\text{SPR}} = \sum_i N_i(\mathbf{x})\hat{\boldsymbol{\sigma}}_i^* \quad (3.33)$$

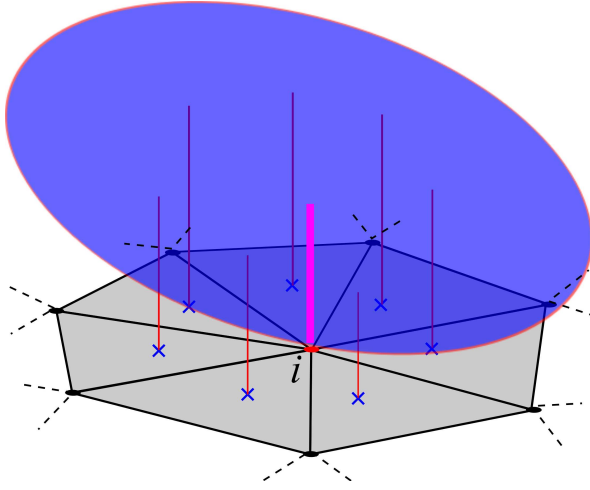


Figure 3.6: Representation of the least squares fitted polynomial surface. The pink line represents the stress value at the assembly node, the only one that is retained in the standard SPR.

where N_i is the shape function associated to $i \in \mathcal{N}$ and $\hat{\sigma}_i^*$ is the corresponding recovered nodal stress value.

3.2.6 A nearly equilibrated recovery procedure. The SPR-C technique

Since the introduction of the plain SPR technique [8] we can find several contributions aimed at improving the quality and the robustness of this technique. In general they couple the stress component equations in order to be able to add constraints that improve the quality of the recovered field. Wiberg and Abdulwahab [101, 68] proposed to take into account the equilibrium of the recovered field by using a penalty method, reporting considerably better results in comparison with the standard SPR. Blacker and Belytschko [69], however reported that by only using the enforcement of the internal equilibrium equation and the natural condition does not always improve the convergence

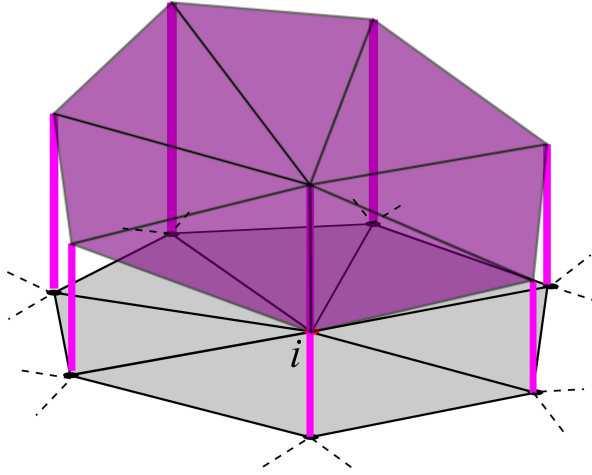


Figure 3.7: Representation of the final recovered stress field σ^* over the problem domain. The nodal recovered values are interpolated by the FE shape functions.

rate of the recovered field. However when they use the “Conjoint Polynomial Enhancement”, introduced in that contribution, they always found the appropriate behavior of the recovered solution, also along the boundaries. Looking for a technique to obtain upper error bounds, Aalto *et al.* [102, 103, 104] proposed the use of self-equilibrated polynomial basis to represent the recovered solution. Kvamsdal and Okstad [105] introduced a recovery process including equilibration techniques and the Conjoint Polynomial enhancement. They consider that the lack of internal equilibrium of their recovered field was negligible. Although their technique is not SPR-based, Boroomand and Zienkiewicz [106, 107] presented a technique, the so-called Recovery Equilibrium in patches (REP), that tries to obtain equilibrated recovered stress fields by using the equilibrium information provided by the FE analysis.

More recently, Ródenas and coworkers introduced the so-called SPR-C technique [71], where the “C” stands for constraints. The SPR-C technique was also applied in the XFEM context by Ródenas *et al.* [74] and finally adapted to geometry-mesh independent FE formulations [108], such as the cgFEM. This technique has some differences with respect to the previous improvements. As

in the SPR technique, a patch \mathcal{P}^i is defined as the set of elements connected to a vertex node i . On each patch, a polynomial expansion for each one of the components of the recovered stress field is expressed according to (3.30). In this case they need to simultaneously consider all the components of the stress vector to be able to include the required constrain equations. Thus, in the SPR-C, the recovered stress field for the 2D case, for each patch, reads:

$$\hat{\boldsymbol{\sigma}}_i^*(\mathbf{x}) = \begin{Bmatrix} \hat{\sigma}_{xx}^*(\mathbf{x}) \\ \hat{\sigma}_{yy}^*(\mathbf{x}) \\ \hat{\sigma}_{xy}^*(\mathbf{x}) \end{Bmatrix} = \mathbf{P}(\mathbf{x})\mathbf{A} = \begin{bmatrix} \mathbf{p}(\mathbf{x}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{p}(\mathbf{x}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{p}(\mathbf{x}) \end{bmatrix} \begin{Bmatrix} \mathbf{a}_{xx} \\ \mathbf{a}_{yy} \\ \mathbf{a}_{xy} \end{Bmatrix} \quad (3.34)$$

To obtain the stress field coefficients \mathbf{A} , the L_2 -norm of the following functional is minimized:

$$\Phi'(\mathbf{A}) := \int_{\mathcal{P}^i} (\mathbf{P}(\mathbf{x})\mathbf{A} - \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}^h(\mathbf{x})))^2 d\Omega \quad (3.35)$$

Resulting in a linear system of equations to solve at each patch \mathcal{P}^i :

$$\mathbf{M}\mathbf{A} = \mathbf{H} \quad (3.36)$$

where after numerical integration \mathbf{M} and \mathbf{H} are:

$$\begin{aligned} \mathbf{M} &= \sum_l^{NGP} \mathbf{P}(\mathbf{x}_l)^T \mathbf{P}(\mathbf{x}_l) \omega_l |\mathbf{J}(\mathbf{x}_l)| \\ \mathbf{H} &= \sum_l^{NGP} \mathbf{P}(\mathbf{x}_l)^T \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}^h(\mathbf{x}_l)) \omega_l |\mathbf{J}(\mathbf{x}_l)| \end{aligned} \quad (3.37)$$

where NGP is the number of integration points (not necessarily super-convergent), ω_l is the corresponding weight, \mathbf{x}_l is the integration point into the patch and $\mathbf{J}(\mathbf{x}_l)$ the the Jacobian matrix of the coordinates transformation used for numerical integration.

As a difference from standard smoothing techniques, the SPR-C technique uses a continuous approach (3.37) to take into account the different densities of sampling points in patches placed in the vicinity of the boundaries. The main difference between the continuous approach and discrete approach (3.32) is that in the continuous approach the values at sampling points are weighted by

their corresponding area ($\omega_l |\mathbf{J}(\mathbf{x}_l)|$), whereas in discrete approach all sampling points have the same weight. The results have shown that the continuous approach performs better than the discrete one in the cgFEM.

The SPR-C technique uses constrain equations to consider the known information of the linear elasticity problem during the recovery process. Lagrange multipliers are used to consider the satisfaction of the internal equilibrium equation (int), boundary equilibrium equation (ext) and compatibility equation (cmp), when evaluating coefficients \mathbf{A} . The constrain equations to be considered are described below.

- Internal equilibrium equation: the constraint equation for the internal equilibrium in the patch is defined as:

$$c^{\text{int}}(\mathbf{x}_j) : \mathbf{L}^T \hat{\boldsymbol{\sigma}}_i^*(\mathbf{x}_j) + \mathbf{L}^T (\boldsymbol{\sigma}_0(\mathbf{x}_j) - \mathbf{D}\boldsymbol{\varepsilon}_0(\mathbf{x}_j)) + \hat{\mathbf{b}}(\mathbf{x}_j) = 0 \quad \forall \mathbf{x}_j \in \mathcal{P}^i \quad (3.38)$$

$\hat{\mathbf{b}}(\mathbf{x})$ is a polynomial least squares fitting of degree $p - 1$, being p the degree of the recovered stress field $\hat{\boldsymbol{\sigma}}^*$, to the actual body forces $\mathbf{b}(\mathbf{x})$. Note that this approximation will allow to have $\hat{\mathbf{b}}(\mathbf{x}) = \mathbf{b}(\mathbf{x})$ in the vast majority of practical cases. The approximation will be necessary if $\mathbf{b}(\mathbf{x})$ is too complex to be represented by $\hat{\boldsymbol{\sigma}}^*$. $c^{\text{int}}(\mathbf{x}_j)$ is enforced in a sufficient number of non-aligned points (*nice*) to guarantee the exact representation of $\hat{\mathbf{b}}(\mathbf{x})$.

- Boundary equilibrium equation: the constraint equation reads:

$$c^{\text{ext}}(\mathbf{x}_j) : \mathbf{G}\hat{\boldsymbol{\sigma}}_i^*(\mathbf{x}_j) + \mathbf{G}(\boldsymbol{\sigma}_0(\mathbf{x}_j) - \mathbf{D}\boldsymbol{\varepsilon}_0(\mathbf{x}_j)) = \mathbf{t}(\mathbf{x}_j) \quad \forall \mathbf{x}_j \in \Gamma_N \cap \mathcal{P}^i \quad (3.39)$$

$c^{\text{ext}}(\mathbf{x}_j)$ is enforced in $p + 1 = nbee$ points along $\Gamma_N \cap \mathcal{P}^i$. In the case where more than one boundary is intersecting the patch, only one curve is considered in order to avoid over-constraining issues.

- Compatibility equation: $c^{\text{cmp}}(\mathbf{x}_j)$ is only imposed in the case that $p \geq 2$ in a sufficient number of non-aligned points *nc*. For example, for $p = 2$ we

have $nc = 1$. $\hat{\boldsymbol{\sigma}}^*$ directly satisfies \mathbf{c}^{cmp} for $p = 1$. The 2D compatibility equation expressed in terms of stresses, (see [109]) is:

$$\begin{aligned} \mathbf{c}^{cmp}(\mathbf{x}_j) : \frac{\partial^2}{\partial y^2} (k\hat{\sigma}_{xx}(\mathbf{x}_j) - \nu q\hat{\sigma}_{yy}(\mathbf{x}_j)) + \frac{\partial^2}{\partial x^2} (k\hat{\sigma}_{yy}(\mathbf{x}_j) - \nu q\hat{\sigma}_{xx}(\mathbf{x}_j)) - \\ 2(1 + \nu) \frac{\partial^2 \hat{\sigma}_{xy}(\mathbf{x}_j)}{\partial x \partial y} = 0 \quad \forall \mathbf{x}_j \in \mathcal{P}^i \end{aligned} \quad (3.40)$$

where k, q are functions of the Poisson's coefficient ν

$$\begin{cases} k = 1, q = 1 & \text{for plane stress} \\ k = (1 - \nu)^2, q = (1 + \nu) & \text{for plane strain} \end{cases}$$

Thus, the functional to be optimized considering the constraint equations for a patch \mathcal{P}^i can be written as:

$$\begin{aligned} \Phi(\mathbf{A}, \boldsymbol{\lambda}) := \Phi'(\mathbf{A}) + \\ \sum_j^{n_{bee}} \lambda_j^{\text{int}} (c^{\text{int}}(\mathbf{x}_j)) + \sum_j^{n_{iee}} \lambda_j^{\text{ext}} (c^{\text{ext}}(\mathbf{x}_j)) + \sum_j^{n_{cee}} \lambda_j^{\text{cmp}} (c^{\text{cmp}}(\mathbf{x}_j)) \end{aligned} \quad (3.41)$$

yielding a modified system of equations to solve at each path:

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}^T \\ \mathbf{C}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{A} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{H} \\ \boldsymbol{\Lambda} \end{Bmatrix} \quad (3.42)$$

where \mathbf{C} represents the constraint equations, \mathbf{A} are the coefficients of the polynomial expansion and $\boldsymbol{\Lambda}$ is the r.h.s. of the constraint equations. To obtain a continuous field, a partition of unity procedure (the conjoint polynomial enhancement [69]) properly weighting the stress interpolation polynomials, obtained from patches corresponding to each of the vertex nodes of the element containing point \mathbf{x} , is used. The field $\boldsymbol{\sigma}_{\text{SPR-C}}^*$ is interpolated using linear shape functions N_i associated with the n_v vertex nodes such that:

$$\boldsymbol{\sigma}_{\text{SPR-C}}^*(\mathbf{x}) = \sum_{J=1}^{n_v} N_J(\mathbf{x}) \hat{\boldsymbol{\sigma}}_J^*(\mathbf{x}) - \mathbf{D}\boldsymbol{\varepsilon}_0(\mathbf{x}) + \boldsymbol{\sigma}_0(\mathbf{x}) \quad (3.43)$$

3.3 Nearly equilibrated displacement recovery procedure. The SPR-CD technique

In this work we propose to obtain a recovered displacement field from the FE pair $(\mathbf{u}^h, \boldsymbol{\sigma}^h)$, instead of a recovered stress field as in the SPR-C technique. The proposed method, denoted as SPR-CD (where C stands for constraints and D for displacements), is more complete in the sense that it is able to provide an improved recovered pair $(\mathbf{u}_u^*, \boldsymbol{\sigma}_\sigma^*)^2$. To evaluate the displacement recovered field, some authors such as Tabbara *et. al.* [110] propose the use of a Moving Least Squares (MLS) technique. These authors reported a higher accuracy than the standard SPR technique for the recovered fields. More recently Aalto *et. al* [102, 103, 104] also proposed to recover the displacement field, adding some equilibrium information to directly obtain a continuous recovered stress field. Wiberg and Abdulwahab [111] proposed also to obtain a recovered pair but with two different recovery procedures, one for stresses and another for the displacements. However the references related with displacement recovery procedures are not as common as those related with stress-based recovery methods.

Regarding to the MLS approaches we will present in Appendix A an equilibrated displacement recovery procedure based on the MLS approach, the MLS-CD. “C” stands for the constrains to impose the equilibrium and “D” because it is based on the recovery of the displacement field. We recommend the use of the MLS-CD technique in meshless methods because it does not need any mesh structure.

This Section shows the displacement-based recovery procedure SPR-CD, based on the SPR-C scheme, specially adapted to the cgFEM frame-work and capable to deal with singular problems. This technique consists in a displacement recovery procedure where at each patch, we will be able to impose the satisfac-

²Hereinafter subindex u will refer to kinematically admissible global fields and subindex σ will refer to nearly-statically admissible global fields.

tion of the Dirichlet boundary conditions, internal equilibrium equation and boundary equilibrium equation. The SPR-C was used to obtain upper bounds of the error in energy norm [73]. However some correction terms depending on the exact error of the displacement field were needed. The SPR-CD technique results more handy than the technique based on stresses, SPR-C [71], because the recovered displacement field could be used to evaluate an approximation to those correction terms.

As in the case of SPR-C, SPR-CD is based on subdividing the domain in small regions or patches \mathcal{P}^i . In Figure 3.8 we show an example of a patch of elements around a node i . This is the general case where the patch is subdivided in two zones by an internal geometrical curve, that could represent the boundary between two materials or a crack surface.

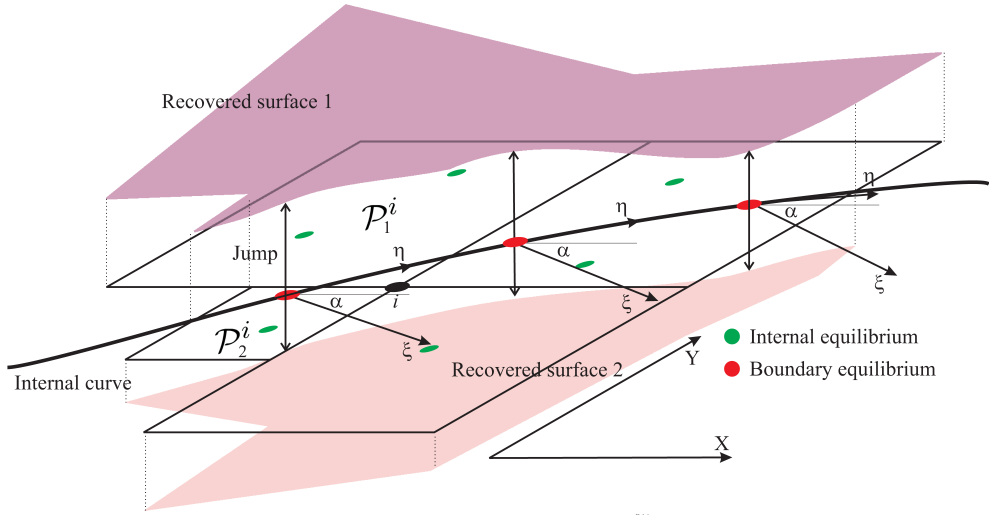


Figure 3.8: Example of an internal patch split by an internal curve. The black circle indicates the position of the node shared between the elements of the patch. The red points indicate the position where boundary equilibrium is imposed and the green points indicate some random points where the internal equilibrium constraints are imposed.

In the SPR-CD we fit, at each patch a polynomial surface per component to recover a displacement field as in [71] for stresses. Next, we add extra informa-

tion to improve the solution at each patch. This information is related to the internal and boundary equilibrium and Dirichlet boundary conditions. Then a local recovered displacement field $\hat{\mathbf{u}}_i^*$ will be evaluated into each patch around the node i . $\hat{\mathbf{u}}_i^*$ will be evaluated using a scheme similar to that used for the SPR technique [7] but in this case we use the FE displacements of the elements within the patch instead of the corresponding FE stresses. The polynomial surface $\hat{\mathbf{u}}_i^*$ will be forced to satisfy the Dirichlet boundary conditions and the stresses $\hat{\boldsymbol{\sigma}}_i^*(\hat{\mathbf{u}}_i^*)$, will be forced to satisfy the internal equilibrium equation and the boundary equilibrium equation. Note that as $\hat{\boldsymbol{\sigma}}_i^*$ is directly evaluated from $\hat{\mathbf{u}}_i^*$, the compatibility equation is satisfied and it does not need to be explicitly considered as in the case of the SPR-C technique. The satisfaction of these equations will be enforced using the Lagrange multipliers technique by a point collocation technique in a sufficient number of points, according to the degree of the recovered displacement field. Thus, we define the local recovered displacement and stress fields as follows:

$$\hat{\mathbf{u}}_i^*(\mathbf{x}) = \begin{Bmatrix} \hat{u}_x^*(\mathbf{x}) \\ \hat{u}_y^*(\mathbf{x}) \end{Bmatrix} = \mathbf{P}(\mathbf{x})\mathbf{A} = \begin{bmatrix} \mathbf{p}(\mathbf{x}) & \mathbf{0} \\ \mathbf{0} & \mathbf{p}(\mathbf{x}) \end{bmatrix} \begin{Bmatrix} \mathbf{a}_x \\ \mathbf{a}_y \end{Bmatrix} \quad (3.44)$$

$$\hat{\boldsymbol{\sigma}}_i^*(\mathbf{x}) = \mathbf{DL}\hat{\mathbf{u}}_i^*(\mathbf{x}) \quad (3.45)$$

where $\mathbf{p}(\mathbf{x}) = \{1, x, y\}$ for the linear case, $\mathbf{p}(\mathbf{x}) = \{1, x, y, x^2, xy, y^2\}$ for the quadratic case and $\mathbf{p}(\mathbf{x}) = \{1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3\}$ for the cubic case, and \mathbf{a}_k are the corresponding vectors of unknown coefficients. Note that, in this case, the degree p of the recovered displacement is one degree higher than the FE nodal interpolation. Under this definition, the functional to be optimized at each patch \mathcal{P}^i reads as follows:

$$\begin{aligned} \Phi_D(\mathbf{A}, \boldsymbol{\lambda}) := & \int_{\mathcal{P}^i} (\mathbf{P}(\mathbf{x})\mathbf{A} - \mathbf{u}^h(\mathbf{x}))^2 d\Omega + \\ & \sum_j^{nbee} \lambda_j^{nbee} (c^{\text{int}}(\mathbf{x}_j)) + \sum_j^{niee} \lambda_j^{niee} (c^{\text{ext}}(\mathbf{x}_j)) + \sum_j^{ndce} \lambda_j^{ndce} (c^{\text{dir}}(\mathbf{x}_j)) \end{aligned} \quad (3.46)$$

where $c^{\text{int}}(\mathbf{x}_j)$ is defined in (3.38) and $c^{\text{ext}}(\mathbf{x}_j)$ in (3.39), considering that $\hat{\boldsymbol{\sigma}}_i^*(\mathbf{x}_j)$ is defined in (3.45), and $c^{\text{dir}}(\mathbf{x}_j)$ in (3.49). Finally, optimizing (3.46)

we obtain the linear system of equations to solve at each patch:

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}^T \\ \mathbf{C}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{A} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{H} \\ \Lambda \end{Bmatrix} \quad (3.47)$$

which yields the coefficients \mathbf{A} for the recovered displacement field. The global kinematically admissible displacement field is evaluated at each element using the "Conjoint Polynomial Enhancement" [69], using the displacement field $\hat{\mathbf{u}}_i^*$ evaluated from the patches corresponding to each of the n_v vertex nodes of the element and the linear shape functions N_i associated to these nodes:

$$\mathbf{u}_u^*(\mathbf{x}) = \sum_{i=1}^{n_v} N_i(\mathbf{x}) \hat{\mathbf{u}}_i^*(\mathbf{x}) \quad (3.48)$$

Note that because of the use of (3.48) we will lose the internal equilibration of the patch recovered stress field $\hat{\boldsymbol{\sigma}}_i^*$ as it will be detailed in Section 3.3.3.

3.3.1 Constraints definition

As in the SPR-C, we enrich the recovered field with known information about boundary conditions and equilibrium increasing the accuracy of the recovered pair $(\mathbf{u}_u^*, \boldsymbol{\sigma}_\sigma^*)$. The constraints considered, at patch level, for the evaluation of $\hat{\mathbf{u}}_i^*$ are:

- Dirichlet constraints: constraints related with the Dirichlet boundary conditions can be written as:

$$c^{\text{dir}}(\mathbf{x}_j) : \hat{\mathbf{u}}_i^*(\mathbf{x}_j) - \mathbf{u}(\mathbf{x}_j) = 0 \quad \forall \mathbf{x}_j \in \Gamma_D \cap \mathcal{P}^i \quad (3.49)$$

We impose the satisfaction of the Dirichlet boundary conditions at $ndce = p+1$ points (being p the degree of the recovered displacement field) along the part of Γ_D falling into the patch.

- Internal equilibrium equations: defined in (3.38)
- Boundary equilibrium equations: defined in (3.39)

3.3.2 Patches cut by the boundary

In some situations a patch of elements is cut by an *internal* boundary. This type of situations are common in multi-material problems or along cracks modeled with the XFEM. In Figure 3.8 we have shown a patch intersected by an internal boundary. In this case we define two recovered surfaces $\hat{\mathbf{u}}_i^*|_{\mathcal{P}_1^i}$ and $\hat{\mathbf{u}}_i^*|_{\mathcal{P}_2^i}$, one at each side of the boundary. Each one of these recovered surfaces will satisfy the internal equilibrium equation by considering (3.38) in each of them. Boundary equilibrium and displacement continuity is also imposed by using an extended version of equations (3.49) and (3.39):

- Dirichlet constraints: usually, internal boundaries correspond to Neumann-type boundaries where traction are imposed. However the recovery process could provide discontinuous recovered displacement fields where \mathbf{u}_{jump} represents the jump of the displacement field along the respective boundary.

$$c^{\text{dir}}(\mathbf{x}_j) : \hat{\mathbf{u}}_i^*(\mathbf{x}_j)|_{\mathcal{P}_1^i} - \hat{\mathbf{u}}_i^*(\mathbf{x}_j)|_{\mathcal{P}_2^i} = \mathbf{u}_{\text{jump}}(\mathbf{x}_j) \quad \forall \mathbf{x}_j \in \Gamma_D \cap \mathcal{P}^i \quad (3.50)$$

where, to ensure the continuity of the displacement field $\mathbf{u}_{\text{jump}} = 0$, thus introducing a weak discontinuity. In the case that $\mathbf{u}_{\text{jump}} \neq 0$ a strong discontinuity will be introduced.

- Boundary equilibrium equations: in this case we allow a strong discontinuity between the recovered stress fields at each side of the internal boundary, provided by an internal traction along Γ_N .

$$\begin{aligned} c^{\text{ext}}(\mathbf{x}_j) : \mathbf{G} \{ \mathbf{DL}\hat{\mathbf{u}}_i^*(\mathbf{x}_j) + \boldsymbol{\sigma}_0(\mathbf{x}_j) - \mathbf{D}\boldsymbol{\varepsilon}_0(\mathbf{x}_j) \} |_{\mathcal{P}_1^i} - \\ \mathbf{G} \{ \mathbf{DL}\hat{\mathbf{u}}_i^*(\mathbf{x}_j) + \boldsymbol{\sigma}_0(\mathbf{x}_j) - \mathbf{D}\boldsymbol{\varepsilon}_0(\mathbf{x}_j) \} |_{\mathcal{P}_2^i} = \mathbf{t}(\mathbf{x}_j) \quad \forall \mathbf{x}_j \in \Gamma_N \cap \mathcal{P}^i \end{aligned} \quad (3.51)$$

This strong discontinuity between the recovered stress fields will be interesting for Goal Oriented Adaptivity as it will be indicated in Chapter 4.

3.3.3 Recovered stress evaluation

$\mathbf{u}_u^*(\mathbf{x})$ is a kinematically admissible recovered displacement field. To obtain a consistent recovered stress field we should differentiate (3.48) according to the following expression:

$$\begin{aligned}\boldsymbol{\sigma}_u^*(\mathbf{x}) &= \mathbf{DL} \sum_{i=1}^{n_v} N_i(\mathbf{x}) \hat{\mathbf{u}}_i^*(\mathbf{x}) + \boldsymbol{\sigma}_0(\mathbf{x}) - \mathbf{D}\boldsymbol{\varepsilon}_0(\mathbf{x}) \\ \boldsymbol{\sigma}_u^*(\mathbf{x}) &= \underbrace{\sum_{i=1}^{n_v} \mathbf{D}(\mathbf{L}N_i(\mathbf{x})) \hat{\mathbf{u}}_i^*(\mathbf{x})}_{\text{discontinuous}} + \underbrace{\sum_{i=1}^{n_v} N_i(\mathbf{x}) \underbrace{\mathbf{DL}\hat{\mathbf{u}}_i^*(\mathbf{x})}_{\hat{\boldsymbol{\sigma}}_i^*(\mathbf{x})}}_{\boldsymbol{\sigma}_\sigma^*(\mathbf{x}) \text{ continuous}} + \boldsymbol{\sigma}_0(\mathbf{x}) - \mathbf{D}\boldsymbol{\varepsilon}_0(\mathbf{x})\end{aligned}\tag{3.52}$$

When we apply the differential operator \mathbf{L} to the kinematically admissible displacement field \mathbf{u}_u^* we generate the kinematically admissible pair $(\mathbf{u}_u^*, \boldsymbol{\sigma}_u^*)$. If we observe equation (3.52), the field $\boldsymbol{\sigma}_u^*$ is split into two parts, one continuous and one discontinuous. The continuous part coincides with the partition of unity of the patch-wise recovered stress field $\hat{\boldsymbol{\sigma}}_i^*$, directly derived from $\hat{\mathbf{u}}_i^*$ (see (3.45)). Note that $\hat{\boldsymbol{\sigma}}_i^*$ will satisfy the equilibrium equations.

It can be said that the main part of the stress field description will be taken into account in the continuous part $\boldsymbol{\sigma}_\sigma^*$, since the discontinuous part will tend to zero with the mesh refinement. Note that in the infinite dimensional space, $\hat{\mathbf{u}}_i^*$ will have the same value at each point when evaluated from the different patches, the discontinuous part will be zero because of the partition of nullity of the derivatives of the shape functions. Moreover, the statically admissible stress fields have, in general, a better quality than the kinematically admissible ones since the equilibrium conditions are strongly enforced, and in this particular case we could have a high control of the statical admissibility properties of the continuous part. Because of these reasons and in order to retain the continuity of the recovered stress field we will use as general output for

the recovered stress field only the continuous part:

$$\boldsymbol{\sigma}_\sigma^*(\mathbf{x}) = \sum_{i=1}^{n_v} N_i(\mathbf{x}) \mathbf{D} \mathbf{L} \mathbf{u}_i^*(\mathbf{x}) = \sum_{i=1}^{n_v} N_i(\mathbf{x}) \boldsymbol{\sigma}_i^*(\mathbf{x}) + \boldsymbol{\sigma}_0(\mathbf{x}) - \mathbf{D} \boldsymbol{\varepsilon}_0(\mathbf{x}) \quad (3.53)$$

Note that, because of the use partition of unity technique in (3.53), $\boldsymbol{\sigma}_\sigma^*$ will not satisfy the internal equilibrium equation:

$$\begin{aligned} \mathbf{L}^T \boldsymbol{\sigma}_\sigma^*(\mathbf{x}) &= \mathbf{L}^T \sum_{i=1}^{n_v} N_i(\mathbf{x}) \boldsymbol{\sigma}_i^*(\mathbf{x}) + \mathbf{L}^T (\boldsymbol{\sigma}_0(\mathbf{x}) - \mathbf{D} \boldsymbol{\varepsilon}_0(\mathbf{x})) \\ \mathbf{L}^T \boldsymbol{\sigma}_\sigma^*(\mathbf{x}) &= \underbrace{\sum_{i=1}^{n_v} \mathbf{L}^T N_i(\mathbf{x}) \boldsymbol{\sigma}_i^*(\mathbf{x})}_{-\mathbf{s}_\sigma^*} + \sum_{i=1}^{n_v} N_i(\mathbf{x}) + \mathbf{L}^T (\boldsymbol{\sigma}_0(\mathbf{x}) - \mathbf{D} \boldsymbol{\varepsilon}_0(\mathbf{x})) \quad (3.54) \\ \mathbf{L}^T \boldsymbol{\sigma}_\sigma^*(\mathbf{x}) &= -\mathbf{s}_\sigma^*(\mathbf{x}) - \mathbf{b}(\mathbf{x}) \end{aligned}$$

This expression is a modified version of the internal equilibrium equation where \mathbf{s}_σ^* represents the lack of internal equilibrium. Furthermore, there could also exist a lack of boundary equilibrium of $\boldsymbol{\sigma}_\sigma^*$ over Γ_N which can be evaluated as $\mathbf{r}_\sigma^* = \mathbf{G} \boldsymbol{\sigma}_\sigma^* - \mathbf{t}_{\Gamma_N}$, where \mathbf{t}_{Γ_N} are the exact tractions over the Neumann boundaries.

Hence, the standard output of the cgFEM code will be the pair $(\mathbf{u}_u^*, \boldsymbol{\sigma}_\sigma^*)$ instead of $(\mathbf{u}^h, \boldsymbol{\sigma}^h)$. Error estimator for the recovered solution and h -adaptive processes based on the error of the recovered solution will be presented in Chapter 6.

3.3.4 Singular fields

The SPR-CD recovery technique can also consider the singular behavior of the solution due to a re-entrant corner, a crack, etc. We have to take into account that the SPR-based techniques tend to increase the smoothness of the solution. However this characteristic, being convenient for standard situations, can decrease the accuracy in the surrounding of the singularity. Different

techniques have been proposed to account for the singular behavior during the recovery process [80, 76, 78]. Here, following the ideas in [80], for stresses in singular problems, our solution $(\mathbf{u}, \boldsymbol{\sigma})$ will be split into 2 parts, one singular $(\mathbf{u}_{\text{sing}}, \boldsymbol{\sigma}_{\text{sing}})$ and one smooth $(\mathbf{u}_{\text{smo}}, \boldsymbol{\sigma}_{\text{smo}})$:

$$\begin{aligned}\mathbf{u} &= \mathbf{u}_{\text{smo}} + \mathbf{u}_{\text{sing}} \\ \boldsymbol{\sigma} &= \boldsymbol{\sigma}_{\text{smo}} + \boldsymbol{\sigma}_{\text{sing}}\end{aligned}\tag{3.55}$$

note that the displacement field \mathbf{u} does not have singular behavior, however we maintain the subscripts “smo” and “sing” for consistency. The recovered fields $(\mathbf{u}^*, \boldsymbol{\sigma}^*)$ can also be expressed as the contribution of two recovered fields, one smooth and one singular:

$$\begin{aligned}\mathbf{u}_u^* &= \mathbf{u}_{\text{smo}}^* + \mathbf{u}_{\text{sing}}^* \\ \boldsymbol{\sigma}_\sigma^* &= \boldsymbol{\sigma}_{\text{smo}}^* + \boldsymbol{\sigma}_{\text{sing}}^*\end{aligned}\tag{3.56}$$

For the recovery of the singular part we will use the expressions which describe the asymptotic fields near the crack tip with respect to a coordinate system (r, ϕ) at the tip (see Figure 3.9) as described in [112]:

$$\mathbf{u}_{\text{sing}}(r, \phi) = K_{\text{I}} r^{\lambda_{\text{I}}} \boldsymbol{\Psi}_{\text{I}}(\lambda_{\text{I}}, \phi) + K_{\text{II}} r^{\lambda_{\text{II}}} \boldsymbol{\Psi}_{\text{II}}(\lambda_{\text{II}}, \phi)\tag{3.57}$$

$$\boldsymbol{\sigma}_{\text{sing}}(r, \phi) = K_{\text{I}} \lambda_{\text{I}} r^{\lambda_{\text{I}}-1} \boldsymbol{\Phi}_{\text{I}}(\lambda_{\text{I}}, \phi) + K_{\text{II}} \lambda_{\text{II}} r^{\lambda_{\text{II}}-1} \boldsymbol{\Phi}_{\text{II}}(\lambda_{\text{II}}, \phi)\tag{3.58}$$

where r is the radial distance to the corner, λ_m (with $m = \text{I, II}$) are the eigenvalues that determine the order of the singularity, $\boldsymbol{\Psi}_m$ and $\boldsymbol{\Phi}_m$ are sets of trigonometric functions that depend on the angular position ϕ , and K_m are the so-called Generalized Stress Intensity Factors (GSIFs). The GSIF is a multiplicative constant that depends on the loading of the problem and linearly determines the intensity of the displacement and stress fields in the vicinity of the singular point. Therefore, the eigenvalues λ and the Generalized Stress Intensity Factor (GSIF) K define the singular field.

The Generalized Stress Intensity Factor K is the characterizing parameter in fracture mechanics problems with singularities. In the particular case in which $2\alpha = 360^\circ$ the problem will correspond to a crack as considered in the context of Linear Elastic Fracture Mechanics where this parameter is called

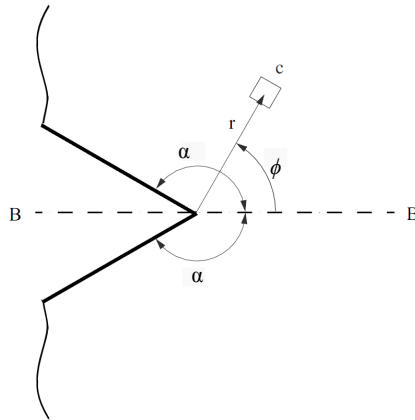


Figure 3.9: Elastic solid with a V-notch

the Stress Intensity Factor (SIF). Let us consider the general singular problem of a V-notch domain subjected to loads in the infinite as shown in Figure 3.9.

The analytical solution for this singular elasticity problem can be found in [112, 113] where, in accordance with the polar reference system shown in Figure 3.9, the displacement and stress fields at points sufficiently close to the corner can be described according to (3.57). In practice, the eigenvalue λ is easily known in advance because it depends solely on the corner angle α and can be obtained as the smallest positive root of the following characteristic equations:

$$\begin{aligned} \sin \lambda_{\text{I}}\alpha + \lambda_{\text{I}} \sin \alpha &= 0 \\ \sin \lambda_{\text{II}}\alpha - \lambda_{\text{II}} \sin \alpha &= 0 \end{aligned} \tag{3.59}$$

The set of trigonometric functions for the displacement and stress fields under model I are [112]:

$$\begin{aligned}\Psi_I(\lambda_I, \phi) &= \begin{Bmatrix} \Psi_{I,x}(\lambda_I, \phi) \\ \Psi_{I,y}(\lambda_I, \phi) \end{Bmatrix} \\ &= \frac{1}{2\mu} \begin{Bmatrix} (\kappa - Q_I(\lambda_I + 1)) \cos \lambda_I \phi - \lambda_I \cos(\lambda_I - 2)\phi \\ (\kappa + Q_I(\lambda_I + 1)) \sin \lambda_I \phi + \lambda_I \sin(\lambda_I - 2)\phi \end{Bmatrix} \end{aligned} \quad (3.60)$$

$$\begin{aligned}\Phi_I(\lambda_I, \phi) &= \begin{Bmatrix} \Phi_{I,xx}(\lambda_I, \phi) \\ \Phi_{I,yy}(\lambda_I, \phi) \\ \Phi_{I,xy}(\lambda_I, \phi) \end{Bmatrix} \\ &= \begin{Bmatrix} (2 - Q_I(\lambda_I + 1)) \cos(\lambda_I - 1)\phi - (\lambda_I - 1) \cos(\lambda_I - 3)\phi \\ (2 + Q_I(\lambda_I + 1)) \cos(\lambda_I - 1)\phi + (\lambda_I - 1) \cos(\lambda_I - 3)\phi \\ Q_I(\lambda_I + 1) \sin(\lambda_I - 1)\phi + (\lambda_I - 1) \sin(\lambda_I - 3)\phi \end{Bmatrix} \end{aligned} \quad (3.61)$$

where κ is the Kolosov's constant, μ is the shear modulus and Q is a constant for a given notch angle:

$$Q_I = -\frac{\left(\frac{\lambda_I - 1}{\lambda_I + 1}\right) \cdot \sin\left((\lambda_I - 1)\frac{\alpha}{2}\right)}{\sin\left((\lambda_I + 1)\frac{\alpha}{2}\right)} \quad (3.62)$$

$$(3.63)$$

For mode II we have:

$$\begin{aligned} \Psi_{II}(\lambda_{II}, \phi) &= \begin{Bmatrix} \Psi_{II,x}(\lambda_{II}, \phi) \\ \Psi_{II,y}(\lambda_{II}, \phi) \end{Bmatrix} \\ &= \frac{1}{2\mu} \begin{Bmatrix} (\kappa - Q_{II}(\lambda_{II} + 1)) \sin \lambda_{II} \phi - \lambda_{II} \sin(\lambda_{II} - 2)\phi \\ -(\kappa + Q_{II}(\lambda_{II} + 1)) \cos \lambda_{II} \phi - \lambda_{II} \cos(\lambda_{II} - 2)\phi \end{Bmatrix} \end{aligned} \quad (3.64)$$

$$\begin{aligned} \Phi_{II}(\lambda_{II}, \phi) &= \begin{Bmatrix} \Phi_{II,xx}(\lambda_{II}, \phi) \\ \Phi_{II,yy}(\lambda_{II}, \phi) \\ \Phi_{II,xy}(\lambda_{II}, \phi) \end{Bmatrix} \\ &= \begin{Bmatrix} (2 - Q_{II}(\lambda_{II} + 1)) \sin(\lambda_{II} - 1)\phi - (\lambda_{II} - 1) \sin(\lambda_{II} - 3)\phi \\ (2 + Q_{II}(\lambda_{II} + 1)) \sin(\lambda_{II} - 1)\phi + (\lambda_{II} - 1) \sin(\lambda_{II} - 3)\phi \\ Q_{II}(\lambda_{II} + 1) \cos(\lambda_{II} - 1)\phi + (\lambda_{II} - 1) \cos(\lambda_{II} - 3)\phi \end{Bmatrix} \end{aligned} \quad (3.65)$$

$$Q_{II} = -\frac{\sin\left((\lambda_{II} - 1)\frac{\alpha}{2}\right)}{\sin\left((\lambda_{II} + 1)\frac{\alpha}{2}\right)} \quad (3.66)$$

To evaluate the GSIF it is a common practice to use the interaction integral in its Equivalent Domain Integral (EDI) form. There are different expressions already available to evaluate the EDI for singular problems. In this work, we consider the expression shown in [113], expressed in indicial notation.

$$K = -\frac{1}{C} \int_{\Omega} (\sigma_{jk} u_k^{\text{aux}} - \sigma_{jk}^{\text{aux}} u_k) \frac{\partial q}{\partial x_j} d\Omega \quad (3.67)$$

where u^{aux} , σ^{aux} are the auxiliary fields associated with the extraction functions for the GSIFs in mode I or mode II, q is a function used to define the extraction zone and x_j is referred to the local coordinates system at the crack tip. This expression can be recast in vectorial notation as:

$$K = -\frac{1}{C} \int_{\Omega} \boldsymbol{\sigma}^T \begin{Bmatrix} u_x^{\text{aux}} q_{,x} \\ u_y^{\text{aux}} q_{,y} \\ u_y^{\text{aux}} q_{,x} + u_x^{\text{aux}} q_{,y} \end{Bmatrix} - \mathbf{u}^T \begin{Bmatrix} \sigma_{xx}^{\text{aux}} q_{,x} + \sigma_{xy}^{\text{aux}} q_{,y} \\ \sigma_{xy}^{\text{aux}} q_{,x} + \sigma_{yy}^{\text{aux}} q_{,y} \end{Bmatrix} d\Omega \quad (3.68)$$

where $q_{,x} = \frac{\partial q}{\partial x}$ and $q_{,y} = \frac{\partial q}{\partial y}$.

The function q must meet some requirements. It must be 1 at the singular point and 0 on the outer boundary and also be at least as regular as the FE solution. A suitable function q used in this work can be defined as:

$$q(s) = \begin{cases} 1 & \text{if } s \leq r_{int} \\ 1 - 6X^2 + 8X^3 - 3X^4 & \text{if } s \in [r_{int}, r_{ext}] \\ 0 & \text{if } s > r_{ext} \end{cases} \quad (3.69)$$

where $X = \frac{s - r_{int}}{r_{ext} - r_{int}}$, r_{int} and r_{ext} are the internal and external radius of the extraction zone respectively and $s(x, y) = \sqrt{(x - x_s)^2 + (y - y_s)^2}$, being (x_s, y_s) the coordinates of the singular point. This function q has been selected in order to provide a plateau area around the singular point and then a smooth transition between 1 to 0 between r_{int} and r_{ext} .

In (3.68) the auxiliary fields for the problem in mode I are defined as:

$$\mathbf{u}^{aux}(r, \phi) = r^{-\lambda_I} \Psi_I(-\lambda_I, \phi) \quad (3.70)$$

$$\boldsymbol{\sigma}^{aux}(r, \phi) = -\lambda_I r^{-\lambda_I-1} \Phi_I(-\lambda_I, \phi) \quad (3.71)$$

Analogously, the auxiliary fields for the problem in mode II are defined as:

$$\mathbf{u}^{aux}(r, \phi) = r^{-\lambda_{II}} \Psi_{II}(-\lambda_{II}, \phi) \quad (3.72)$$

$$\boldsymbol{\sigma}^{aux}(r, \phi) = -\lambda_{II} r^{-\lambda_{II}-1} \Phi_{II}(-\lambda_{II}, \phi) \quad (3.73)$$

Finally C is a constant that can be evaluated for mode I using the expression

$$C_I = \int_{-\alpha/2}^{\alpha/2} [\lambda_I \Xi_I(\lambda_I, \phi) \cdot \Psi_I(-\lambda_I, \phi) - (-\lambda_I) \Xi_I(-\lambda_I, \phi) \cdot \Psi_I(\lambda_I, \phi)] d\phi \quad (3.74)$$

and for mode II, using the expression

$$C_{II} = \int_{-\alpha/2}^{\alpha/2} [\lambda_{II} \Xi_{II}(\lambda_{II}, \phi) \cdot \Psi_{II}(-\lambda_{II}, \phi) - (-\lambda_{II}) \Xi_{II}(-\lambda_{II}, \phi) \cdot \Psi_{II}(\lambda_{II}, \phi)] d\phi \quad (3.75)$$

where Ψ_I and Ψ_{II} are the displacement trigonometric functions given in (3.60) and (3.64), and Ξ_I and Ξ_{II} are the trigonometric functions associated with the

tractions vector related to Φ_I and Φ_{II} respectively, i.e.:

$$\Xi_I = \begin{Bmatrix} \Phi_{I,xx}(\lambda, \phi) \cos \phi + \Phi_{I,xy}(\lambda, \phi) \sin \phi \\ \Phi_{I,xy}(\lambda, \phi) \cos \phi + \Phi_{I,yy}(\lambda, \phi) \sin \phi \end{Bmatrix} \quad (3.76)$$

$$\Xi_{II} = \begin{Bmatrix} \Phi_{II,xx}(\lambda, \phi) \cos \phi + \Phi_{II,xy}(\lambda, \phi) \sin \phi \\ \Phi_{II,xy}(\lambda, \phi) \cos \phi + \Phi_{II,yy}(\lambda, \phi) \sin \phi \end{Bmatrix} \quad (3.77)$$

The singular field ($\mathbf{u}_{\text{sing}}^*, \boldsymbol{\sigma}_{\text{sing}}^*$) will be evaluated as in (3.57) considering the GSIF value evaluated with expression (3.68). $\mathbf{u}_{\text{smo}}^h$ will be defined subtracting the singular part $\mathbf{u}_{\text{sing}}^*$ to the FE solution \mathbf{u}^h :

$$\mathbf{u}_{\text{smo}}^h = \mathbf{u}^h - \mathbf{u}_{\text{sing}}^* \quad (3.78)$$

$\mathbf{u}_{\text{smo}}^h$ will be used as the input for the SPR-CD recovery process that will yield the pair ($\mathbf{u}_{\text{smo}}^*, \boldsymbol{\sigma}_{\text{smo}}^*$). The final recovered solution will be evaluated using (3.56). Note that $\mathbf{L}^T \boldsymbol{\sigma}_{\text{sing}}^* = 0$, therefore no additional terms will be considered in (3.38) when applying the SPR-CD technique to $\boldsymbol{\sigma}_{\text{smo}}^h = \boldsymbol{\sigma}^h - \boldsymbol{\sigma}_{\text{sing}}^h$. However, equation (3.39) and also (3.51) will be modified by subtraction to the r.h.s. the singular part of the traction $\mathbf{t}_{\text{sing}}^* = \mathbf{G} \boldsymbol{\sigma}_{\text{sing}}^*$. The same will occurs in the constrains related with the Dirichlet boundaries, (3.49) and (3.50), where the r.h.s will be modified by subtracting the singular part of the displacement field $\mathbf{u}_{\text{sing}}^*$.

This splitting procedure is not used in the whole domain of the problem but only in an area close to the singular point in order to localize the process allowing for a number of different singularities in the same problem. The area affected by this process is defined by the user, and it should be related with extension of the singular behavior of the stress field.

3.3.5 Efficiency of the recovery procedure

A maximum difference of one level is allowed in 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. Let us define as the internal patches those composed by internal elements only. Figure 3.10 represents the 19 possible configurations of internal patches for 2D.

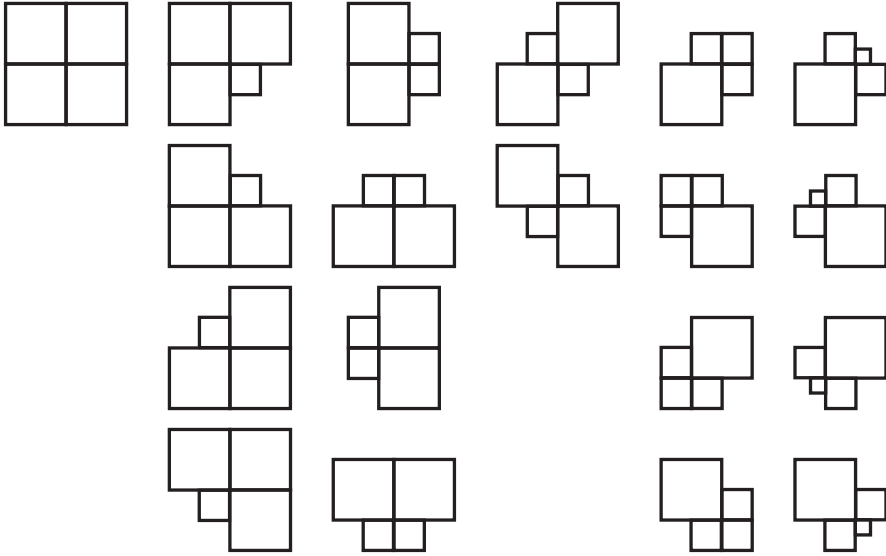


Figure 3.10: Possible configurations of internal patches in 2D.

The polynomial coefficients \mathbf{A} used to describe the recovered displacement field according to (3.47) are obtained for a normalized coordinate system. Then, the matrix in (3.47) will be exactly the same for all internal patches having the same configuration. This implies that we will only need to invert it a maximum of 19 times 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 3.10, then we invert the coefficient matrix in (3.47) for each one of the different patch configurations. Once the r.h.s. in (3.47) have been evaluated for each patch, the unknown coefficients are directly evaluated. 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 a different system configuration. 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.

3.4 Error estimation in energy norm. Error bounding

3.4.1 Error estimation in energy norm

Recovery-type error estimators rely on the use of the Zienkiewicz and Zhu (ZZ) error estimators presented in (2.28). The quality of the estimation is highly related with the quality of the recovered field $\boldsymbol{\sigma}^*$, so the accuracy of the estimate will be strongly affected by the quality of the recovered field. We will compare two different estimators, the first one with the stress field provided by the SPR-C technique $\boldsymbol{\sigma}_{\text{SPR-C}}^*$, and the second one with the stress field provided by the SPR-CD technique $\boldsymbol{\sigma}_\sigma^*$:

$$\mathcal{E}_{\text{SPR-C}}^2 := \int_{\Omega} (\boldsymbol{\sigma}_{\text{SPR-C}}^* - \boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} (\boldsymbol{\sigma}_{\text{SPR-C}}^* - \boldsymbol{\sigma}^h) \, d\Omega \quad (3.79)$$

$$\mathcal{E}_{\text{SPR-CD}}^2 := \int_{\Omega} (\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} (\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h) \, d\Omega \quad (3.80)$$

Note that both fields, $\boldsymbol{\sigma}_{\text{SPR-C}}^*$ and $\boldsymbol{\sigma}_\sigma^*$, are locally equilibrated, but SPR-C is a stress-based smoothing technique, while SPR-CD is displacement-based. We expect similar results, but the SPR-CD recovery process, having a similar computational cost than the SPR-C, will provide us also a recovered displacement field that will be useful in the following Sections.

3.4.2 Error bounds in energy norm

Recovery based error estimators have a small computational cost and are easy to adapt to existing FE commercial codes. Numerical experiences demonstrate their robustness and accuracy [98, 99] in the vast majority of engineering problems. However these kinds of error estimators have an important drawback, despite of their accuracy, so far they are unable to provide guaranteed error bounds in energy norm.

If the recovered stress field used in the ZZ error estimator is a statically admissible one, since the FE stress field $\boldsymbol{\sigma}^h$ is kinematically admissible, the ZZ error estimator becomes equivalent to the Constitutive Relation Error (CRE) [81], directly providing upper error bounds.

Both, the CRE and the ZZ error estimators are usually based on comparing the finite element solution with an improved one, let us call it $\check{\boldsymbol{\sigma}}$. The estimated error is evaluated with the following expression:

$$\mathcal{E}^2 = \bar{a}(\check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}^h, \check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}^h) \quad (3.81)$$

Note that if we use $\check{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^*$ we obtain the ZZ error estimator (2.28), but if we consider $\check{\boldsymbol{\sigma}} = \check{\boldsymbol{\sigma}}$ with, as described in Section 3.2.3, $\check{\boldsymbol{\sigma}}$ being a statically admissible stress field, we obtain the CRE (3.29). Thus (3.81) yields different estimators having different properties as previously explained.

If we expand (3.81), comparing both approximate solutions with the exact one, we can write the following expression:

$$\begin{aligned} \mathcal{E}^2 = & \bar{a}(\check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}, \check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}) + a(\mathbf{u} - \mathbf{u}^h, \mathbf{u} - \mathbf{u}^h) + \\ & 2 \int_{\Omega} (\check{\boldsymbol{\sigma}} - \boldsymbol{\sigma})^T (\boldsymbol{\varepsilon}(\mathbf{u}) - \boldsymbol{\varepsilon}(\mathbf{u}^h)) \, d\Omega \end{aligned} \quad (3.82)$$

Performing an integration by parts to the last term in (3.82), and assuming a sufficient smoothness of the improved stress field $\check{\boldsymbol{\sigma}}$ at each element, we obtain

the following expression:

$$\begin{aligned}
 \int_{\Omega} (\check{\boldsymbol{\sigma}} - \boldsymbol{\sigma})^T (\boldsymbol{\varepsilon}(\mathbf{u}) - \boldsymbol{\varepsilon}(\mathbf{u}^h)) \, d\Omega &= - \sum_{K \in \mathcal{T}} \int_{\Omega^K} (\mathbf{L}^T (\check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}))^T (\mathbf{u} - \mathbf{u}^h) \, d\Omega \\
 &+ \sum_{K \in \mathcal{T}} \int_{\partial\Omega^K} (\mathbf{G}(\check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}))^T (\mathbf{u} - \mathbf{u}^h) \, d\Gamma \\
 &= \sum_{K \in \mathcal{T}} \int_{\partial\Omega^K \cap \Gamma_N} (\mathbf{G}\check{\boldsymbol{\sigma}} - \mathbf{t})^T (\mathbf{u} - \mathbf{u}^h) \, d\Gamma \\
 &+ \sum_{K \in \mathcal{T}} \int_{\partial\Omega^K \setminus \Gamma_N} (\mathbf{G}\check{\boldsymbol{\sigma}})^T (\mathbf{u} - \mathbf{u}^h) \, d\Gamma \\
 &- \sum_{K \in \mathcal{T}} \int_{\Omega^K} (\mathbf{L}^T \check{\boldsymbol{\sigma}} + \mathbf{b})^T (\mathbf{u} - \mathbf{u}^h) \, d\Omega
 \end{aligned} \tag{3.83}$$

Finally we could observe that if the stress field $\check{\boldsymbol{\sigma}}$ is equilibrated at each element the last term in (3.82) is zero. This is because (3.83) represent the work of a load-free problem if $\check{\boldsymbol{\sigma}}$ is equilibrated. Thus, from (3.82), if the improved field $\check{\boldsymbol{\sigma}}$ is statically admissible, then:

$$\|\mathbf{e}\|_{\Omega}^2 = a(\mathbf{u} - \mathbf{u}^h, \mathbf{u} - \mathbf{u}^h) = \mathcal{E}^2 - \bar{a}(\check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}, \check{\boldsymbol{\sigma}} - \boldsymbol{\sigma}) \leq \mathcal{E}^2 \tag{3.84}$$

It can be observed that both, the CRE and the ZZ error estimator, become similar when statically admissible recovered fields are used for the recovered solution. Given this equivalence our purpose would be to use SPR-type recovery techniques to generate statically admissible stress fields which will be used in the CRE equation to provide upper bounds of the error in energy norm. The upper error bound in energy norm is achieved if:

$$\begin{aligned}
 &\sum_{K \in \mathcal{T}} \int_{\partial\Omega^K \cap \Gamma_N} (\mathbf{G}\check{\boldsymbol{\sigma}} - \mathbf{t})^T (\mathbf{u} - \mathbf{u}^h) \, d\Gamma \\
 &+ \sum_{K \in \mathcal{T}} \int_{\partial\Omega^K \setminus \Gamma_N} (\mathbf{G}\check{\boldsymbol{\sigma}})^T (\mathbf{u} - \mathbf{u}^h) \, d\Gamma \\
 &- \sum_{K \in \mathcal{T}} \int_{\Omega^K} (\mathbf{L}^T \check{\boldsymbol{\sigma}} + \mathbf{b})^T (\mathbf{u} - \mathbf{u}^h) \, d\Omega = 0.
 \end{aligned} \tag{3.85}$$

For the stress field coming from the SPR-CD recovered procedure, $\boldsymbol{\sigma}_\sigma^*$, we could assume that the second term in (3.85) is null since the recovered stress field is continuous. However, the third term in (3.85) is, in general, not null due to the use of the partition of unity to obtain $\boldsymbol{\sigma}_\sigma^*$, as explained in Section 3.2.6. This leads to a lack of internal equilibrium at each element $\mathbf{s}_\sigma^* = \mathbf{L}^T \boldsymbol{\sigma}_\sigma^* + \mathbf{b}$. Moreover, for curved boundaries or non-polynomial natural constraints we could not guarantee the strict fulfillment of the prescribed traction over the Neumann boundaries Γ_N , then $\mathbf{r}_\sigma^* = \mathbf{G} \boldsymbol{\sigma}_\sigma^* - \mathbf{t}$.

Similar conclusions were also obtained by other authors. Díez *et. al.* [73] and Ródenas *et. al.* [74] introduced an expression to directly obtain upper bounds of the error in energy norm, based on the SPR-C technique, adding some correction terms:

$$\begin{aligned} \|\mathbf{e}\|_\Omega^2 &\leq \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 = \int_\Omega \left(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h \right)^T \mathbf{D}^{-1} \left(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h \right) d\Omega \\ &\quad - 2 \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{e} d\Omega - 2 \int_\Gamma (\mathbf{r}_\sigma^*)^T \mathbf{e} d\Gamma \end{aligned} \quad (3.86)$$

where $\mathbf{e} := \mathbf{u} - \mathbf{u}^h$ is the exact error in displacements. In general this quantity is not available, in [73] the authors obtained an estimation $\mathbf{e}_{e_{s_i}} \approx \mathbf{e}_i$ for mesh i of a sequence of increasingly refined meshes as follows: $\mathbf{e}_i \approx \mathbf{e}_{e_{s_i}} = \mathbf{u}_N^h - \mathbf{u}_i^h \forall i \in 1 \dots N - 1$, where \mathbf{u}_N^h corresponds to the solution obtained for the last mesh of the sequence. The error $\mathbf{e}_{e_{s_N}}$ for the last mesh was obtained by extrapolating form $\mathbf{e}_{e_{s_{N-1}}}$. This process requires the projection from the last mesh of the sequence and it is not accurate enough for the last mesh. Now, with the SPR-CD recovery technique we could approximate \mathbf{e} as $\mathbf{e} \approx \mathbf{e}_u := \mathbf{u}_u^* - \mathbf{u}^h$ at a lower cost. Using that definition we obtain a computable and not guaranteed version of the upper bound in energy norm:

$$\begin{aligned} \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 &= \int_\Omega \left(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h \right)^T \mathbf{D}^{-1} \left(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h \right) d\Omega \\ &\quad - 2 \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{e}_u d\Omega - 2 \int_\Gamma (\mathbf{r}_\sigma^*)^T \mathbf{e}_u d\Gamma \end{aligned} \quad (3.87)$$

So far, we have presented a first low cost numerical upper error bound in energy norm. This is a first approach to obtain upper error bounds in energy norm. More advanced approaches will be described in Chapter 5.

The next step is to obtain a lower bound on the error in energy norm by using the recovered fields at hand. Taking the ideas presented in [65] it is possible to obtain a lower bound of this value from a statically admissible solution in combination with a kinematically admissible solution.

Theorem 3.1. *Being \mathbf{u} the exact displacement solution of the linear elasticity problem presented in Section 2.2, \mathbf{u}^h the corresponding approximation obtained with a displacement-based FEM, \mathbf{u}_u^* a kinematically admissible displacement field, $\boldsymbol{\sigma}_\sigma^*$ a nearly-statically admissible stress field with \mathbf{s}_σ^* and \mathbf{r}_σ^* the corresponding lacks in internal and boundary equilibrium, respectively. The following expression*

$$2\lambda \left\{ \bar{a}(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h, \sigma(\mathbf{e}_u)) - \int_{\Omega} (\mathbf{s}_\sigma^*)^T \mathbf{e}_u \, d\Omega - \int_{\Gamma} (\mathbf{r}_\sigma^*)^T \mathbf{e}_u \, d\Gamma \right\} - \lambda^2 \|\mathbf{e}_u\|_{\Omega}^2 \quad (3.88)$$

is a lower error bound in energy norm for any $\lambda \in \mathbb{R}$.

Proof. Considering the following expression in [73] and considering the residuals of internal equilibrium \mathbf{s}_σ^* and boundary equilibrium \mathbf{r}_σ^* of the recovered field, we can write:

$$\bar{a}(\boldsymbol{\sigma}_\sigma^*, \sigma(\mathbf{v})) = l(\mathbf{v}) + \int_{\Omega} (\mathbf{s}_\sigma^*)^T \mathbf{v} \, d\Omega + \int_{\Gamma} (\mathbf{r}_\sigma^*)^T \mathbf{v} \, d\Gamma \quad \forall \mathbf{v} \in V \quad (3.89)$$

Subtracting $\bar{a}(\boldsymbol{\sigma}^h, \sigma(\mathbf{v}))$ from (3.89) we write

$$\begin{aligned} \bar{a}(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h, \sigma(\mathbf{v})) &= l(\mathbf{v}) + \int_{\Omega} (\mathbf{s}_\sigma^*)^T \mathbf{v} \, d\Omega + \int_{\Gamma} (\mathbf{r}_\sigma^*)^T \mathbf{v} \, d\Gamma - \bar{a}(\boldsymbol{\sigma}^h, \sigma(\mathbf{v})) \\ &= R(\mathbf{v}) + \int_{\Omega} (\mathbf{s}_\sigma^*)^T \mathbf{v} \, d\Omega + \int_{\Gamma} (\mathbf{r}_\sigma^*)^T \mathbf{v} \, d\Gamma \quad \forall \mathbf{v} \in V \end{aligned} \quad (3.90)$$

and considering $R(\mathbf{v}) := \bar{a}(\sigma(\mathbf{e}), \sigma(\mathbf{v})) = a(\mathbf{e}, \mathbf{v})$

$$\bar{a}(\sigma(\mathbf{e}), \sigma(\mathbf{v})) = \bar{a}(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h, \sigma(\mathbf{v})) - \int_{\Omega} (\mathbf{s}_\sigma^*)^T \mathbf{v} \, d\Omega - \int_{\Gamma} (\mathbf{r}_\sigma^*)^T \mathbf{v} \, d\Gamma \quad \forall \mathbf{v} \in V \quad (3.91)$$

Considering that \mathbf{e}_u is continuous because both, the FE solution and the recovered solutions are continuous, and considering also that both fulfill the Dirichlet boundary conditions, then \mathbf{e}_u remains into V , the standard solution space. Then we can substitute by $\mathbf{v} = \mathbf{e}_u$ in the previous equation:

$$\bar{a}(\boldsymbol{\sigma}(\mathbf{e}), \boldsymbol{\sigma}(\mathbf{e}_u)) = \bar{a}(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h, \boldsymbol{\sigma}(\mathbf{e}_u)) - \int_{\Omega} (\mathbf{s}_\sigma^*)^T \mathbf{e}_u \, d\Omega - \int_{\Gamma} (\mathbf{r}_\sigma^*)^T \mathbf{e}_u \, d\Gamma \quad (3.92)$$

This allows us to evaluate a lower bound following the ideas presented in [65]. Expanding the following symmetric bilinear form $0 \leq \bar{a}(\boldsymbol{\sigma}(\mathbf{e}) - \lambda \boldsymbol{\sigma}(\mathbf{e}_u), \boldsymbol{\sigma}(\mathbf{e}) - \lambda \boldsymbol{\sigma}(\mathbf{e}_u))$ for any $\lambda \in \mathbb{R}$:

$$0 \leq \bar{a}(\boldsymbol{\sigma}(\mathbf{e}), \boldsymbol{\sigma}(\mathbf{e})) + \lambda^2 \bar{a}(\boldsymbol{\sigma}(\mathbf{e}_u), \boldsymbol{\sigma}(\mathbf{e}_u)) - 2\lambda \bar{a}(\boldsymbol{\sigma}(\mathbf{e}), \boldsymbol{\sigma}(\mathbf{e}_u)) \quad (3.93)$$

Substituting the last term by expression (3.92) and rearranging terms we proof the theorem:

$$2\lambda \left\{ \bar{a}(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h, \boldsymbol{\sigma}(\mathbf{e}_u)) - \int_{\Omega} (\mathbf{s}_\sigma^*)^T \mathbf{e}_u \, d\Omega - \int_{\Gamma} (\mathbf{r}_\sigma^*)^T \mathbf{e}_u \, d\Gamma \right\} - \lambda^2 |||\mathbf{e}_u|||_{\Omega}^2 \leq |||\mathbf{e}|||_{\Omega}^2 \quad (3.94)$$

□

The previous expression is valid for any $\lambda \in \mathbb{R}$. Differentiating with respect to λ we obtain the expression for the optimum λ and thus, the optimum lower bound.

$$\mathcal{E}_{\text{SPR-CD}_{LB_{Opt}}}^2 = \frac{\left\{ a(\mathbf{e}_\sigma, \mathbf{e}_u) - \int_{\Omega} (\mathbf{s}_\sigma^*)^T \mathbf{e}_u \, d\Omega - \int_{\Gamma} (\mathbf{r}_\sigma^*)^T \mathbf{e}_u \, d\Gamma \right\}^2}{|||\mathbf{e}_u|||_{\Omega}^2} \leq |||\mathbf{e}|||_{\Omega}^2 \quad (3.95)$$

Since (3.94) is valid for any $\lambda \in \mathbb{R}$ another possibility is to consider $\lambda = 1$ obtaining a simpler expression that will be used afterwards:

$$\begin{aligned} \mathcal{E}_{\text{SPR-CD}_{LB_1}}^2 &= 2 \left\{ \bar{a}(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h, \boldsymbol{\sigma}(\mathbf{e}_u)) - \int_{\Omega} (\mathbf{s}_\sigma^*)^T \mathbf{e}_u \, d\Omega - \int_{\Gamma} (\mathbf{r}_\sigma^*)^T \mathbf{e}_u \, d\Gamma \right\} \\ &\quad - |||\mathbf{e}_u|||_{\Omega}^2 \leq |||\mathbf{e}|||_{\Omega}^2 \end{aligned} \quad (3.96)$$

3.5 Numerical results

In this Section we present the results concerning error estimation in the energy norm and bounding techniques presented before. We have chosen a set of benchmark problems for testing the methods. The results are obtained by performing the analysis with the cgFEM code and recovering the solution with the SPR-CD technique. Results are presented for both, bi-linear (Q4) and bi-quadratic (Q8) square elements.

The main objective of this Section is to check the behavior of both, the proposed recovery procedure SPR-CD and also the behavior of the proposed error estimators and error bounds in energy norm. Both quantities are closely related since the ZZ error estimator (2.28) is used. We will check the behavior of both equations by checking the local and global results of the error estimation.

Remember that, as shown previously, for the right convergence of the FE solution, we need, at least, the same degree for solution interpolation than the degree used for the geometry interpolation. For the results presented in this Section we have used a quadratic approximation to the triangular-shape integration subdomains used in elements cut by the boundary as described in Section 2.3.7.

In order to check the reliability of the combination of the cgFEM and the recovery procedure we solve a set of problems with known analytical solution and compared the cgFEM solution with the exact one. In these situations, we compare them locally and at a global level. However, in the vast majority of the situations the exact solution is not available. Then, we evaluate the convergence of the error in energy norm \mathcal{E} to measure the quality of the recovery process and the error estimator.

To measure the quality of the solution when the exact solution is available, we define the global effectivity index θ (3.97) and the corresponding local

effectivity index D (3.98):

$$\theta = \frac{\mathcal{E}}{\|e\|_{\Omega}} \quad (3.97)$$

$$D = \begin{cases} \theta_K - 1 & \text{if } \theta_K > 1 \\ 1 - \frac{1}{\theta_K} & \text{if } \theta_K \leq 1 \end{cases} \quad (3.98)$$

where θ_K is the effectivity index evaluated only at each element $K \in \mathcal{T}$. We will also consider the mean value of the local effectivity index $m(D)$ and its standard deviation $\sigma(|D|)$ as a quality measure.

Once the error in energy norm \mathcal{E} has been estimated at each element, it will be used to guide the *h-adaptive* refinement process. The refinement of the mesh using the error estimate as the guiding parameter considers a stopping criterion that checks the value of the global estimated error against the prescribed or desired error. If the estimated error is higher than the desired error then the mesh is refined. Several procedures to perform the refinement are available in the literature. To define the size of the elements in the new mesh we follow the adaptive process described in [114, 115, 81] which minimizes the number of elements in the new mesh for a given accuracy level. This criterion is equivalent to the traditional approach of equally distributing the error in each element of the new mesh as shown in [116, 55]. This criterion provides the size of the elements in the new mesh as a function of *i*) the ratio of the estimated error in energy norm in the current mesh to the desired error in the new mesh, and *ii*) the estimated error in energy norm in each element, which always takes non negative values.

The refinement technique provides a distribution of the required new element sizes. These sizes are specified in each element of the current mesh, which will be recursively split into 4 new elements until the sizes of the elements are smaller than the required size. A maximum difference of only one refinement level will be allowed between adjacent elements. In these cases C^0 continuity will be enforced by means of the use of MPC [49, 50].

3.5.1 Problems with exact solution

In this Section we will present a set of benchmark problems with known analytical solution in order to analyze the reliability of the cgFEM and the proposed error estimation techniques. First we will present all the problems and then we will analyze their results all together.

Problem definitions

Problem 1a: Square domain with a 3^{rd} order solution The First benchmark problem considers an infinite domain problem where we have extracted a 2×2 square domain, centered at the origin of the reference system. The exact displacement solution consists in a 3^{rd} order polynomial, yielding linear body loads over the domain. We have imposed the corresponding Neumann boundary conditions. A model of the problem and the analytical solution considering plane strain conditions are shown in Figure 3.11.

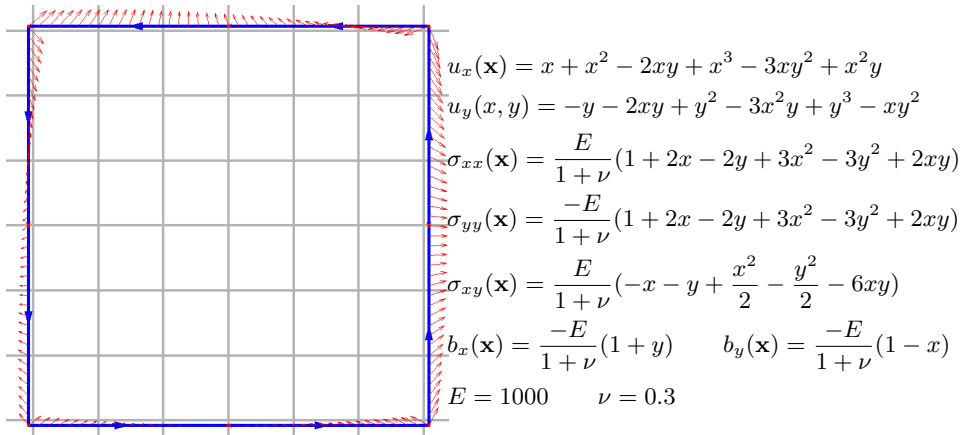
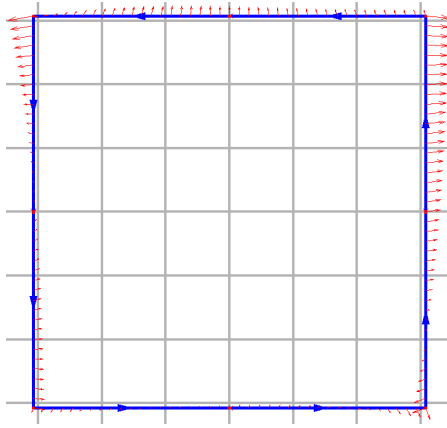


Figure 3.11: Problem 1a. Problem model and analytical solution.

Problem 1b: Square domain with a 4rd order solution This problem also considers the 2×2 domain as the previous one but in this case the exact displacement solution consists in a 4rd order polynomial, yielding quadratic body loads over the domain. A model of the problem and the analytical solution considering plane strain conditions are shown Figure 3.12.



$$\begin{aligned}
 u_x(x, y) &= x^4 + 5x^3y - 3x^2y^2 + x^3 \\
 u_y(x, y) &= y^4 - 6y^2x^2 + 3yx^3 + 2y \\
 A(\mathbf{x}) &= 4x^3 + 15x^2y - 6xy^2 + 3x^2 \\
 B(\mathbf{x}) &= 4y^3 - 12x^2y + 3x^3 + 2 \\
 C &= \frac{E}{(1 + \nu)(1 - 2\nu)} \\
 D(\mathbf{x}) &= 5x^3 + 3x^2y - 12xy^2 \\
 \sigma_{xx}(\mathbf{x}) &= C((1 - \nu)A(\mathbf{x}) + \nu B(\mathbf{x})) \\
 \sigma_{yy}(\mathbf{x}) &= C(\nu A(\mathbf{x}) + (1 - \nu)B(\mathbf{x})) \\
 \sigma_{xy}(\mathbf{x}) &= C\frac{1 - 2\nu}{2}D(\mathbf{x})
 \end{aligned}$$

$$\begin{aligned}
 b_x(x, y) &= \frac{-3E}{2(1 + \nu)(2\nu - 1)}9x^2 - 12xy + 4y^2 - 4x + \nu(4x^2 + 20xy - 4y^2 + 4x) \\
 b_y(x, y) &= \frac{3E}{2(1 + \nu)(2\nu - 1)}4y^2 - 3x^2 + 2xy + \nu(8x^2 - 12xy) \\
 E &= 1000 \quad \nu = 0.3
 \end{aligned}$$

Figure 3.12: Problem 1b. Problem model and analytical solution.

Problem 2: Thick-wall cylinder subjected to internal pressure The geometrical model for this problem is represented in Figure 3.13. Only $1/4$ of the section is modeled together with the appropriate symmetry boundary conditions. The internal and external surfaces are of radius a and b with $a = 5$, $b = 20$. Young's modulus is $E = 1000$, Poisson's ratio is $\nu = 0.3$ and the load $P = 1$.

The exact solution for the radial displacement assuming plane strain conditions is given by:

$$u_r(r) = \frac{P(1 + \nu)}{E(c^2 - 1)} \left(r(1 - 2\nu) + \frac{b^2}{r} \right) \quad (3.99)$$

where $c = b/a$, $r = \sqrt{x^2 + y^2}$ and $\phi = \arctan(y/x)$. The stresses in cylindrical coordinates are given by:

$$\sigma_r(r) = \frac{P}{c^2 - 1} \left(1 - \frac{b^2}{r^2} \right) \quad \sigma_\phi(r) = \frac{P}{c^2 - 1} \left(1 + \frac{b^2}{r^2} \right) \quad \sigma_z = 2\nu \frac{P}{c^2 - 1} \quad (3.100)$$

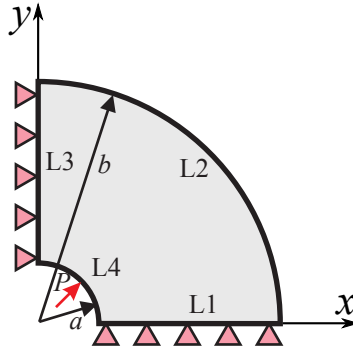


Figure 3.13: Problem 2. Thick-wall cylinder subjected to internal pressure.

Problem 3: L-Shape plate Let us consider the singular problem of a finite portion of an infinite domain with a reentrant corner. The model is loaded on the boundary with the tractions corresponding to the first terms of the asymptotic expansion that describes the exact solution under mixed mode loading conditions around the singular vertex, see Figure 3.14. The exact values of boundary tractions on the boundaries represented by discontinuous thick lines have been imposed in the FE analyses.

The exact displacement and stress fields (3.57) for this singular elasticity problem can be found in [112]. Exact values of the GSIF [112] under mixed mode have been taken as $K_I = 1$ and $K_{II} = 1$. The material parameters are Young's modulus $E = 1000$ and Poisson's ratio $\nu = 0.3$. The analytical solution of this

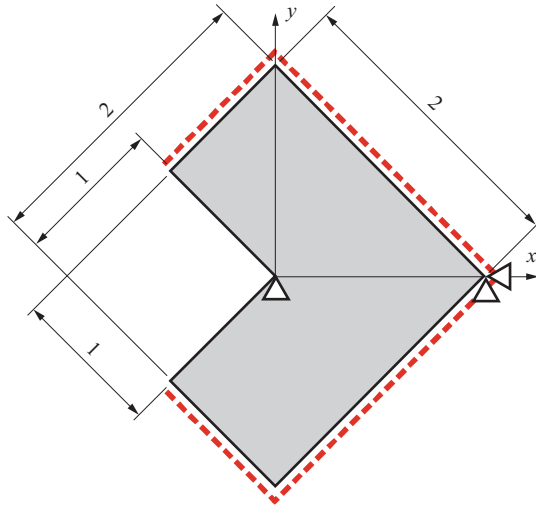


Figure 3.14: Problem 3. L-shaped domain.

problem is singular at the reentrant corner of the plate. Therefore we will apply the *singular+smooth* decomposition for the stress recovery as explained in Section 3.3. We will use a domain integral method based on extraction functions to obtain an approximation of the recovered singular part as explained in Section 3.3.

Convergence analysis

The convergence results and the corresponding convergence rate for both Q4 (blue line) and Q8 (red line) elements have been studied. This convergence study has been carried out with h -uniform refinement is presented in Figure 3.15. Each of the problems is indicated with a small picture included into the graph. On the left column we present the convergence results for all the problems. The quantity plotted is the relative exact error in energy norm of the FE solution $R\|e\|_{\Omega}$. We can observe that we have obtained a smooth convergence in all situations.

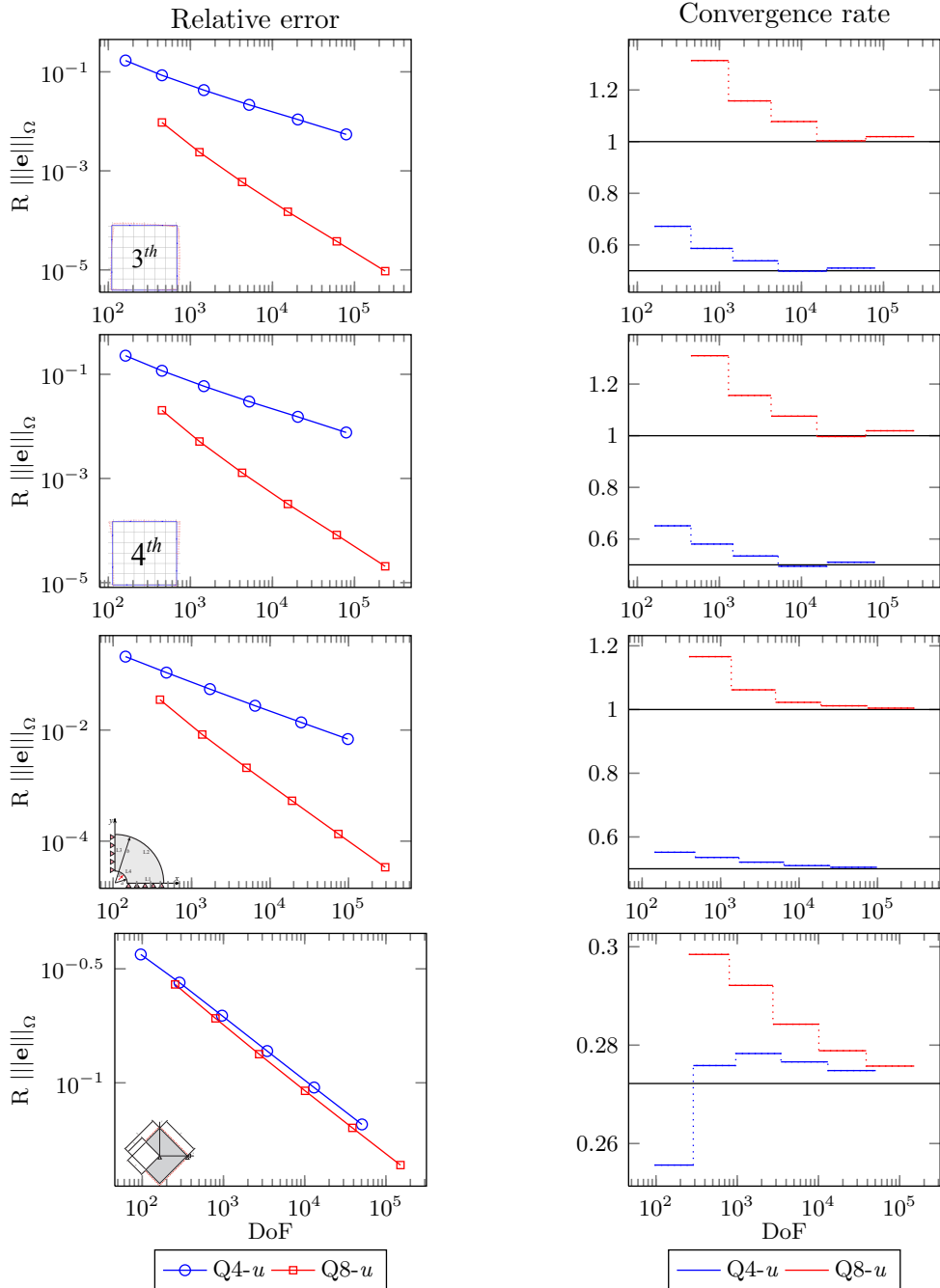
On the right column of the Figure 3.15 we present the convergence rate of the exact error in energy norm of the FE solution as a function of the number of Degrees of Freedom (DoF). For problems with non-singular solution the theoretical convergence rate is 1 for Q8 elements and 0.5 for Q4 elements. In the graphics these convergence rates are indicated with black lines. We observe that for all problems solution the convergence rate tends asymptotically to the expected values. Note that for the L-Shape problem, the convergence rate tends to a value slightly below 0.28, as expected. For these kind of problems the convergence rate, under a h -uniform refinement process, is defined by the minimum value between the interpolation degree and the intensity of the singularity which is 0.2722 in this case.

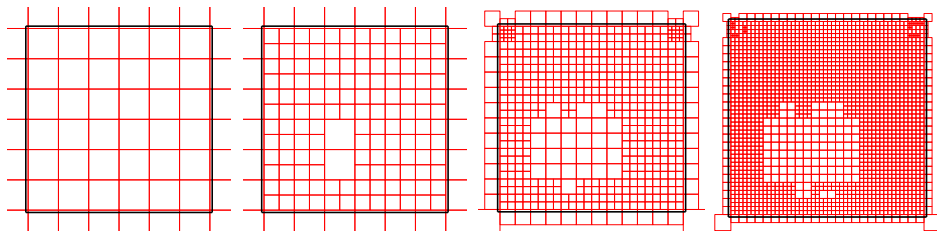
h -adaptive refinement process

We have also performed a h -adaptive refinement process guided by the local estimation in energy norm $\mathcal{E}_{\text{SPR-CD}}$. First we present the set of the first 4 meshes for each problem. Note that for Problem 2, we would expect the refinement process to provide symmetric meshes with respect to line $x = y$ given the solution and geometry symmetry. However we observe a lack of symmetry from the second mesh that tends to disappear. This is because of the lack of symmetry during the imposition of the boundary equilibrium equation (3.39), that as explained before, is only imposed in one curve when the patch is cut by more than one curve.

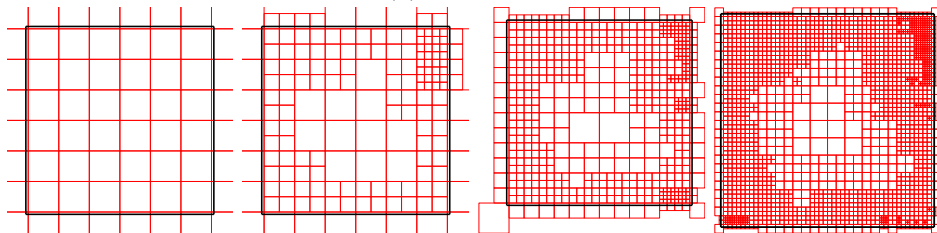
Global effectivity index for the h -adaptive refinement process

The global effectivity index of the proposed error estimation technique provides very accurate results. Tables 3.1 and 3.2, for Q4 and Q8 elements respectively, shows the global effectivity index θ for the four problems. We easily

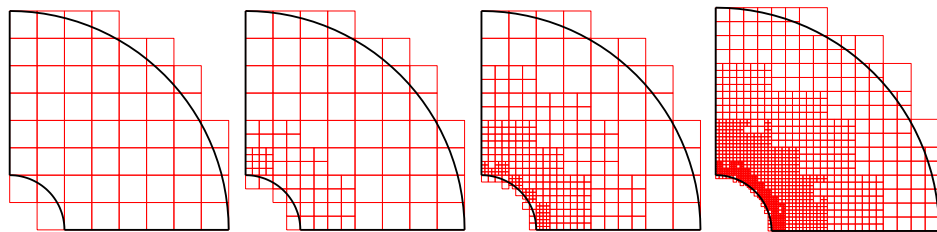
Figure 3.15: Convergence analysis for h -uniform refinement.



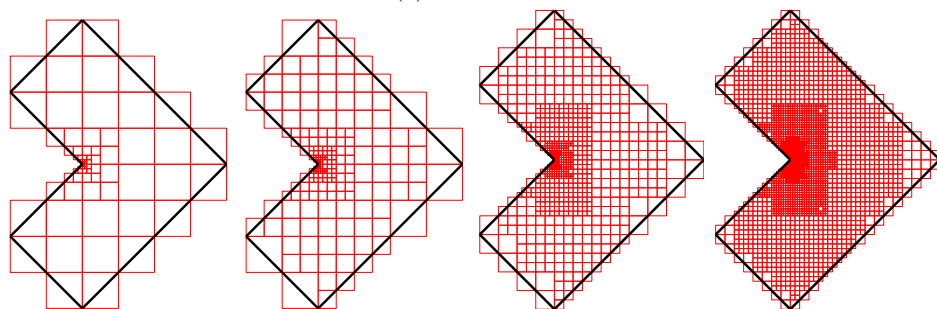
(a) Problem 1a



(b) Problem 1b

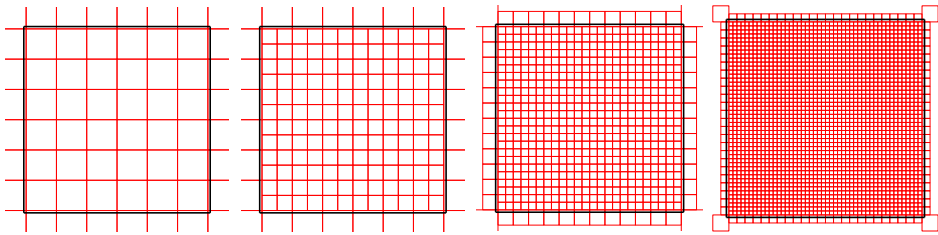


(c) Problem 2

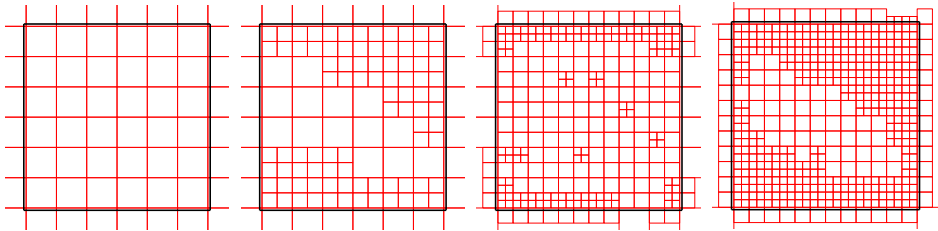


(d) Problem 3

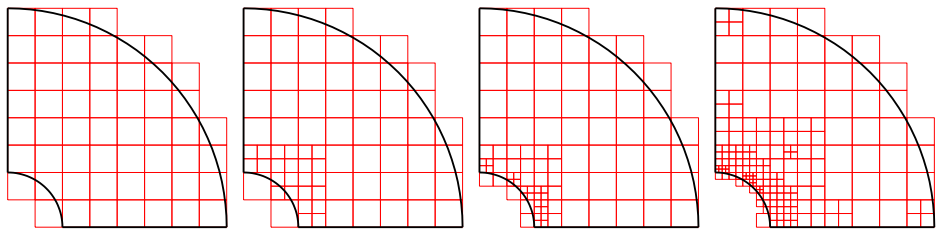
Figure 3.16: Q4. First four meshes of the h -adaptive refinement process.



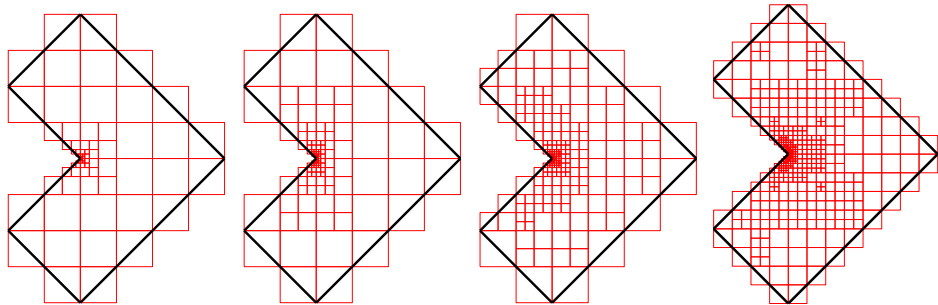
(a) Problem 1a



(b) Problem 1b



(c) Problem 2



(d) Problem 3

Figure 3.17: Q8. First four meshes of the h -adaptive refinement process.

observe that the values tend to 1 when increasing the mesh refinement by using the h -adapted refinement process.

Figure 3.18 shows the behavior of the bounding techniques. Black lines represent the guaranteed upper error bound and the brown ones the guaranteed lower error bound of the error in energy norm. We have to remark that while the lower error bound $\mathcal{E}_{\text{SPR-CD}_{\text{LB}_{\text{Opt}}}}$ is fully evaluable, the upper bound $\mathcal{E}_{\text{SPR-CD}_{\text{UB}}}$ does need the evaluation of correction terms that depend on the exact solution, then it is not available for the vast majority of problems. To overcome this difficulty we propose the use of $\hat{\mathcal{E}}_{\text{SPR-CD}_{\text{UB}}}$, the numerical approximation to the actual upper bound, given by (3.87). This is just an improvement of the standard error estimator. We observe that this last estimator provides better results, in general, than the plain error estimator $\mathcal{E}_{\text{SPR-CD}}$.

We should observe that for Q4 elements, left column in Figure 3.18, the error estimator $\mathcal{E}_{\text{SPR-CD}}$ and the numerical version of the upper bound take effectivity values below 1 in all cases, but with very accurate results, smoothly converging to one when refining. Therefore these error estimators are asymptotically exact. The same occurs with the guaranteed upper $\mathcal{E}_{\text{SPR-CD}_{\text{UB}}}$ and lower bounds $\mathcal{E}_{\text{SPR-CD}_{\text{LB}_{\text{Opt}}}}$.

For Q8 elements, right column in Figure 3.18, we observe that for the three last problems, the plain error estimation $\mathcal{E}_{\text{SPR-CD}}$ and the guaranteed upper bound (black line) provide quite similar results. In the first Problem, we observe that the natural trend for both, the error estimator and the numerical approximation to the upper bound is to tend to the guaranteed upper bound. In other words, we observe in this Figure that for Q8 elements the natural tendency of the proposed error estimator is to provide upper error bounds when we refine the mesh. Finally, the lower error bound provides a good behavior for the vast majority of situations, specially for finer meshes.

Note that for Problem 1b, 4th order plate, the results of the guaranteed upper bound, $\mathcal{E}_{\text{SPR-CD}_{\text{UB}}}$, shows small underestimations. This is because there is a

lack of internal equilibrium that is not taken into account because the body forces in this problem are represented by a quadratic polynomial and the recovered stress field is only able to represent linear body forces, as explained in Section 3.3.

Table 3.1: Q4h. Values for the global effectivity θ considering h -adapted meshes.

DoF	$\theta_{\text{SPR-CD}}^{\text{P1a}}$	DoF	$\theta_{\text{SPR-CD}}^{\text{P1b}}$	DoF	$\theta_{\text{SPR-CD}}^{\text{P2}}$	DoF	$\theta_{\text{SPR-CD}}^{\text{P3}}$
162	0.9975828	162	0.9808512	144	1.1809610	246	0.8828344
396	0.9890647	434	0.9957558	256	0.9990164	728	0.9520967
1,316	0.9910361	1,378	0.9878902	780	0.9915373	2,320	0.9721498
4,544	0.9948067	4,548	0.9915437	2,836	0.9907888	8,324	0.9832810
16,902	0.9972978	15,964	0.9946252	10,648	0.9926603	31,414	0.9907382
66,706	0.9970884	58,692	0.9953141	40,728	0.9954087	122,982	0.9938090

Table 3.2: Q8h. Values for the global effectivity θ considering h -adapted meshes.

DoF	$\theta_{\text{SPR-CD}}^{\text{P1a}}$	DoF	$\theta_{\text{SPR-CD}}^{\text{P1b}}$	DoF	$\theta_{\text{SPR-CD}}^{\text{P2}}$	DoF	$\theta_{\text{SPR-CD}}^{\text{P3}}$
450	1.0036098	450	1.0427583	396	1.8271363	654	1.2126743
1,146	0.9747222	918	1.0982516	528	1.7001104	1,238	1.2445716
3,994	0.9857416	2,028	1.0262182	684	1.2059119	2,046	1.2391891
14,842	0.9937332	3,222	1.0152258	1,326	1.1264376	4,644	1.1351134
57,274	1.0000077	4,844	1.0156769	2,122	1.1241507	7,582	1.0998022
227,386	1.0025933	11,160	1.0060170	4,434	1.0357824	12,494	1.0683767
–	–	15,898	1.0070371	6,970	1.0232700	26,816	1.0751757

We also observe in Figure 3.18 that for the Problem 3 and Q8 elements the error estimation deteriorates in the last mesh of the sequence, this being represented by effectivity values moving away from the ideal value $\theta = 1$. This is produced because of the pollution error introduced due to the insufficient refinement near the singular point. The cgFEM code has a limit for the smallest size of the element as explained in Section 2.3. When this limit is reached during the refinement process (as in this case when trying to get increasingly refined meshes as we move towards the singularity) uniform meshes are produced, thus introducing pollution error.

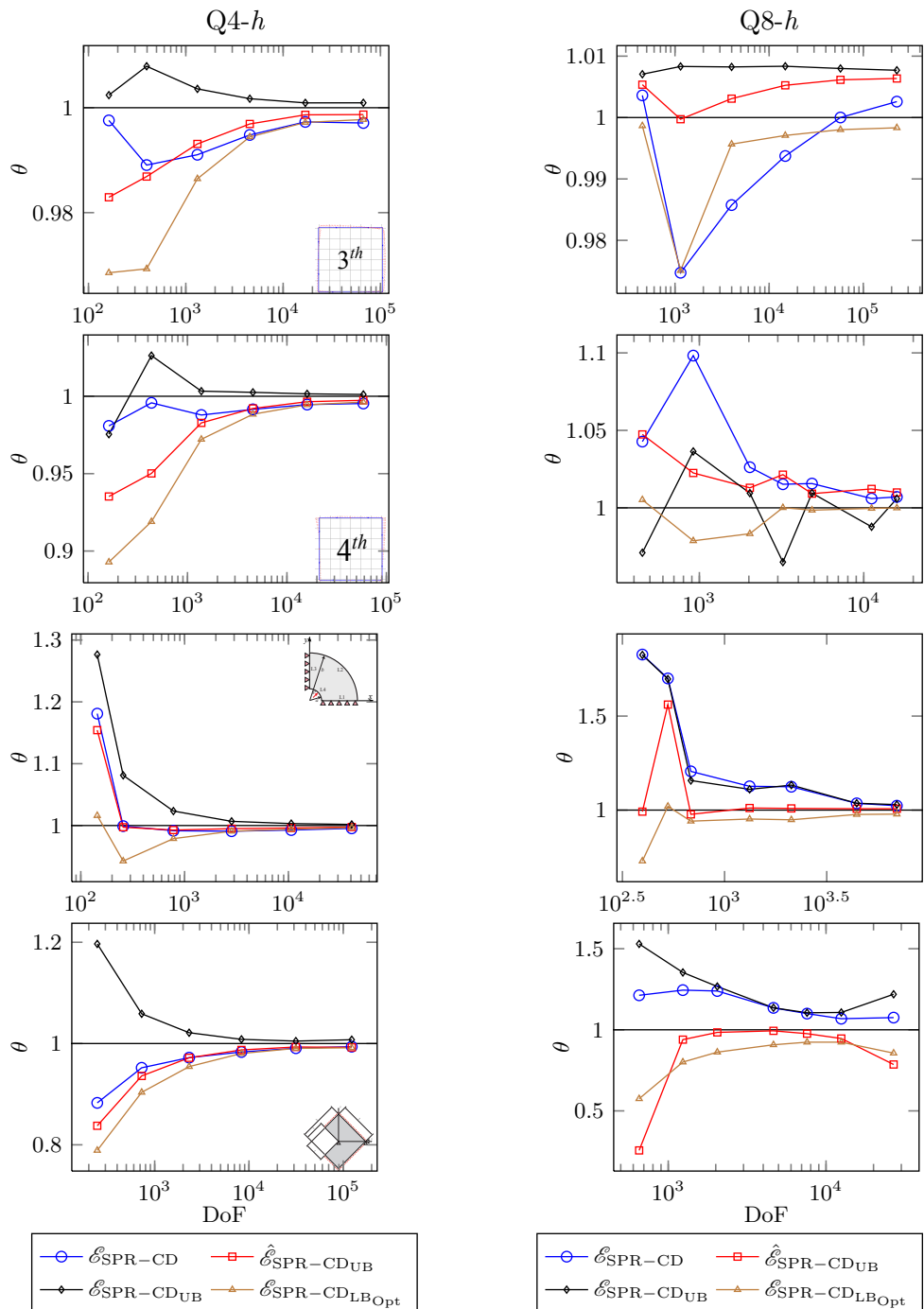


Figure 3.18: Effectivity of the error estimators for h -adaptive refinement.

Local effectivity index for the h -adaptive refinement process

In the previous Subsection we have presented the results in a global sense. However, the h -adaptive refinement process relies on the local error estimation. Therefore it is also important to assess the behavior of the estimator at the local level. To evaluate the accuracy of our local error estimations we have defined the local effectivity index D (3.98) taking positive values for overestimations and negative values for underestimations of the exact error in energy norm at element level. The mean value $m(D)$ and the standard deviation $\sigma(D)$ should tend to zero with mesh refinement and be as close to zero as possible. The evolution of these parameters is presented in Figure 3.19. In that Figure, we have compared the values obtained with the proposed recovery process SPR-CD (blue lines for Q4 and black lines for Q8) and the previous version based on stress smoothing SPR-C (red lines for Q4 and brown lines for Q8).

In this case we observe that behavior for both, Q4 and Q8 elements, tends to zero, for the SPR-CD technique, as expected. We also observe that the results for the proposed technique are, in general, of the same quality as for the SPR-C technique, or in some situations, even better. For instance, in Problem 1a, for Q8 elements the mean value of D for the SPR-C seems not to converge, with the SPR-CD technique we obtain a sharp convergence trend. The SPR-C technique have already been compared with the SPR technique in several works [71, 74] reporting a local improvement.

In Figure 3.20 and Figure 3.21 we present the local effectivity maps for Q4 and Q8 elements, respectively. We observe that when the refinement increases the extreme values tend to get closer to zero, thus the local error indicator becomes more accurate, providing better h -adapted meshes.

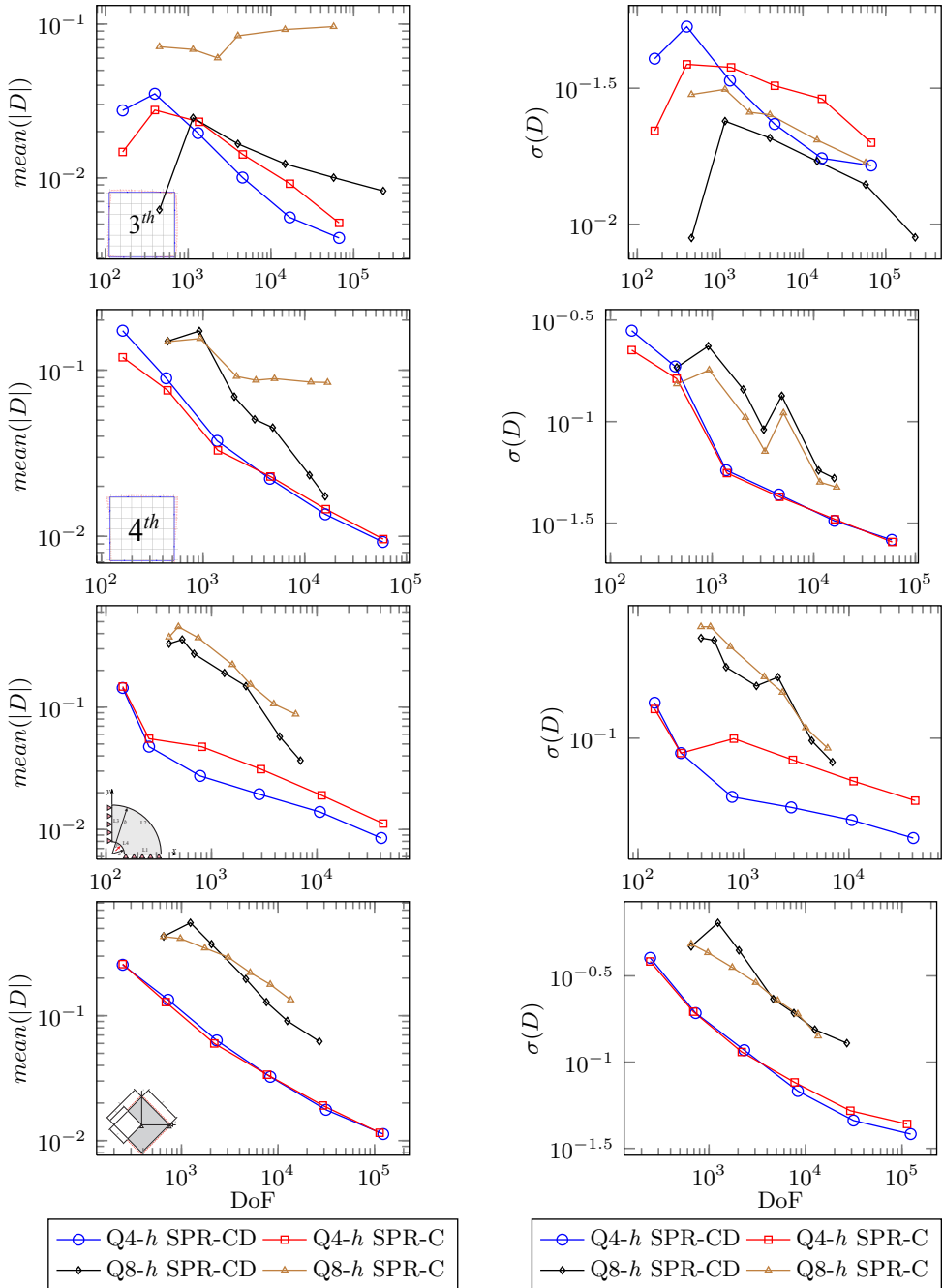
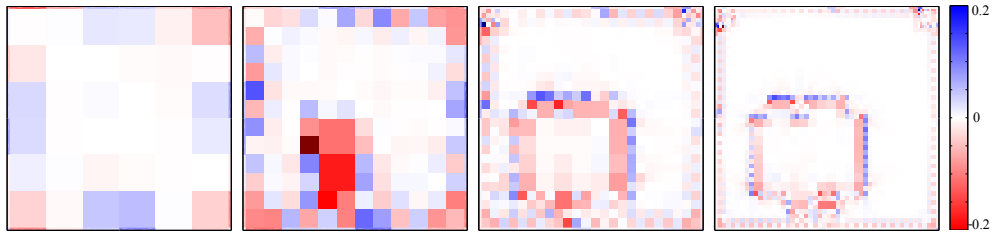
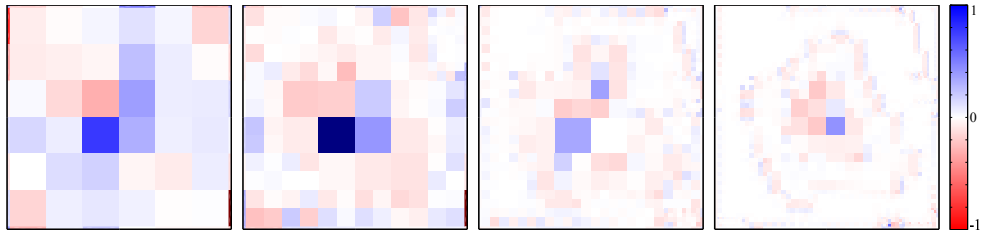


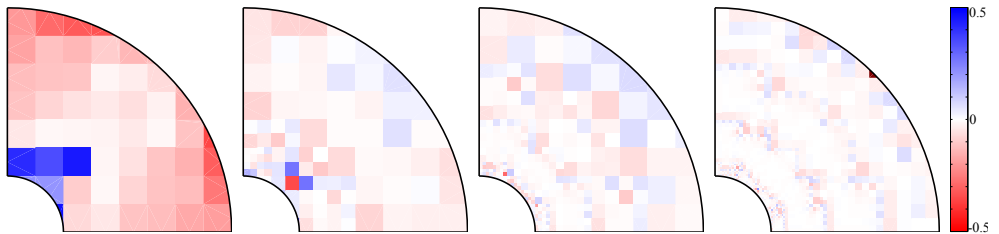
Figure 3.19: Local effectivity indicators for h -adaptive refinement.



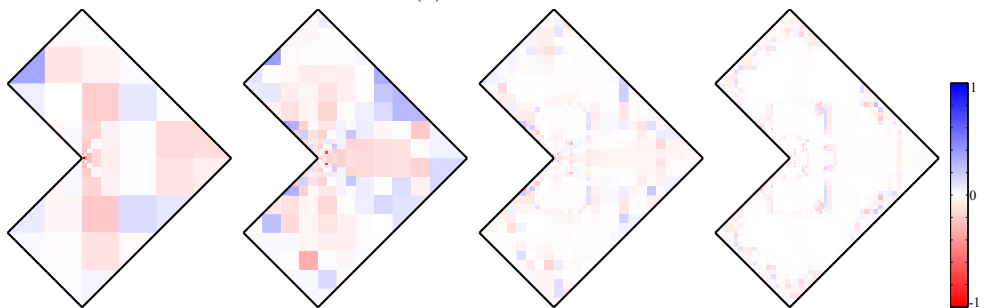
(a) Problem 1a



(b) Problem 1b

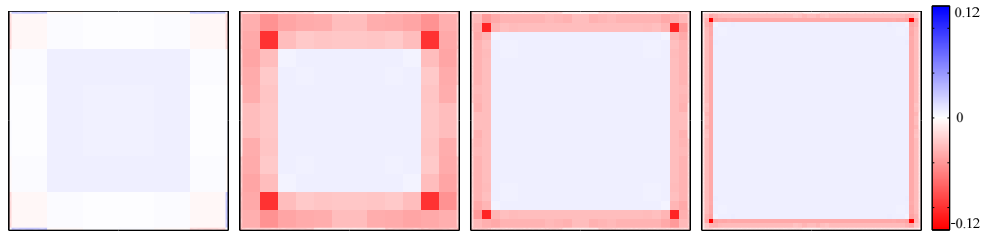


(c) Problem 2

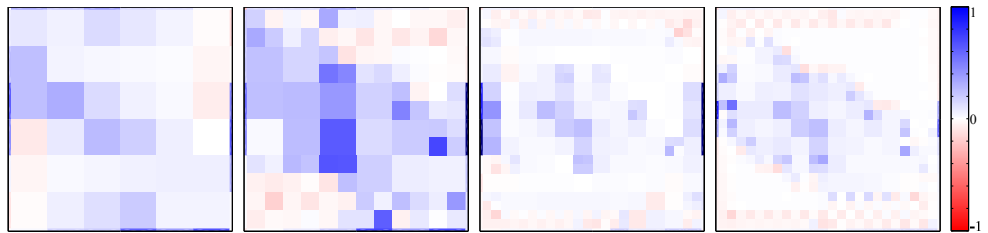


(d) Problem 3

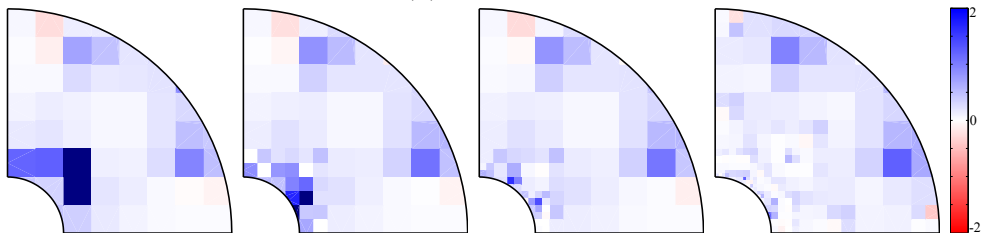
Figure 3.20: Q4. SPR-CD. Local effectivity indicator D for the h -adaptive refinement process.



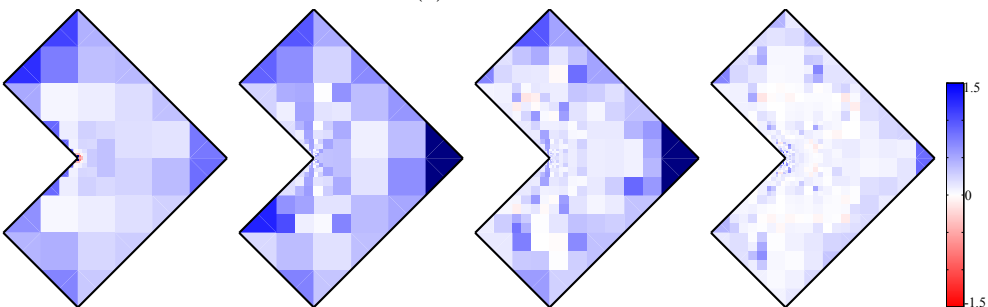
(a) Problem 1a



(b) Problem 1b



(c) Problem 2



(d) Problem 3

Figure 3.21: Q8. SPR-CD. Local effectivity indicator D for the h -adaptive refinement process.

3.5.2 Problems without analytical solution

In this Section our purpose is to show the performance of the h -adaptive refinement process in more complex geometries. Naturally, in these kind of problems there is no available exact solution, then we will check the behavior of the method by measuring the convergence rate of the estimated error in energy norm of the FE solution [76, 77].

Problem definitions

Problem 4: Gravity dam This problem represents a cross section of a gravity dam. In this case, the dam is under hydrostatic pressure on the left side. The displacement normal to the three inferior boundaries is constrained, as represented in Figure 3.22. The material of the dam is concrete with density $\rho_{concrete} = 2300 \frac{Kg}{m^3}$. The density of the water is considered as $\rho_{water} = 1000 \frac{Kg}{m^3}$. The Young modulus of the concrete is taken as $27.5 \cdot 10^9 Pa$ and the Poisson ratio $\nu = 0.3$. Volume load due to gravity $g = 9.81 \frac{m}{s^2}$ is also considered. We assume plane strain behavior.

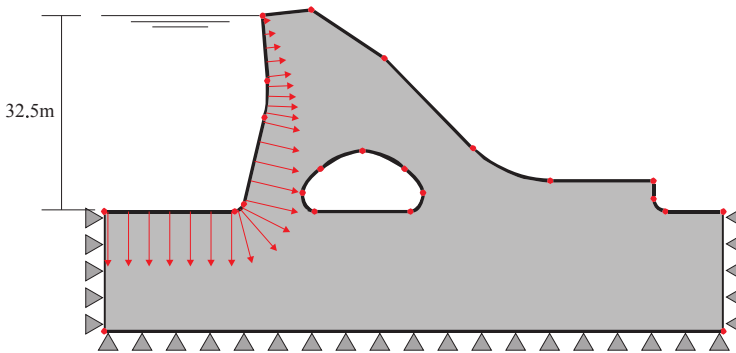


Figure 3.22: Problem 4. Gravity dam: Model of the problem, loads and constrains.

Problem 5: 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 3.23. The material is aluminum with elastic modulus $E = 70 \cdot 10^9 Pa$ and Poisson ratio $\mu = 0.33$. Plane stress conditions are considered.

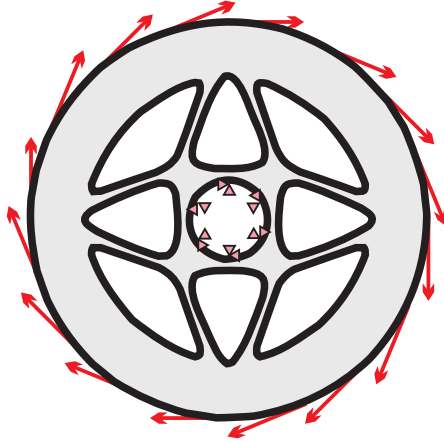


Figure 3.23: Problem 5. Flywheel: Model of the problem, loads and constrains

h-adapted meshes

Figures 3.24 and 3.25 shows the set of *h*-adapted meshes for both, Q4 and Q8 elements. We observe how, in the first mesh, the geometrical refinement process provides an initial mesh considering the geometrical details without any user intervention. Second, third and fourth meshes are guided by the standard *h*-adaptive refinement process, which is fed by the error estimation in energy norm, utilizing the recovered stress field σ_σ^* .

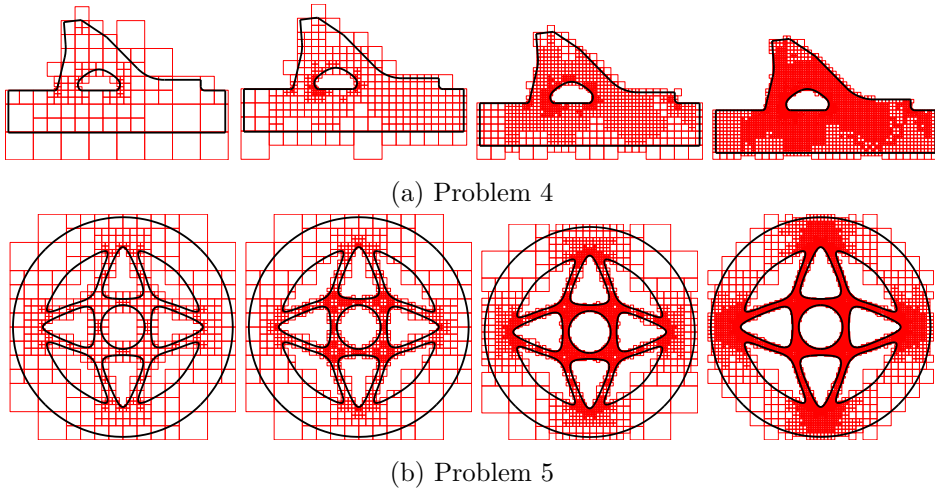


Figure 3.24: Q4. Problems without exact solution. First four meshes of the h -adaptive refinement process.

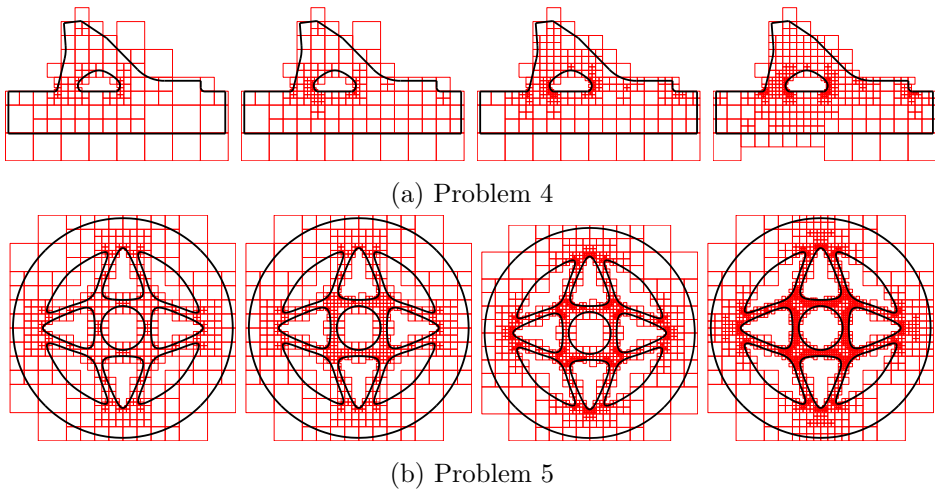


Figure 3.25: Q8. Problems without exact solution. First four meshes of the h -adaptive refinement process.

Convergence analysis for h -adapted mesh refinement

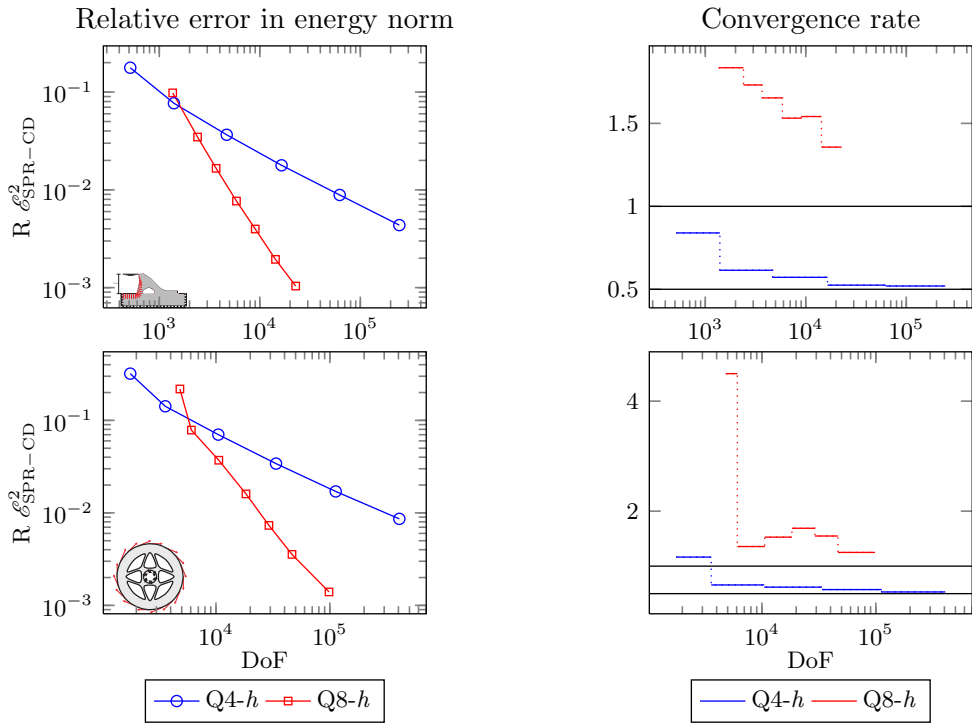
In Figure 3.26 we present the convergence analysis of these two problems without analytical solution. As the analytical solution is unknown, the only parameters in which we can rely on are the convergence rate and the smoothness of the convergence process. In both cases, for Q4 elements (blue line) we observe that around 10 thousand degrees of freedom the asymptotic behavior is achieved. We also observe that the estimated convergence rate tends to the right one, that is 0.5 for Q4 elements.

However, for Q8 elements, red lines, the asymptotic behavior seems not to be achieved that fast. But attending to the convergence rate, we observe that it tend to 1. That indicates the good behavior for both, the displacement recovered field and the ZZ error estimator for more challenging geometries.

3.6 Conclusions

In this Chapter we have presented a brief summary of different error estimation and bounding techniques. Additionally we have described in detail two methods based on the well-known SPR technique, the SPR-C introduced in [71] and the SPR-CD, briefly introduced in [117] and developed in this Chapter. The numerical results have compared both techniques in order to validate the newer one, the SPR-CD technique, with satisfactory results. The SPR-CD, introduced in this Chapter, provides the recovered solution $(\mathbf{u}_u^*, \boldsymbol{\sigma}_\sigma^*)$ which is, in general, more accurate than the FE solution pair $(\mathbf{u}_h, \boldsymbol{\sigma}_h)$. Therefore we propose to substitute $(\mathbf{u}_h, \boldsymbol{\sigma}_h)$ by $(\mathbf{u}_u^*, \boldsymbol{\sigma}_\sigma^*)$ as the standard output of the FE code. An error estimator for this pair will be presented in Section 6.

Both techniques, $\mathcal{E}_{\text{SPR-C}}$ and $\mathcal{E}_{\text{SPR-CD}}$, provides very accurate results when they are used as error estimator. Additionally the $\mathcal{E}_{\text{SPR-CD}}$, can also be used

Figure 3.26: Convergence analysis for h -adaptive refinement.

to evaluate an accurate the lower error bound, increasing its versatility in comparison with traditional stress-based recovery procedures. However, the numerical approach for the upper bound of the error in energy norm proposed in this Chapter does not always provides upper bounds and further research will be needed. In Chapter 5 we will investigate other techniques that provide guaranteed upper error bounds.

Chapter 4

Goal Oriented Adaptivity

4.1 Introduction and motivation

Traditionally error estimators are based on the evaluation, in one or another way, of an approximation to the true error in energy norm of a problem. However, this quantity is generally useless for practitioners. In practice, engineers and final users run simulations in order to evaluate stresses, displacements, *etc...* in an area of the domain, but they do not generally care about the energy norm involved in the problem. In the late 90s, a new paradigm appeared [64, 118, 85]. In this case, instead of evaluating the error of the solution in terms of energy norm, the error in a Quantity of Interest (QoI) into a Domain of Interest (DoI) was evaluated. That is, some relevant quantity for the final user is selected as the main output. Then, instead of using the error in energy norm to evaluate the accuracy of the FE solution, we directly control the error in the QoI into the DoI which actually is more useful for practitioners. The error estimation of a QoI requires the evaluation of two problems simultaneously. The first one, so-called primal problem, is the one we are interested in. The second problem, so-called dual or adjoint problem, is used to extract the

information for the error in the QoI. Both problems are geometrically identical, but subjected to different loading conditions. The loads of the dual problem depend only on the DoI and the QoI and have no relation with the loads of the primal problem. The construction of the dual problem will be explained later in more detail. Subsequently, Oden and Prudhomme [119] introduced the basis for the error bounding in QoI. This bounding technique is based on the evaluation of upper and lower bounds of the error in energy norm for a combination of the primal and the dual problems. These bounds are usually obtained by using residual based error estimators that directly provides upper and lower error bounds in energy norm as shown in Chapter 3.

The main difference between the approach proposed in this thesis to obtain highly accurate error estimations and bounds in QoI and previous ones is that the proposed technique is based on the use of recovered fields both, for the primal and the dual problems. The proposed technique starts with the evaluation of recovered displacement fields considering the fulfillment of boundary and internal equilibrium equations, Dirichlet constraints and, for singular problems, the splitting of the displacement and stress fields into singular and smooth parts, as described in Section 3.3. Similar stress based recovery techniques were previously used to obtain upper bounds of the error in energy norm [73, 74]. However, enforcing continuity over the locally equilibrated stress fields evaluated on patches introduced a lack of equilibrium which had to be taken into account by using correction terms. The evaluation of these correction terms requires approximations of the exact error in the displacements that were obtained using projection techniques, which led to a higher computational effort. To overcome this difficulty, we use a recovery procedure based on the displacement field, introduced in Chapter 3, to directly obtain an estimation of the error in the displacements, thus easing the evaluation of the correction terms.

First, the recovered displacements are used to obtain nearly statically admissible recovered fields for the primal and the dual problems. Then, we evaluate a compatible stress field using the recovered continuous displacement field that fulfills the essential boundary conditions. With these two recovered stress

fields we can evaluate lower error bounds (adapting the procedure presented in [65]), computed versions of the upper error bounds, and very accurate error estimates for different QoI.

One particular feature of the proposed approach is that the error bounds for the QoI are given in terms of the errors in energy norm for the primal and dual solutions, following the ideas presented by Oden and Prudhomme [119]. After obtaining the solutions for the primal and dual problems we evaluate recovered fields for both problems using our equilibrated recovery technique. For the dual problem, we must define analytical expressions that describe the loads and required by the equilibrating recovery process. Once the recovered fields are obtained we evaluate correction terms and the error estimates in energy norm for the dual and primal solutions. In this work we propose the use of a technique to approximate lower and upper bounds for the QoI at hand, based on the methodology presented in [119].

In order to investigate the quality of the proposed technique numerical tests are performed using 2D benchmark problems with exact solution. The results for different quantities of interest show that the technique provides numerical error bounds and sharp error estimates that can be used in goal oriented adaptive (GOA) procedures.

4.2 Auxiliary problem statement. The dual problem

In this Chapter, for the sake of simplicity, we assume that the Dirichlet boundary conditions are homogeneous. We will consider a quantity of interest $Q : V \rightarrow \mathbb{R}$, defined as a linear functional of the displacement field¹. The

¹The explanations are restricted to linear quantities of interest. In the developments, affine quantities of the displacement will also be considered, but we will show that these particular cases can be recast into the linear case.

aim is to estimate the error in functional Q , which is expressed by

$$Q(\mathbf{u}) - Q(\mathbf{u}^h) = Q(\mathbf{u} - \mathbf{u}^h) = Q(\mathbf{e}) \quad (4.1)$$

Standard procedures to evaluate $Q(\mathbf{e})$ introduce the *dual* problem (this terminology comes from the optimal control community) or auxiliary problem. Find $\tilde{\mathbf{u}} \in V$ such that $\forall \mathbf{v} \in V$,

$$\int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D} \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) \, d\Omega = Q(\mathbf{v}). \quad (4.2)$$

Problem (4.2) can be seen as the variational form of an auxiliary mechanical problem. Dual field $\tilde{\mathbf{u}} \in V$ is a displacement that vanishes over Γ_D . Test function \mathbf{v} is a virtual displacement. Field $\tilde{\boldsymbol{\sigma}} = \mathbf{D}(\boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) - \tilde{\boldsymbol{\varepsilon}}_0) + \tilde{\boldsymbol{\sigma}}_0$, where $\tilde{\boldsymbol{\sigma}}_0$ and $\tilde{\boldsymbol{\varepsilon}}_0$ are the initial stress and strain fields, can be interpreted as a mechanical stress field. The left-hand side of (4.2) is the work of the internal forces of the auxiliary mechanical problem. As detailed later on, $Q(\mathbf{v})$ is the work of an abstract external load for the auxiliary mechanical problem.

The dual problem is solved using the cgFEM software. Here, we will make use of the same finite dimensional space used to solve the initial problem (or “primal” problem). Therefore, we will look for an approximation of $\tilde{\mathbf{u}} \in V$ using the Galerkin approach:

Find $\tilde{\mathbf{u}}^h \in V^h$ such that $\forall \mathbf{v} \in V^h$,

$$\int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D} \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}^h) \, d\Omega = Q(\mathbf{v}). \quad (4.3)$$

By substituting $\mathbf{v} = \mathbf{u} - \mathbf{u}^h = \mathbf{e}$ in (4.2) we obtain:

$$Q(\mathbf{e}) = \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{e})^T \mathbf{D} \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) \, d\Omega \quad (4.4)$$

considering the Galerkin orthogonality of the primal problem $a(\mathbf{e}, \tilde{\mathbf{u}}^h) = 0$ we can rewrite expression (4.2) as follows (see [64] for more details):

$$Q(\mathbf{e}) = \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{e})^T \mathbf{D} \boldsymbol{\varepsilon}(\tilde{\mathbf{u}} - \tilde{\mathbf{u}}^h) \, d\Omega = \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{e})^T \mathbf{D} \boldsymbol{\varepsilon}(\tilde{\mathbf{e}}) \, d\Omega \quad (4.5)$$

where $\tilde{\mathbf{e}} = \tilde{\mathbf{u}} - \tilde{\mathbf{u}}^h$ is the discretization error of the dual problem (4.2). Expression (4.5) can now be written in terms of mechanical stresses:

$$Q(\mathbf{e}) = \int_{\Omega} (\boldsymbol{\sigma} - \boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} (\tilde{\boldsymbol{\sigma}} - \tilde{\boldsymbol{\sigma}}^h) \, d\Omega \quad (4.6)$$

where $\tilde{\boldsymbol{\sigma}}^h = \mathbf{D}(\boldsymbol{\varepsilon}(\tilde{\mathbf{u}}^h) - \tilde{\boldsymbol{\varepsilon}}_0) + \tilde{\boldsymbol{\sigma}}_0$ is the finite element stress field for the dual problem. An error estimation for the QoI is obtained following the Zienkiewicz and Zhu ideas [3]:

$$\tilde{\mathcal{E}}_{SPR-CD} = \int_{\Omega} (\boldsymbol{\sigma}_{\sigma}^* - \boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} (\tilde{\boldsymbol{\sigma}}_{\sigma}^* - \tilde{\boldsymbol{\sigma}}^h) \, d\Omega \quad (4.7)$$

where $\boldsymbol{\sigma}_{\sigma}^*$ and $\tilde{\boldsymbol{\sigma}}_{\sigma}^*$ represent the recovered stress field for both, the primal and dual problem and will be evaluated using the SPR-CD technique.

4.3 Analytical definitions of the dual problem for equilibrium enforcement

The SPR-CD procedure relies on the enforcement of equilibrium for the recovered stress fields evaluated at each patch. To apply this technique with the dual problem, the corresponding mechanical equilibrium must be made explicit. In order to do so, the right-hand side of (4.2) is interpreted as the work of the mechanical external forces. The analytical expression of these forces will be derived for some the quantities of interest:

Equation (4.2) can be reformulated as:

Find $\tilde{\mathbf{u}} \in V : \forall \mathbf{v} \in V$:

$$\begin{aligned} \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D} \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) \, d\Omega &= Q(\mathbf{v}) \\ &= \int_{\Omega} \mathbf{v}^T \tilde{\mathbf{b}} \, d\Omega + \int_{\Gamma_N} \mathbf{v}^T \tilde{\mathbf{t}} \, d\Gamma + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D} \tilde{\boldsymbol{\varepsilon}}_0 \, d\Omega - \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \tilde{\boldsymbol{\sigma}}_0 \, d\Omega \end{aligned} \quad (4.8)$$

The problem in (4.8) is solved using a FE approximation with test and trial functions in V^h . The finite element solution is denoted by $\tilde{\mathbf{u}}^h \in V^h$.

Such derivations have been presented in [120, 121, 122]. Here, we will recall some of the results presented in these papers. Additionally, we will provide the analytical expression of the dual load when the quantity of interest is the GSIF in problems with singularities.

4.3.1 Mean displacement in Ω_I

Let us assume that the objective is to evaluate the average of the displacement in a sub-domain of interest $\Omega_I \subset \Omega$. In this case, the quantity of interest is:

$$Q(\mathbf{u}) = \frac{1}{|\Omega_I|} \int_{\Omega_I} \mathbf{u}^T \mathbf{c}_u \, d\Omega \quad (4.9)$$

where $|\Omega_I|$ is the measure of Ω_I and \mathbf{c}_u is a vector used to select the combination of the components of the displacement field that give us the displacement along the defined direction. For example, $\mathbf{c}_u = \{1, 0\}^T$ to extract the first component of \mathbf{u} .

The definition of this QoI (4.9) can be easily identified in (4.8) with $\mathbf{v} = \mathbf{u}$ with the term corresponding to the body forces. Then, the right-hand side of expression (4.8) that defines the dual problem is:

$$Q(\mathbf{v}) = \int_{\Omega_I} \mathbf{v}^T \left(\frac{\mathbf{c}_u}{|\Omega_I|} \right) \, d\Omega \quad (4.10)$$

In this case, the dual problem is equivalent to a problem loaded with the body force $\tilde{\mathbf{b}}$ defined by $\tilde{\mathbf{b}} = \mathbf{c}_u/|\Omega_I|$.

4.3.2 Mean displacement along Γ_I

The quantity of interest is now the mean value of the displacement field along a subset Γ_I of the Neumann boundary Γ_N :

$$Q(\mathbf{u}) = \frac{1}{|\Gamma_I|} \int_{\Gamma_I} \mathbf{u}^T \mathbf{c}_u \, d\Gamma \quad (4.11)$$

where $|\Gamma_I|$ is the length of Γ_I and \mathbf{c}_u an extractor acting on \mathbf{u} .

The QoI defined in (4.11) can also be easily identified with the traction term in (4.8). Then the right hand side of the dual problem is defined by:

$$Q(\mathbf{v}) = \int_{\Gamma_I} \mathbf{v}^T \left(\frac{\mathbf{c}_u}{|\Gamma_I|} \right) \, d\Gamma \quad (4.12)$$

Note that the quantity $\tilde{\mathbf{t}} = \mathbf{c}_u/|\Gamma_I|$ can be interpreted as a traction vector applied along the boundary in the problem defined in (4.8). Thus, $\tilde{\mathbf{t}}$ is a vector of traction applied on Γ_I and that can be used in the dual problem to extract the mean displacements along Γ_I .

4.3.3 Mean strain in Ω_I

In this case we are interested in some combination of the components of the strain over a subdomain Ω_I such that the QoI is given by:

$$Q(\mathbf{u}) = \frac{1}{|\Omega_I|} \int_{\Omega_I} \boldsymbol{\varepsilon}(\mathbf{u})^T \mathbf{c}_\varepsilon \, d\Omega \quad (4.13)$$

where \mathbf{c}_ε is the extraction operator used to define the combination of strains under consideration.

Comparing the expression that defines the QoI (4.13) with the expression (4.8) we identify it with the term corresponding to initial stresses. Then we

can rewrite the right hand side of the dual problem as:

$$Q(\mathbf{v}) = \int_{\Omega_I} \boldsymbol{\varepsilon}(\mathbf{v})^T \frac{\mathbf{c}_\varepsilon}{|\Omega_I|} d\Omega \quad (4.14)$$

where the term $\tilde{\boldsymbol{\sigma}}_0 = \mathbf{c}_\varepsilon/|\Omega_I|$ represents the vector of initial stresses that is used to define the auxiliary problem for this particular QoI.

4.3.4 Mean stress value in Ω_I

Let us consider now $Q(\mathbf{u})$ as the mean stress value given by a combination \mathbf{c}_σ of the stress components $\boldsymbol{\sigma}$ in a domain of interest:

$$Q(\mathbf{u}) = \frac{1}{|\Omega_I|} \int_{\Omega_I} \mathbf{c}_\sigma^T (\mathbf{D}(\boldsymbol{\varepsilon}(\mathbf{u}) - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0) d\Omega \quad (4.15)$$

Q is an affine functional. Let us define

$$\tilde{Q}(\mathbf{v}) = \frac{1}{|\Omega_I|} \int_{\Omega} \mathbf{c}_\sigma^T \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{v}) d\Omega \quad (4.16)$$

for \mathbf{v} an arbitrary vector of $H^1(\Omega)$. \tilde{Q} is such that $\tilde{Q}(\mathbf{e}) = Q(\mathbf{e})$, so that by solving the dual problem

$$\int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D}\boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) d\Omega = \tilde{Q}(\mathbf{v}) \quad (4.17)$$

the derivations of Section 4.2 applies.

As in the previous cases, the right-hand side of the auxiliary problem is defined by the term $\tilde{\boldsymbol{\varepsilon}}_0 = \mathbf{c}_\sigma^T/|\Omega_I|$, which represents in this case a vector of initial strains.

4.3.5 Mean tractions along Γ_I included in Γ_D

Let us assume that we want to evaluate the mean value of a combination of the tractions \mathbf{T}_R on a part Γ_I of the Dirichlet boundary Γ_D .

The application of the principle of virtual work with test functions $\mathbf{v} \in H^1$ that do not necessarily vanish over Γ_D gives:

$$\begin{aligned} \int_{\Gamma_D} \mathbf{v}^T \mathbf{T}_R \, d\Gamma &= \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{u})^T \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{v}) \, d\Omega - \int_{\Gamma_N} \mathbf{v}^T \mathbf{t} \, d\Gamma \\ &\quad - \int_{\Omega} \mathbf{v}^T \mathbf{b} \, d\Omega + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \boldsymbol{\sigma}_0 \, d\Omega - \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D} \boldsymbol{\varepsilon}_0 \, d\Omega \end{aligned} \quad (4.18)$$

Extracting the quantity $\frac{1}{|\Gamma_I|} \int \mathbf{c}_R^T \mathbf{T}_R \, d\Gamma$, where \mathbf{c}_R is an extractor defined over Γ_I , is done by defining the prolongation $\boldsymbol{\delta} \in H^1(\Omega)$ of extractor \mathbf{c}_R such that $\boldsymbol{\delta}|_{\Gamma_I} = \mathbf{c}_R/|\Gamma_I|$. For instance, $\boldsymbol{\delta}$ can be the finite element field that is null at every place that does not belong to Γ_I . Taking $\mathbf{v} = \boldsymbol{\delta}$ for \mathbf{v} in (4.18) yields

$$\begin{aligned} Q(\mathbf{u}) &= \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{u})^T \mathbf{D} \boldsymbol{\varepsilon}(\boldsymbol{\delta}) \, d\Omega - \int_{\Gamma_N} \boldsymbol{\delta}^T \mathbf{t} \, d\Gamma \\ &\quad - \int_{\Omega} \boldsymbol{\delta}^T \mathbf{b} \, d\Omega + \int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{\delta})^T \boldsymbol{\sigma}_0 \, d\Omega - \int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{\delta})^T \mathbf{D} \boldsymbol{\varepsilon}_0 \, d\Omega \end{aligned} \quad (4.19)$$

Q is an affine functional. We define

$$\tilde{Q}(\mathbf{v}) = \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D} \boldsymbol{\varepsilon}(\boldsymbol{\delta}) \, d\Omega \quad (4.20)$$

\tilde{Q} is such that $\tilde{Q}(\mathbf{e}) = Q(\mathbf{e})$, so that by solving the dual problem

$$\int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D} \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) \, d\Omega = \tilde{Q}(\mathbf{v}) \quad (4.21)$$

the derivations of Section 4.2 applies. The dual load is an initial strain $\boldsymbol{\varepsilon}(\boldsymbol{\delta})$. Alternatively, by recalling the variational form in (4.8) we see that the dual problem is a mechanical problem in $\tilde{\mathbf{u}} = \tilde{\mathbf{u}} - \boldsymbol{\delta}$, where $\tilde{\mathbf{u}}|_{\Gamma_I} = -\boldsymbol{\delta}|_{\Gamma_I} = -\mathbf{c}_R/|\Gamma_I|$, hence, we can use a boundary value problem with Dirichlet boundary conditions $\tilde{\mathbf{u}} = -\mathbf{c}_R/|\Gamma_I|$ on Γ_I for this Q oI.

4.3.6 Generalized stress intensity factor

Consider the problem of evaluating the generalized stress intensity factor (GSIF) (3.67), that characterizes the singular solution in problems with reen-

trant corners and cracks, as the quantity of interest, thus $Q(\mathbf{u}) = K$. Function q is used to define the extraction zone Ω_I .

Rearranging the terms of the integral in (3.68) in order to split it in two integrals, we can obtain:

$$K = \int_{\Omega} \mathbf{u}^T \left(\frac{1}{C} \right) \left\{ \begin{array}{l} \sigma_{xx}^{aux} q_{,x} + \sigma_{xy}^{aux} q_{,y} \\ \sigma_{xy}^{aux} q_{,x} + \sigma_{yy}^{aux} q_{,y} \end{array} \right\} d\Omega - \int_{\Omega} \boldsymbol{\sigma}^T \left(-\frac{1}{C} \right) \left\{ \begin{array}{l} u_x^{aux} q_{,x} \\ u_y^{aux} q_{,y} \\ u_y^{aux} q_{,x} + u_x^{aux} q_{,y} \end{array} \right\} d\Omega \quad (4.22)$$

Rewriting the previous expression, we obtain

$$K = \int_{\Omega} \mathbf{u}^T \tilde{\mathbf{b}} d\Omega + \int_{\Omega} \boldsymbol{\sigma}^T \tilde{\boldsymbol{\varepsilon}}_0 d\Omega \quad (4.23)$$

where $\tilde{\mathbf{b}}$ and the $\tilde{\boldsymbol{\varepsilon}}_0$ are equivalent to body forces and initial strains, respectively.

Comparing (4.23) with the weak form of the dual problem (4.8) we can easily cast the right hand side of the dual problem as:

$$Q(\mathbf{v}) = \int_{\Omega} \mathbf{v}^T \tilde{\mathbf{b}} d\Omega + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^T \mathbf{D} \tilde{\boldsymbol{\varepsilon}}_0 d\Omega \quad (4.24)$$

Hence, the body loads in the dual problem are:

$$\tilde{\mathbf{b}} = \left(\frac{1}{C} \right) \left\{ \begin{array}{l} \sigma_{xx}^{aux} q_{,x} + \sigma_{xy}^{aux} q_{,y} \\ \sigma_{xy}^{aux} q_{,x} + \sigma_{yy}^{aux} q_{,y} \end{array} \right\} \quad (4.25)$$

and the initial strains in the dual problem are:

$$\tilde{\boldsymbol{\varepsilon}}_0 = \left(-\frac{1}{C} \right) \left\{ \begin{array}{l} u_x^{aux} q_{,x} \\ u_y^{aux} q_{,y} \\ u_y^{aux} q_{,x} + u_x^{aux} q_{,y} \end{array} \right\} \quad (4.26)$$

It must be taken into account that these expressions can be used either for mode I or mode II.

4.4 Error bounds in QoI

A methodology to get upper and lower bounds in QoI was introduced in [119]. The ideas are summarized in the following expressions. Let us apply the parallelogram identity to the following quantities:

$$\begin{aligned} \|\mathbf{e} + \tilde{\mathbf{e}}\|_{\Omega}^2 &= a(\mathbf{e} + \tilde{\mathbf{e}}, \mathbf{e} + \tilde{\mathbf{e}}) = a(\mathbf{e}, \mathbf{e}) + a(\tilde{\mathbf{e}}, \tilde{\mathbf{e}}) + 2a(\mathbf{e}, \tilde{\mathbf{e}}) \\ \|\mathbf{e} - \tilde{\mathbf{e}}\|_{\Omega}^2 &= a(\mathbf{e} + \tilde{\mathbf{e}}, \mathbf{e} + \tilde{\mathbf{e}}) = a(\mathbf{e}, \mathbf{e}) + a(\tilde{\mathbf{e}}, \tilde{\mathbf{e}}) - 2a(\mathbf{e}, \tilde{\mathbf{e}}) \end{aligned} \quad (4.27)$$

subtracting the second expression to the first one we can define the error in the QoI as:

$$Q(\mathbf{e}) = \frac{1}{4} \left\| \left\| \tau \mathbf{e} + \frac{\tilde{\mathbf{e}}}{\tau} \right\|_{\Omega} \right\|^2 - \frac{1}{4} \left\| \left\| \tau \mathbf{e} - \frac{\tilde{\mathbf{e}}}{\tau} \right\|_{\Omega} \right\|^2 \quad (4.28)$$

where $\tau = \sqrt{\|\tilde{\mathbf{e}}\|_{\Omega} / \|\mathbf{e}\|_{\Omega}}$ is a scaling factor. From this expression we can evaluate upper and lower bounds of $Q(\mathbf{e})$ by only bounding the energy norms in (4.28). Let us consider that η_{low}^+ , η_{upp}^+ , η_{low}^- and η_{upp}^- represent global error estimates such that

$$\eta_{low}^+ \leq \left\| \left\| \tau \mathbf{e} + \frac{\tilde{\mathbf{e}}}{\tau} \right\|_{\Omega} \right\| \leq \eta_{upp}^+ \quad \eta_{low}^- \leq \left\| \left\| \tau \mathbf{e} - \frac{\tilde{\mathbf{e}}}{\tau} \right\|_{\Omega} \right\| \leq \eta_{upp}^- \quad (4.29)$$

Then, using (4.28) we have

$$\tilde{\mathcal{E}}_{LB} = \frac{1}{4}(\eta_{low}^+)^2 - \frac{1}{4}(\eta_{upp}^-)^2 \leq Q(\mathbf{e}) \leq \frac{1}{4}(\eta_{upp}^+)^2 - \frac{1}{4}(\eta_{low}^-)^2 = \tilde{\mathcal{E}}_{UB} \quad (4.30)$$

The following step is to obtain appropriate values for η_{upp}^- , η_{upp}^+ , η_{low}^- , η_{low}^+ . These quantities are upper and lower bounds of the energy norms in (4.29), thus any bounding technique for energy norm presented in Section 3.4 can be used. Now, we are going to adapt those bounding techniques for this particular situation.

4.4.1 Definitions

First, let us define the following quantity $\mathbf{E}^{\pm} = \tau \mathbf{e} \pm \frac{\tilde{\mathbf{e}}}{\tau}$, where \pm symbol indicates the adding and subtracting forms. By evaluating the energy norms

$|||\mathbf{E}^\pm|||_\Omega$ we could evaluate the error in the QoI since it represents the terms in (4.28). Then, to evaluate the bounds of the QoI we should obtain the upper and lower bounds of $|||\mathbf{E}^\pm|||_\Omega$.

Let us taking into account the residual for the primal problem (3.6), we define the residual for the dual problem:

$$\tilde{R}(\mathbf{v}) = a(\tilde{\mathbf{e}}, \mathbf{v}) \quad \forall \mathbf{v} \in V \quad (4.31)$$

Now, recalling the expression (3.89) presented in [73], considering the lack of equilibrium along boundaries, and writing it for the dual problem, we will have:

$$\bar{a}(\tilde{\boldsymbol{\sigma}}_\sigma^*, \boldsymbol{\sigma}(\mathbf{v})) = Q(\mathbf{v}) + \int_\Omega (\tilde{\mathbf{s}}_\sigma^*)^T \mathbf{v} \, d\Omega + \int_\Gamma (\tilde{\mathbf{r}}_\sigma^*)^T \mathbf{v} \, d\Gamma \quad \forall \mathbf{v} \in V \quad (4.32)$$

This allow us to write the residual expression, as in (3.91), for the dual problem in the following way:

$$\tilde{R}(\mathbf{v}) = \bar{a}(\tilde{\boldsymbol{\sigma}}_\sigma^* - \tilde{\boldsymbol{\sigma}}^h, \boldsymbol{\sigma}(\mathbf{v})) - \int_\Omega (\tilde{\mathbf{s}}_\sigma^*)^T \mathbf{v} \, d\Omega - \int_\Gamma (\tilde{\mathbf{r}}_\sigma^*)^T \mathbf{v} \, d\Gamma \quad \forall \mathbf{v} \in V \quad (4.33)$$

We define the residual equation for \mathbf{E}^\pm as a linear combination of the residuals of the primal and dual problems:

$$R^\pm(\mathbf{v}) := \tau R(\mathbf{v}) \pm \frac{1}{\tau} \tilde{R}(\mathbf{v}) = a(\mathbf{z}^\pm, \mathbf{v}) \quad \forall \mathbf{v} \in V \quad (4.34)$$

Finally, considering (3.91) and (4.33) into (4.34) we obtain the residual equation for \mathbf{E}^\pm with the lacks of equilibrium for the primal and dual recovered solution.

$$\begin{aligned} R^\pm(\mathbf{v}) &= a(\tau \mathbf{e}, \mathbf{v}) \pm a\left(\frac{\tilde{\mathbf{e}}}{\tau}, \mathbf{v}\right) = a(\mathbf{E}^\pm, \mathbf{v}) \\ &= a(\mathbf{E}_\sigma^{*\pm}, \mathbf{v}) - \tau \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{v} \, d\Omega \mp \frac{1}{\tau} \int_\Omega (\tilde{\mathbf{s}}_\sigma^*)^T \mathbf{v} \, d\Omega \\ &\quad - \tau \int_\Gamma (\mathbf{r}_\sigma^*)^T \mathbf{v} \, d\Gamma \mp \frac{1}{\tau} \int_\Gamma (\tilde{\mathbf{r}}_\sigma^*)^T \mathbf{v} \, d\Gamma \end{aligned} \quad (4.35)$$

for all $\mathbf{v} \in V$. In this case $\mathbf{E}_\sigma^{*\pm}$ indicates the estimated error using the recovered fields.

4.4.2 The upper bound for GOA

Theorem 4.1. *Under the definitions given in Subsection 4.4.1:*

$$\begin{aligned}
 a(\mathbf{E}^\pm, \mathbf{E}^\pm) &\leq a(\mathbf{E}_\sigma^{*\pm}, \mathbf{E}_\sigma^{*\pm}) - 2\tau \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{E}^\pm \, d\Omega \mp \frac{2}{\tau} \int_\Omega (\tilde{\mathbf{s}}_\sigma^*)^T \mathbf{E}^\pm \, d\Omega \\
 &\quad - 2\tau \int_\Gamma (\mathbf{r}_\sigma^*)^T \mathbf{E}^\pm \, d\Gamma \mp \frac{2}{\tau} \int_\Gamma (\tilde{\mathbf{r}}_\sigma^*)^T \mathbf{E}^\pm \, d\Gamma
 \end{aligned} \tag{4.36}$$

is an upper bound of $a(\mathbf{E}^\pm, \mathbf{E}^\pm)$.

Proof. Taking the positive quantity $0 \leq a(\mathbf{E}^\pm - \mathbf{E}_\sigma^{*\pm}, \mathbf{E}^\pm - \mathbf{E}_\sigma^{*\pm})$, and expanding it we obtain:

$$0 \leq a(\mathbf{E}^\pm, \mathbf{E}^\pm) + a(\mathbf{E}_\sigma^{*\pm}, \mathbf{E}_\sigma^{*\pm}) - 2a(\mathbf{E}^\pm, \mathbf{E}_\sigma^{*\pm}) \tag{4.37}$$

finally combining (4.35), with $\mathbf{v} = \mathbf{E}^\pm$, with the last term in (4.37) we obtain:

$$\begin{aligned}
 0 &\leq a(\mathbf{E}^\pm, \mathbf{E}^\pm) + a(\mathbf{E}_\sigma^{*\pm}, \mathbf{E}_\sigma^{*\pm}) - 2a(\mathbf{E}^\pm, \mathbf{E}^\pm) - 2\tau \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{E}^\pm \, d\Omega \\
 &\quad \mp \frac{2}{\tau} \int_\Omega (\tilde{\mathbf{s}}_\sigma^*)^T \mathbf{E}^\pm \, d\Omega - 2\tau \int_\Gamma (\mathbf{r}_\sigma^*)^T \mathbf{E}^\pm \, d\Gamma \mp \frac{2}{\tau} \int_\Gamma (\tilde{\mathbf{r}}_\sigma^*)^T \mathbf{E}^\pm \, d\Gamma
 \end{aligned} \tag{4.38}$$

proving (4.36). Consider then that the upper bounds for both terms in (4.29) are obtained. \square

Note that (4.36) is condensed for \mathbf{E}^\pm and the part for the primal and dual problem are not explicitly described. The following expression decompose (4.36) into the primal and dual contributions:

$$\begin{aligned}
 a(\mathbf{E}^\pm, \mathbf{E}^\pm) &\leq \tau^2 \left\{ \|\|\mathbf{e}_\sigma^*\|\|_\Omega^2 - 2 \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{e} \, d\Omega - 2 \int_\Gamma (\mathbf{r}_\sigma^*)^T \mathbf{e} \, d\Gamma \right\} \\
 &\quad + \frac{1}{\tau^2} \left\{ \|\|\tilde{\mathbf{e}}_\sigma^*\|\|_\omega^2 - 2 \int_\Omega (\tilde{\mathbf{s}}_\sigma^*)^T \tilde{\mathbf{e}} \, d\Omega - 2 \int_\Gamma (\tilde{\mathbf{r}}_\sigma^*)^T \tilde{\mathbf{e}} \, d\Gamma \right\} \\
 &\quad \pm 2a(\mathbf{e}_\sigma^*, \tilde{\mathbf{e}}_\sigma^*) \mp 2 \int_\Omega (\mathbf{s}_\sigma^*)^T \tilde{\mathbf{e}} \, d\Omega \mp 2 \int_\Gamma (\mathbf{r}_\sigma^*)^T \tilde{\mathbf{e}} \, d\Gamma \\
 &\quad \mp 2 \int_\Omega (\tilde{\mathbf{s}}_\sigma^*)^T \mathbf{e} \, d\Omega \mp 2 \int_\Gamma (\tilde{\mathbf{r}}_\sigma^*)^T \mathbf{e} \, d\Gamma
 \end{aligned} \tag{4.39}$$

4.4.3 The lower bound for GOA

Theorem 4.2. *Under the definitions given in Subsection 4.4.1:*

$$\begin{aligned}
a(\mathbf{E}^\pm, \mathbf{E}^\pm) &\geq -\lambda^2 a(\mathbf{E}_u^{*\pm}, \mathbf{E}_u^{*\pm}) + 2\lambda a(\mathbf{E}_\sigma^{*\pm}, \mathbf{E}_u^{*\pm}) - 2\lambda\tau \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{E}_u^{*\pm} \, d\Omega \\
&\mp 2\lambda \frac{1}{\tau} \int_\Omega (\tilde{\mathbf{s}}_\sigma^*)^T \mathbf{E}_u^{*\pm} \, d\Omega - 2\lambda\tau \int_\Gamma (\mathbf{r}_\sigma^*)^T \mathbf{E}_u^{*\pm} \, d\Gamma \mp 2\lambda \frac{1}{\tau} \int_\Gamma (\tilde{\mathbf{r}}_\sigma^*)^T \mathbf{E}_u^{*\pm} \, d\Gamma
\end{aligned} \tag{4.40}$$

is a lower bound of $a(\mathbf{E}^\pm, \mathbf{E}^\pm)$ for any $\lambda \in \mathbb{R}$.

Proof. Taking the quantity positive $0 \leq a(\mathbf{E}^\pm - \lambda \mathbf{E}_u^{*\pm}, \mathbf{E}^\pm - \lambda \mathbf{E}_u^{*\pm})$, and expanding it we obtain:

$$0 \leq a(\mathbf{E}^\pm, \mathbf{E}^\pm) + \lambda^2 a(\mathbf{E}_u^{*\pm}, \mathbf{E}_u^{*\pm}) - 2\lambda a(\mathbf{E}^\pm, \mathbf{E}_u^{*\pm}) \tag{4.41}$$

$\mathbf{E}_u^{*\pm}$ corresponds to the kinematic admissible recovered fields for both, primal and dual problems (see Section 3.3). Combining the last term in the previous expressions with (4.35) when $\mathbf{v} = \mathbf{E}_u^{*\pm}$ we obtain:

$$0 \leq a(\mathbf{E}^\pm - \lambda \mathbf{E}_u^{*\pm}, \mathbf{E}^\pm - \lambda \mathbf{E}_u^{*\pm}) = a(\mathbf{E}^\pm, \mathbf{E}^\pm) + \lambda^2 a(\mathbf{E}_u^{*\pm}, \mathbf{E}_u^{*\pm}) - 2\lambda a(\mathbf{E}^\pm, \mathbf{E}_u^{*\pm}) \tag{4.42}$$

from which we can directly proof (4.40). \square

Expression (4.40) is valid for any $\lambda \in \mathbb{R}$. Solving the corresponding optimization problem for λ and rearranging the terms:

$$\begin{aligned}
a(\mathbf{E}^\pm, \mathbf{E}^\pm) &\geq \frac{1}{a(\mathbf{E}_u^{*\pm}, \mathbf{E}_u^{*\pm})} \left\{ a(\mathbf{E}_\sigma^{*\pm}, \mathbf{E}_u^{*\pm}) - \tau \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{E}_u^{*\pm} \, d\Omega \right. \\
&\quad \left. \mp \frac{1}{\tau} \int_\Omega (\tilde{\mathbf{s}}_\sigma^*)^T \mathbf{E}_u^{*\pm} \, d\Omega - \tau \int_\Gamma (\mathbf{r}_\sigma^*)^T \mathbf{E}_u^{*\pm} \, d\Gamma \mp \frac{1}{\tau} \int_\Gamma (\tilde{\mathbf{r}}_\sigma^*)^T \mathbf{E}_u^{*\pm} \, d\Gamma \right\}^2
\end{aligned} \tag{4.43}$$

The next step consist in making explicit the contributions of the primal and dual problems:

$$\begin{aligned}
a(\mathbf{E}^\pm, \mathbf{E}^\pm) &\geq \frac{1}{\|\|\mathbf{e}_u^*\|\|_\Omega^2 + \|\|\tilde{\mathbf{e}}_u^*\|\|_\Omega^2 \pm 2a(\mathbf{e}_u^*, \tilde{\mathbf{e}}_u^*)} \left\{ \tau^2 a(\mathbf{e}_\sigma^*, \mathbf{e}_u^*) + \frac{1}{\tau^2} a(\boldsymbol{\epsilon}_\sigma^*, \boldsymbol{\epsilon}_u^*) \right. \\
&\quad \pm a(\mathbf{e}_\sigma^*, \tilde{\mathbf{e}}_u^*) \pm a(\mathbf{e}_u^*, \tilde{\mathbf{e}}_\sigma^*) - \tau^2 \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{e}_u^* \, d\Omega - \tau^2 \int_\Gamma (\mathbf{r}_\sigma^*)^T \mathbf{e}_u^* \, d\Gamma \\
&\quad \mp \int_\Omega (\tilde{\mathbf{s}}_\sigma^*)^T \mathbf{e}_u^* \, d\Omega \mp \int_\Gamma (\tilde{\mathbf{r}}_\sigma^*)^T \mathbf{e}_u^* \, d\Gamma - \frac{1}{\tau^2} \int_\Omega (\tilde{\mathbf{s}}_\sigma^*)^T \tilde{\mathbf{e}}_u^* \, d\Omega \\
&\quad \left. - \frac{1}{\tau^2} \int_\Gamma (\tilde{\mathbf{r}}_\sigma^*)^T \tilde{\mathbf{e}}_u^* \, d\Gamma \mp \int_\Omega (\mathbf{s}_\sigma^*)^T \tilde{\mathbf{e}}_u^* \, d\Omega \mp \int_\Gamma (\mathbf{r}_\sigma^*)^T \tilde{\mathbf{e}}_u^* \, d\Gamma \right\}^2
\end{aligned} \tag{4.44}$$

4.4.4 Numerical bounding approaches for GOA

In (4.44) all values are computable, however, in (4.39) \mathbf{e} and $\tilde{\mathbf{e}}$ are not computable since its evaluation would require the analytical solution for both, the primal and the dual problem. To overcome this problem we can substitute \mathbf{e} by \mathbf{e}_u^* and $\tilde{\mathbf{e}}$ by $\tilde{\mathbf{e}}_u^*$, both computable. This will lead to a computable version of the upper bound of $\|\|\mathbf{E}^\pm\|\|_\Omega$. However, the evaluation of (4.39) and (4.44) using \mathbf{e}_u^* and $\tilde{\mathbf{e}}_u^*$ is extremely laborious and we do not recommend it.

Instead, we propose the use of a simplified computable version of the upper and lower bounds² of the error in the QoI. Our purpose is to develop a computationally efficient technique to evaluate computable (and thus non-guaranteed) bounds of the error in QoI's. Taking into account that $\|\|\mathbf{e}_\sigma\|\|_\Omega^2 \geq 2a(\mathbf{e}_\sigma, \mathbf{e}_u) - \|\|\mathbf{e}_u\|\|_\Omega^2$ holds:

$$\begin{aligned}
0 &\leq a(\mathbf{e}_\sigma - \mathbf{e}_u, \mathbf{e}_\sigma - \mathbf{e}_u) = a(\mathbf{e}_\sigma, \mathbf{e}_\sigma) + a(\mathbf{e}_u, \mathbf{e}_u) - 2a(\mathbf{e}_\sigma, \mathbf{e}_u) \\
&\text{then, } \|\|\mathbf{e}_\sigma\|\|_\Omega^2 \geq 2a(\mathbf{e}_\sigma, \mathbf{e}_u) - \|\|\mathbf{e}_u\|\|_\Omega^2
\end{aligned} \tag{4.45}$$

²Note that when we refer to a computable bound we are assuming some approximations and therefore, the bounding properties are not strictly guaranteed. Whenever the guaranteed bounds were obtained we will specifically indicate it as *guaranteed bounds*.

and that expression (4.28) could be rewritten as:

$$4Q(\mathbf{e}) = \tau^2 \|\mathbf{e}\|_\Omega^2 + \frac{1}{\tau^2} \|\tilde{\mathbf{e}}\|_\Omega^2 + 2a(\mathbf{e}, \tilde{\mathbf{e}}) - \tau^2 \|\mathbf{e}\|_\Omega^2 - \frac{1}{\tau^2} \|\tilde{\mathbf{e}}\|_\Omega^2 + 2a(\mathbf{e}, \tilde{\mathbf{e}}) \quad (4.46)$$

Let us assume that we can evaluate upper and lower bounds of the error in energy norm for the primal and dual problems, $\mathcal{E}_{\text{SPR-CD}_{LB_1}} \leq \|\mathbf{e}\|_\Omega \leq \mathcal{E}_{\text{SPR-CD}_{UB}}$ and $\tilde{\mathcal{E}}_{\text{SPR-CD}_{LB_1}} \leq \|\tilde{\mathbf{e}}\|_\Omega \leq \tilde{\mathcal{E}}_{\text{SPR-CD}_{UB}}$. These values can be used to bound the first two addends of (4.47). The lower bound in energy norm, for both primal and dual problems, could be obtained with expression (3.96) with $\lambda = 1$. However, since the upper bound of the error in energy norm can not be guaranteed, the bounds in the QoI are also non-guaranteed bounds if the bounding procedure proposed in Section 3.4 is used. Expression (4.46) could be rewritten as follows:

$$4Q(\mathbf{e}) = \tau^2 \left\{ \|\mathbf{e}\|_\Omega^2 - \|\mathbf{e}\|_\Omega^2 \right\} + \frac{1}{\tau^2} \left\{ \|\tilde{\mathbf{e}}\|_\Omega^2 - \|\tilde{\mathbf{e}}\|_\Omega^2 \right\} + 4a(\mathbf{e}, \tilde{\mathbf{e}}) \quad (4.47)$$

Substituting the exact errors in energy norm by the corresponding numerical upper bound and the lower bound, and considering (4.45), we obtain the upper error bound in the QoI:

$$\begin{aligned} \tilde{\mathcal{E}}_{\text{SPR-CD}_{UB}} &:= \frac{\tau^2}{4} \underbrace{\left\{ \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 - \mathcal{E}_{\text{SPR-CD}_{LB_1}}^2 \right\}}_{\geq 0} \\ &+ \frac{1}{4\tau^2} \underbrace{\left\{ \hat{\tilde{\mathcal{E}}}_{\text{SPR-CD}_{UB}}^2 - \tilde{\mathcal{E}}_{\text{SPR-CD}_{LB_1}}^2 \right\}}_{\geq 0} + a(\mathbf{e}, \tilde{\mathbf{e}}) \leq Q(\mathbf{e}) \end{aligned} \quad (4.48)$$

and the corresponding lower bound:

$$\begin{aligned} \tilde{\mathcal{E}}_{\text{SPR-CD}_{LB}} &:= \frac{\tau^2}{4} \underbrace{\left\{ \mathcal{E}_{\text{SPR-CD}_{LB_1}}^2 - \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 \right\}}_{\leq 0} \\ &+ \frac{1}{4\tau^2} \underbrace{\left\{ \tilde{\mathcal{E}}_{\text{SPR-CD}_{LB_1}}^2 - \hat{\tilde{\mathcal{E}}}_{\text{SPR-CD}_{UB}}^2 \right\}}_{\leq 0} + a(\mathbf{e}, \tilde{\mathbf{e}}) \geq Q(\mathbf{e}) \end{aligned} \quad (4.49)$$

Notice that even if the upper error bound in energy norm for both, primal and dual problems is the numerical approximation, because of (4.45) the positiveness or the negativeness of the quantities between braces are guaranteed when $\lambda = 1$. Then the bounding properties of (4.48) and (4.49) hold. In spite of that we are not able to evaluate them since we need the exact solution to obtain the remaining term $a(\mathbf{e}, \bar{\mathbf{e}})$. Thus the numerical approach for the bounding techniques is based on the evaluation of an estimation for that term, which will be the plain error estimation in the QoI (4.7). Thus the bounding formulas are nothing but the error estimation plus some correction terms. The numerical version of the upper bound are:

$$\begin{aligned} \hat{\mathcal{E}}_{\text{SPR-CD}_{UB_1}} &:= \frac{\tau^2}{4} \underbrace{\left\{ \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 - \mathcal{E}_{\text{SPR-CD}_{LB_1}}^2 \right\}}_{\geq 0} \\ &+ \frac{1}{4\tau^2} \underbrace{\left\{ \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 - \tilde{\mathcal{E}}_{\text{SPR-CD}_{LB_1}}^2 \right\}}_{\geq 0} + \tilde{\mathcal{E}}_{\text{SPR-CD}} \gtrsim Q(\mathbf{e}) \end{aligned} \quad (4.50)$$

and for the lower bound we have

$$\begin{aligned} \hat{\mathcal{E}}_{\text{SPR-CD}_{LB_1}} &:= \frac{\tau^2}{4} \underbrace{\left\{ \mathcal{E}_{\text{SPR-CD}_{LB_1}}^2 - \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 \right\}}_{\leq 0} \\ &+ \frac{1}{4\tau^2} \underbrace{\left\{ \tilde{\mathcal{E}}_{\text{SPR-CD}_{LB_1}}^2 - \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 \right\}}_{\leq 0} + \tilde{\mathcal{E}}_{\text{SPR-CD}} \lesssim Q(\mathbf{e}) \end{aligned} \quad (4.51)$$

Finally, another alternative is to use the optimal value of λ for the lower bounds. In this cases ((4.52) and (4.53)) is not guaranteed that the positiveness or the negativeness of the terms in braces holds, however numerical examples will show a better accuracy but losing the bounding property in some situations. The respective expressions are:

$$\begin{aligned} \hat{\mathcal{E}}_{\text{SPR-CD}_{UB_{Opt}}} &:= \frac{\tau^2}{4} \left\{ \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 - \mathcal{E}_{\text{SPR-CD}_{LB_{Opt}}}^2 \right\} \\ &+ \frac{1}{4\tau^2} \left\{ \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 - \tilde{\mathcal{E}}_{\text{SPR-CD}_{LB_{Opt}}}^2 \right\} + \tilde{\mathcal{E}}_{\text{SPR-CD}} \gtrsim Q(\mathbf{e}) \end{aligned} \quad (4.52)$$

$$\begin{aligned} \hat{\mathcal{E}}_{\text{SPR-CD}_{LB_{Opt}}} &:= \frac{\tau^2}{4} \left\{ \mathcal{E}_{\text{SPR-CD}_{LB_{Opt}}}^2 - \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 \right\} \\ &+ \frac{1}{4\tau^2} \left\{ \tilde{\mathcal{E}}_{\text{SPR-CD}_{LB_{Opt}}}^2 - \hat{\mathcal{E}}_{\text{SPR-CD}_{UB}}^2 \right\} + \tilde{\mathcal{E}}_{\text{SPR-CD}} \lesssim Q(\mathbf{e}) \end{aligned} \quad (4.53)$$

The results will show that these computational bounds will provide numerical results with bounding properties. This fact is due to the high accuracy of the error estimator in the QoI, $\tilde{\mathcal{E}}_{\text{SPR-CD}}$ with respect of the difference of the terms in braces.

4.5 Numerical results

In this Section 2D benchmark problems with exact solutions are used to investigate the quality of the proposed technique. All problems have been solved with bilinear elements (Q4) and h -adaptive refinement. The first problem has a smooth solution whilst the second is a singular problem. These two problems correspond to the geometries of problems 2 and 3 shown in Section 3.5. For all models we assume plane strain conditions. To assess the performance of the proposed technique we consider the effectivity index of the error estimator $\tilde{\vartheta}$ defined as the quotient of the estimated error $\tilde{\mathcal{E}}$ in the QoI over the exact error $Q(\mathbf{e})$ as follows:

$$\tilde{\vartheta} = \frac{\tilde{\mathcal{E}} - Q(\mathbf{e})}{|Q(\mathbf{e})|} \quad (4.54)$$

Note that in this definition of global effectivity index for the error estimator in QoI, positive values will indicate over estimation of the true error, while negative values will indicate under estimation of the true error. We can also represent the effectivity in the QoI, $\tilde{\theta}_{QoI}$, defined as the value of the QoI, $Q(\mathbf{u}^h)$ evaluated from the FE analysis, corrected using the error estimate, divided by the exact value $Q(\mathbf{u})$:

$$\tilde{\theta}_{QoI} = \frac{Q(\mathbf{u}^h) + \tilde{\mathcal{E}}}{Q(\mathbf{u})} \quad (4.55)$$

The distribution of the local effectivity index \tilde{D} is analyzed at the element level, following the definitions described in [71] for the error in the energy norm, adapted here to the case of the error in QoI:

$$\begin{aligned} \tilde{D} &= \tilde{\vartheta}_K - 1 & \text{if } \tilde{\theta}_K \geq 1 \\ \tilde{D} &= 1 - \frac{1}{\tilde{\vartheta}_K} & \text{if } \tilde{\theta}_K < 1 \end{aligned} \quad \text{with} \quad \tilde{\vartheta}_K = \frac{\tilde{\mathcal{E}}_K}{Q(\mathbf{e}_K)} \quad (4.56)$$

Nonetheless, we should remark that this is only possible for some problems with analytical solutions as the exact value of the solution is unknown in the vast majority of cases, especially for the dual problem.

Once the error in the QoI is estimated, the local error estimates in each element can be used to perform h -adaptive analysis using techniques similar to those already available for the error in the energy norm. The refinement of the mesh using the error estimate as the guiding parameter considers a stopping criterion that checks the value of the global estimated error against a prescribed or desired error in the QoI. If the estimated error is higher than the desired error the mesh is refined. The technique to evaluate the new size distribution for the new mesh is the same than the one used in Chapter 3. However, we apply some minor modifications in the GOA context because the local contributions to the global error in the QoI can take negative values. Thus, for our implementation using h -adaptive routines developed for the error in the energy norm we prepare as input the square root of the absolute values of the error in the QoI at each element and the ratio of the estimated error in the QoI in the current mesh to the desired error in the new mesh.

4.5.1 Problem 2: Thick-wall cylinder subjected to internal pressure

The definition of this problem is detailed in Section 3.5.1. At this point we are interested in presenting the dual problems to be solved in this Section under the GOA framework. In Figure 4.1 we present the model for the dual problem in which we can perceive the domains of interest (DoI) Ω_I and Γ_o .

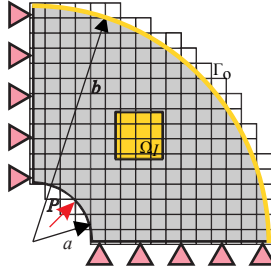


Figure 4.1: Problem 2. Thick-wall cylinder subjected to an internal pressure. The domains of interest Ω_I and Γ_o are indicated in yellow.

Several linear QoI were considered for this problem: the mean radial displacements along Γ_o , the mean displacements \bar{u}_x in the DoI Ω_I and the mean stresses $\bar{\sigma}_x$ in Ω_I .

Problem 2.a: Mean radial displacements along Γ_o

Let Q be the functional that evaluates the mean normal displacement \bar{u}_n along Γ_o such that:

$$Q(\mathbf{u}) = \bar{u}_n = \frac{1}{|\Gamma_o|} \int_{\Gamma_o} (\mathbf{R}\mathbf{u})^T \mathbf{c}_u d\Gamma \tag{4.57}$$

where \mathbf{R} is the standard rotation matrix for the displacements that aligns the coordinate system with the boundary Γ_o and $\mathbf{c}_u = \{1, 0\}^T$ is the extraction vector that selects the normal component. The exact value of the QoI given by (3.99) for $r = b$ is $\bar{u}_n = 2.426 \cdot 10^{-3}$.

In Figure 4.2 we represent the set of the first four meshes of the h -adaptive refinement process. We observe in this case that the refinement process is similar to the one obtained for the energy norm. This peculiarity happens because the dual problem considered for the QoI is a traction field over the boundary Γ_o .

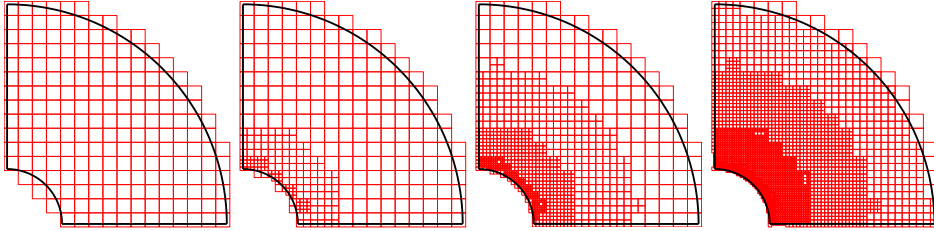


Figure 4.2: Problem 2.a. \bar{u}_n along Γ_o . Q4h. Sequence of h -adaptive refined meshes.

Table 4.1 presents the convergence analysis results under the h -adaptive refinement process for this problem. This Table shows the exact global error in the QoI, $Q(\mathbf{e})$, evaluated using (4.1), the estimated error in the QoI, $\tilde{\mathcal{E}}_{\text{SPR-CD}}$, evaluated with (4.7), the effectivity index for the error estimate, $\tilde{\vartheta}_{\text{SPR-CD}}$, and the effectivity index for the corrected value of the QoI, $\tilde{\theta}_{\text{SPR-CD}}^{\text{QoI}}$, all of them evaluated using the SPR-CD technique. We could observe that both, the estimated error in the QoI and the exact one are very similar yielding a highly accurate error estimation, which is reflected in the effectivity indexes.

Table 4.1: Problem 2.a. \bar{u}_n along Γ_o . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimator $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{\text{QoI}}$ for the GOA.

DoF	$Q(\mathbf{e})$	$\tilde{\mathcal{E}}_{\text{SPR-CD}}$	$\tilde{\vartheta}_{\text{SPR-CD}}$	$\tilde{\theta}_{\text{SPR-CD}}^{\text{QoI}}$
452	0.000020094	0.000024067	0.197752638	1.001637450
800	0.000005751	0.000005714	-0.006289505	0.999985096
2,362	0.000001548	0.000001519	-0.018576279	0.999988153
8,570	0.000000397	0.000000395	-0.007208172	0.999998820
31,888	0.000000102	0.000000101	-0.011756630	0.999999505
121,260	0.000000026	0.000000026	-0.009305513	0.999999900

Table 4.2 shows the performance of the bounding technique presented in Section 4.4 for the error estimation in QoI. This represent the values of $\tilde{\vartheta}$ the global effectivities of the error bounds while $\tilde{\theta}^{\text{QoI}}$ refer to the effectivities of the QoI bounds. Equation (4.50) is used for the upper error bound evaluation whilst equation (4.51) is used for the lower bound. As indicated before, we expect positive values for $\tilde{\vartheta}_{\text{SPR-CD}_{\text{UB}_1}}$ and negative values for $\tilde{\vartheta}_{\text{SPR-CD}_{\text{LB}_1}}$. The values in Table 4.2 shows the high accuracy of the evaluations. However, al-

though $\tilde{\mathcal{E}}_{\text{SPR-CD}_{\text{LB}_1}}$ always produces underestimations of the true error. Note that these are computational versions of the guaranteed bound, then this situation could happen above all when we are having a very high accuracy in the bounds evaluation. In the following examples, with a more complex solution, we will obtain the desired bounding properties.

Table 4.2: Problem 2.a. \bar{u}_n along Γ_o . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{\text{QoI}}$ of the bounding techniques for the GOA.

DoF	$\tilde{\vartheta}_{\text{SPR-CD}_{\text{UB}_1}}$	$\tilde{\vartheta}_{\text{SPR-CD}_{\text{LB}_1}}$	$\tilde{\theta}_{\text{SPR-CD}_{\text{UB}_1}}^{\text{QoI}}$	$\tilde{\theta}_{\text{SPR-CD}_{\text{LB}_1}}^{\text{QoI}}$
452	0.409034150	-0.002689044	1.003386924	0.999977734
800	0.028501282	-0.006034606	1.000067540	0.999985700
2,362	-0.003518890	-0.016444636	0.999997756	0.999989513
8,570	-0.000110991	-0.006310464	0.999999982	0.999998967
31,888	-0.007879272	-0.011211376	0.999999668	0.999999528
121,260	-0.006953840	-0.009182392	0.999999925	0.999999902

Finally, as a summary, these results have been plotted in Figure 4.3. The Figure shows that the global effectivity index for the error estimation on the QoI is stable during the refinement process, and that the value of the QoI is accurately captured from the very beginning of the analysis.

Problem 2.b: Mean displacements \bar{u}_x in Ω_I

Let us consider the mean displacement \bar{u}_x in Ω_I as the quantity of interest. The objective is to evaluate the error when evaluating \bar{u}_x defined by the functional:

$$Q(\mathbf{u}) = \bar{u}_x = \frac{1}{|\Omega_I|} \int_{\Omega_I} u_x d\Omega \quad (4.58)$$

The exact value of the QoI for this problem with analytical solution is $\bar{u}_x = 0.002238239291713$. Figure 4.4 shows the first four meshes used in the h -adaptive refinement process guided by the error estimate in the QoI.

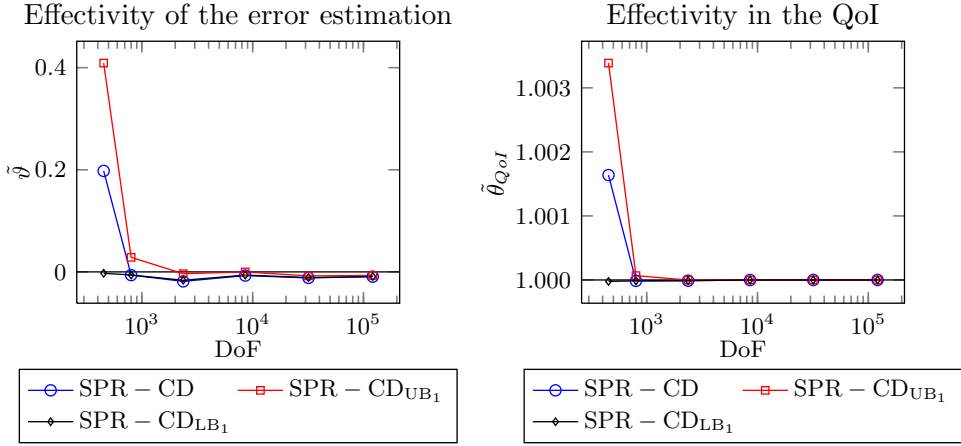


Figure 4.3: Problem 2.a. \bar{u}_n along Γ_o . Q4h. $\lambda = 1$. Evolution of the effectivity index of the error estimation $\tilde{\vartheta}$ and the effectivity in the QoI $\tilde{\theta}^{QoI}$ for the error estimates in (4.7) and the error bounds (4.50) and (4.51) obtained with the SPR-CD technique.

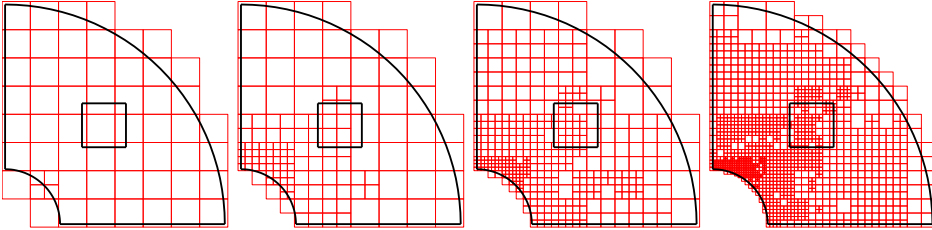


Figure 4.4: Problem 2.b. \bar{u}_x in Ω_I . Q4h. Sequence of h -adaptive refined meshes.

Table 4.3 shows for the QoI its exact error, $Q(\mathbf{e})$, the estimated error, $\tilde{\mathcal{E}}_{\text{SPR-CD}}$, the effectivity of the error estimates, $\tilde{\vartheta}_{\text{SPR-CD}}$, and the effectivity in the QoI, $\tilde{\theta}_{\text{SPR-CD}}^{QoI}$. Comparing $Q(\mathbf{e})$ and $\tilde{\mathcal{E}}_{\text{SPR-CD}}$ we can notice that both values decrease with the mesh refinement and that the estimate $\tilde{\mathcal{E}}_{\text{SPR-CD}}$ gives a good approximation to the exact error. The effectivity of the error estimator, $\tilde{\vartheta}_{\text{SPR-CD}}$, converges and is very close to the optimal value $\tilde{\vartheta} = 0$. As expected from these results, the effectivity $\tilde{\theta}_{\text{SPR-CD}}^{QoI}$ is very accurate as well, with values very close to 1.

Table 4.3: Problem 2.b. \bar{u}_x in Ω_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimator $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ for the GOA.

DoF	$Q(\mathbf{e})$	$\tilde{\mathcal{E}}_{\text{SPR-CD}}$	$\tilde{\vartheta}_{\text{SPR-CD}}$	$\tilde{\theta}_{\text{SPR-CD}}^{QoI}$
144	0.000035829	0.000066111	0.845167524	1.013529191
320	0.000008326	0.000004000	-0.519630338	0.998066960
936	0.000002248	0.000002637	0.172967518	1.000173752
3,071	0.000000494	0.000000536	0.083398620	1.000018424
10,841	0.000000178	0.000000175	-0.01265962	0.999998996
39,669	0.000000050	0.000000050	0.013370641	1.000000296

Regarding the bounding technique presented in Section 4.4, we have evaluated the results for the lower bound when $\lambda = 1$. Thus, for the bounding technique we used equation (4.50) for the upper error bound and (4.51) for the lower error bound in the error estimation of the QoI. Table 4.4 presents the global effectivity index of the error estimator in the QoI for the upper bound, $\tilde{\vartheta}_{\text{SPR-CD}_{\text{UB}_1}}$, and for the lower bound, $\tilde{\vartheta}_{\text{SPR-CD}_{\text{LB}_1}}$. The Table also presents the global effectivity index of the QoI when the FE solution for the QoI, $Q(\mathbf{u}^h)$, is corrected with the bounding estimates (4.55), $\tilde{\theta}_{\text{SPR-CD}_{\text{UB}_1}}^{QoI}$ for the upper bound and $\tilde{\theta}_{\text{SPR-CD}_{\text{LB}_1}}^{QoI}$ for the lower bound, respectively. As indicated before, positive results evaluated with equation (4.54) indicates over estimation of the error, that is, upper bounding properties; whilst negative values indicates lower bounding behavior. We observe that the right behavior holds during the whole refinement process.

Table 4.4: Problem 2.b. \bar{u}_x in Ω_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ of the bounding techniques for the GOA.

DoF	$\tilde{\vartheta}_{\text{SPR-CD}_{\text{UB}_1}}$	$\tilde{\vartheta}_{\text{SPR-CD}_{\text{LB}_1}}$	$\tilde{\theta}_{\text{SPR-CD}_{\text{UB}_1}}^{QoI}$	$\tilde{\theta}_{\text{SPR-CD}_{\text{LB}_1}}^{QoI}$
144	3.996995344	-1.411675200	1.063982716	0.977402322
320	2.136273708	-3.014450269	1.007947002	0.988786154
936	1.493632807	-0.806117295	1.001500407	0.999190227
3,071	0.521672091	-0.316171273	1.000115247	0.999930152
10,841	0.131498419	-0.193872474	1.000010431	0.999984621
39,669	0.074022078	-0.080697612	1.000001639	0.999998213

Finally, Figure 4.5 shows the results presented in Table 4.3 and Table 4.4. We observe that the evolution of the estimates and bounds smoothly converges to the expected values, 0 for $\tilde{\vartheta}$ and 1 for $\tilde{\theta}^{QoI}$. We also observe that the estimates (blue line) is always between the bounds.

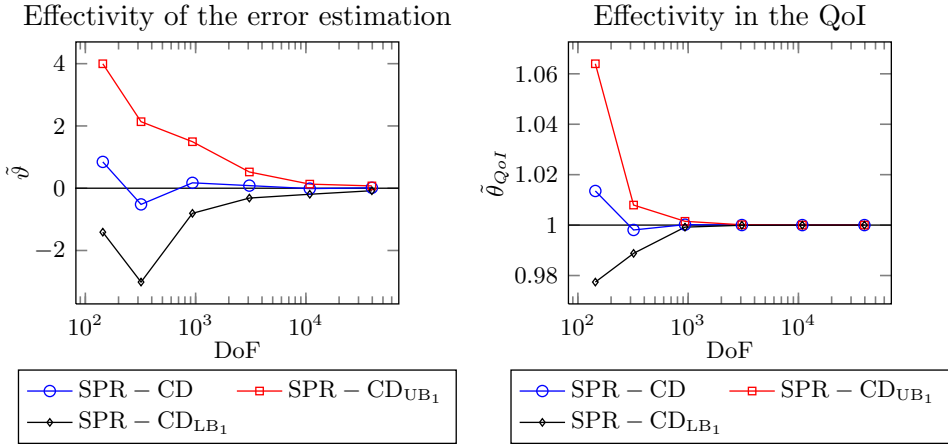


Figure 4.5: Problem 2.b. \bar{u}_x in Ω_I . $Q4h$. $\lambda = 1$. Evolution of the effectivity index of the error estimation $\tilde{\vartheta}$ and the effectivity in the QoI $\tilde{\theta}^{QoI}$ for the error estimates in (4.7) and the error bounds (4.50) and (4.51) obtained with the SPR-CD technique.

Problem 2.c: Mean stress $\bar{\sigma}_x$ in Ω_I

Consider now that the QoI is the mean stress value $\bar{\sigma}_x$ in Ω_I evaluated using (4.15) whose exact value is $0.0\bar{6}$. Figure 4.6 shows the first four meshes of bilinear elements used in the refinement process guided by the error estimated for this QoI.

We observe that the mesh refinement process is not only affecting the DoI. It is also refining the zones of the domain with influence over the solution in the DoI, that is, around the internal radius.

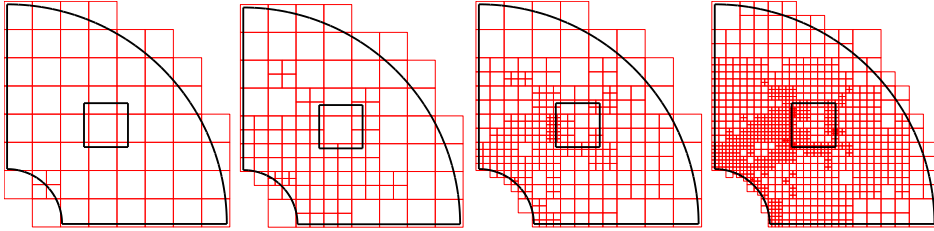


Figure 4.6: Problem 2.c. $\bar{\sigma}_x$ in Ω_I . Q4h. Sequence of h -adaptive refined meshes.

Table 4.5 presents the results obtained for the convergence analysis. For this analysis we obtain good results except for the third mesh. The effectivity of the error estimator in the QoI $\tilde{\vartheta}_{\text{SPR-CD}}$ has a very high value (-56.447333). In order to understand this behavior we should analyze how the exact error behaves. One significant difference between the error analysis in energy norm and the error analysis in the QoI is that the convergence of the error does not necessarily tends to zero monotonically. That is, the true error could have positive or negative values while it is converging to zero. This fact means that during the convergence process, it could take values near to zero as in the third mesh. Notice that $Q(\mathbf{e}) = 0.000001428$ for the third mesh and $Q(\mathbf{e}) = -0.000035007$ for the fourth mesh. Note that in this mesh as a results of $Q(\mathbf{e})$ taking a very small value ($Q(\mathbf{e}) = 0.000001428$) $\tilde{\vartheta}$ is quite big.

Table 4.5: Problem 2.c. $\bar{\sigma}_x$ in Ω_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimator $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ for the GOA.

DoF	$Q(\mathbf{e})$	$\tilde{\mathcal{E}}_{\text{SPR-CD}}$	$\tilde{\vartheta}_{\text{SPR-CD}}$	$\tilde{\theta}_{\text{SPR-CD}}^{QoI}$
144	-0.003488289	-0.003898475	-0.117589385	0.993847213
292	-0.002278696	-0.003066968	-0.345931461	0.988175912
767	0.000001428	-0.000079201	-56.447333124	0.998790566
2,512	-0.000035007	-0.000036466	-0.041663722	0.999978122
8,540	0.000006266	0.000005588	-0.108155159	0.999989835
30,331	0.000004219	0.000004183	-0.008493327	0.999999463

Regarding bounds for $\lambda = 1$ represented in Table 4.6 we observe that the upper and lower bounding properties hold. We devise again a problem with

the third mesh. Notwithstanding the assessment of bounds for the QoI in that mesh (two last columns) remains quite accurate.

Table 4.6: Problem 2.c. $\bar{\sigma}_x$ in Ω_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ of the bounding techniques for the GOA.

DoF	$\tilde{\vartheta}_{\text{SPR-CDUB}_1}$	$\tilde{\vartheta}_{\text{SPR-CDLB}_1}$	$\tilde{\theta}^{QoI}_{\text{SPR-CDUB}_1}$	$\tilde{\theta}^{QoI}_{\text{SPR-CDLB}_1}$
144	14.133077232	-17.499789766	1.739503894	0.084335106
292	7.183236654	-8.878411688	1.245526160	0.696532019
767	1,955.142709777	-2,148.719329010	1.041890673	0.953961776
2,512	8.875868515	-9.487498425	1.004660802	0.995018025
8,540	10.382070501	-10.936829891	1.000975761	0.998972100
30,331	5.320570439	-5.397654465	1.000336686	0.999658436

Finally these results have been represented in Figure 4.7 to observe the evolution of the effectivity indexes.

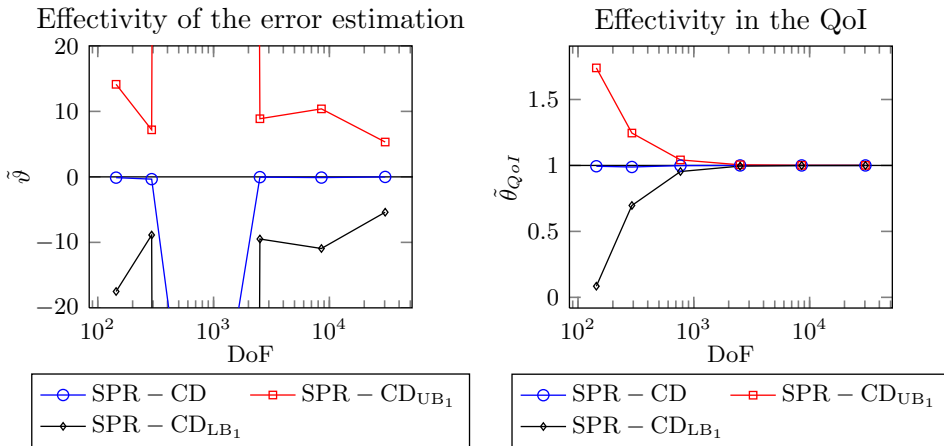


Figure 4.7: Problem 2.c. $\bar{\sigma}_x$ in Ω_I . Q4h. $\lambda = 1$. Evolution of the effectivity index of the error estimation $\tilde{\vartheta}$ and the effectivity in the QoI $\tilde{\theta}^{QoI}$ for the error estimates in (4.7) and the error bounds (4.50) and (4.51) obtained with the SPR-CD technique.

4.5.2 Problem 3: L-Shape plate

Let us consider the singular problem of a finite portion of an infinite domain with a re-entrant corner described in Section 3.5.1. In order to impose equilibrium conditions during the recovery of the displacement field by means of the SPR-CD we use the following approach. For the primal solution, on each patch, we enforce internal equilibrium in Ω , and boundary equilibrium all along the Neumann boundary. For the dual problem, we enforce internal equilibrium using the body loads and the initial strains given by (4.25) and (4.26), and homogeneous Neumann boundary conditions.

Problem 3.a: Generalized stress intensity factor K_I as QoI

In this example, we consider the GSIF K_I as the quantity of interest. Figure 4.8 shows the Cartesian meshes used to solve the primal and dual problems when the mesh is h -adapted for the evaluation of K_I . For the dual problem, we use the same Dirichlet conditions as shown in Figure 3.14 and the set of nodal forces used to extract the QoI in the annular domain Ω_I , shown in Figure 4.8, defined by a plateau function q . Function q is defined such that $q = 1$ for $r \leq r_1 = 0.6$, $q = 0$ for $r \geq r_2 = 0.8$. Further details about q function are described in Section 3.3.

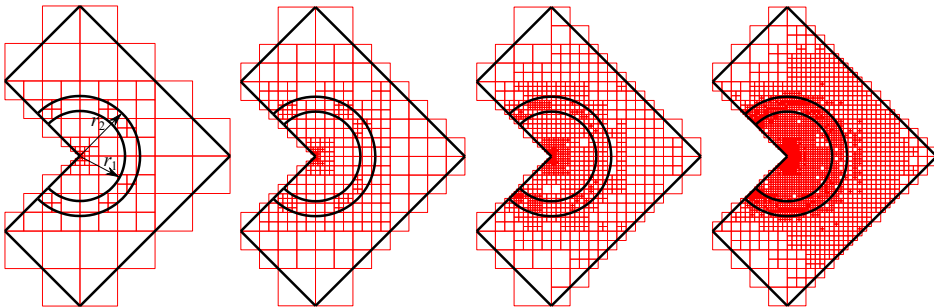


Figure 4.8: Problem 3.a. K_I . $Q4h$. Cartesian meshes with h -adaptive refinement.

Table 4.7 shows the results for the stress intensity factor K_I . Similarly to the results for other QoIs, we observe that the proposed technique provides a tight representation $\tilde{\mathcal{E}}_{\text{SPR-CD}}$ of the exact error $Q(\mathbf{e})$. Thus effectivity index for $\tilde{\vartheta}_{\text{SPR-CD}}$ is always close to the optimal value $\tilde{\vartheta} = 0$ and also, in agreement with the previous cases, the effectivity in the QoI is also close to 1.

Table 4.7: Problem 3.a. K_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimator $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ for the GOA.

DoF	$Q(\mathbf{e})$	$\tilde{\mathcal{E}}_{\text{SPR-CD}}$	$\tilde{\vartheta}_{\text{SPR-CD}}$	$\tilde{\theta}_{\text{SPR-CD}}^{QoI}$
366	0.018202430	0.020032978	0.100566180	1.001830549
1,118	0.006060242	0.007319557	0.207799593	1.001259316
3,088	0.001962537	0.001813805	-0.075785580	0.999851268
9,560	0.000434558	0.000451289	0.038501551	1.000016731
30,510	0.000149663	0.000147368	-0.015334636	0.999997705
106,300	0.000041549	0.000040514	-0.024908876	0.999998965

Table 4.8 presents the results for the bounding techniques for $\lambda = 1$. In this case, we observe that for all situations the bounding technique provides numerical upper and lower bounds of the error in the QoI. These error bounds apparently seem rather conservatives, however the bounds for the QoI are quite tight.

Table 4.8: Problem 3.a. K_I . Q4h. $\lambda = 1$. Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ of the bounding techniques for the GOA.

DoF	$\tilde{\vartheta}_{\text{SPR-CDUB}_1}$	$\tilde{\vartheta}_{\text{SPR-CDLB}_1}$	$\tilde{\theta}_{\text{SPR-CDUB}_1}^{QoI}$	$\tilde{\theta}_{\text{SPR-CDLB}_1}^{QoI}$
366	2.130063328	-1.823004948	1.038772328	0.966816881
1,118	3.212470634	-3.001114026	1.019468348	0.981812524
3,088	4.108177178	-4.210226762	1.008062451	0.991737273
9,560	6.839119537	-6.887298872	1.002971995	0.997007069
30,510	4.779258922	-4.889795715	1.000715279	0.999268178
106,300	3.875314798	-3.969754041	1.000161016	0.999835060

Table 4.9 shows the upper and lower bounds of the error in the QoI for the optimum value of λ evaluated using (4.52), for the upper error bound in the

QoI and (4.53) for the corresponding lower error bound, as explained in Section 4.4. In this case, the positiveness of the term in braces in (4.52) and the negativeness of the term in braces in (4.53) are not guaranteed.

Table 4.9: Problem 3.a. K_I . $Q4h$. Optimal value of λ . Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ of the bounding techniques for the GOA.

DoF	$\tilde{\vartheta}_{\text{SPR-CDUBOpt}}$	$\tilde{\vartheta}_{\text{SPR-CDLBOpt}}$	$\tilde{\theta}_{\text{SPR-CDUBOpt}}^{QoI}$	$\tilde{\theta}_{\text{SPR-CDLBOpt}}^{QoI}$
366	0.010391286	0.296667094	1.000189147	1.005400062
1,118	1.844731138	-1.633374530	1.011179517	0.990101356
3,088	2.156554968	-2.258604552	1.004232319	0.995567405
9,560	3.582652082	-3.630831417	1.001556870	0.998422193
30,510	2.162224282	-2.272761075	1.000323605	0.999659851
106,300	1.681654181	-1.776093424	1.000069871	0.999926205

Figure 4.9 represents the results of Table 4.7, Table 4.8 and Table 4.9. In the graphs, the black and the red lines represent the error bound obtained with expressions (4.50) and (4.51), respectively, with $\lambda = 1$. The green and brown lines represent the error bounds obtained with the optimum value of λ . The results show that for λ_{Opt} the bounding property holds when we refine the mesh, but with more accurate results than in the case where $\lambda = 1$. We have chosen this example to show the results with λ_{Opt} because has an acceptable behavior, however in general when λ_{Opt} is used, the bounding property is harder to fulfill, then we recommend to use $\lambda = 1$ when we are interested in bounding properties. It is also worthy mentioning that the accuracy of the error estimation (blue line) for this singular problem is also remarkable.

Problem 3.b: Generalized stress intensity factor K_{II} as QoI

In this example, we consider the GSIF K_{II} as the QoI. Figure 4.10 shows the Cartesian meshes used to solve the primal and dual problems during the h -adaptive refinement process guided by the error in the QoI, K_{II} . The Figure clearly shows that the refinement is performed not only in the DoI (annular

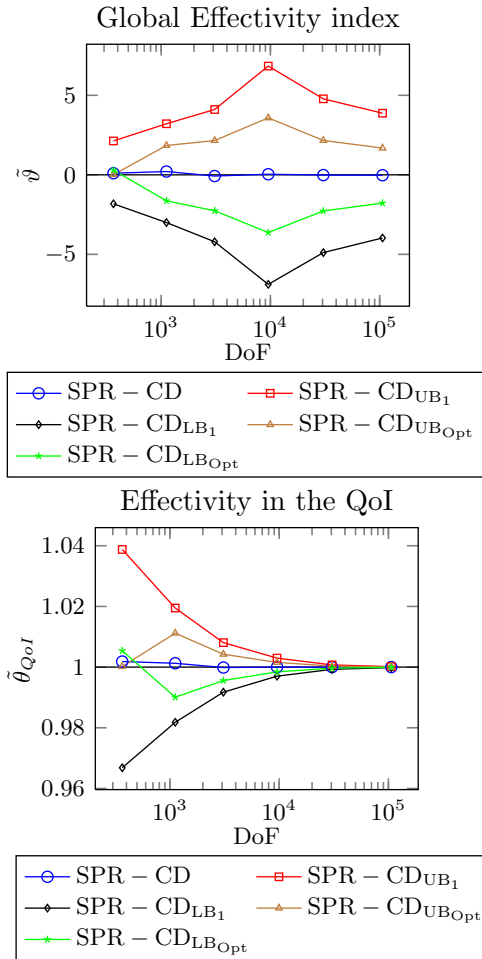


Figure 4.9: Problem 3.a. K_I . $Q4h$. Optimal value of λ and $\lambda = 1$. Evolution of the effectivity index $\tilde{\vartheta}$ and the effectivity in the QoI $\tilde{\theta}^{QoI}$ for the error estimates in (4.7) and the error bounds (4.50) and (4.51) obtained with the SPR-CD technique.

area) but also in the surroundings of the singular point. For the dual problem the Dirichlet boundary conditions and the nodal forces to extract the value of the QoI are built up as in the previous case.

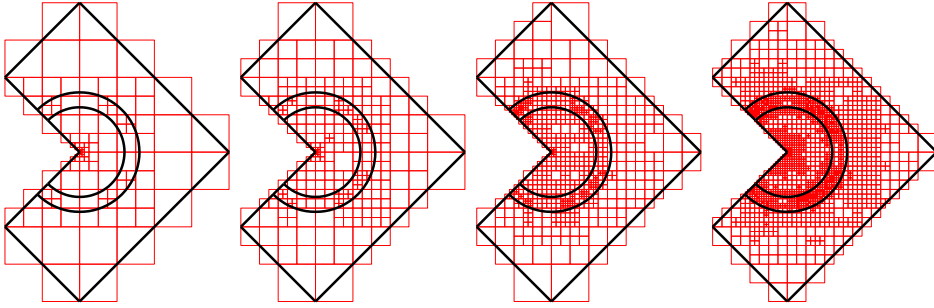


Figure 4.10: Problem 3.b. K_{II} . $Q4h$. Cartesian meshes with h -adaptive refinement.

Table 4.10 shows the results for the generalized stress intensity factor K_{II} . As in the previous cases we observe that the evaluation of the QoI is convergent because the error $Q(\mathbf{e})$ is decreasing. The error estimation $\tilde{\mathcal{E}}_{\text{SPR-CD}}$ is very accurate as its effectivity, $\tilde{\vartheta}_{\text{SPR-CD}}$, indicates. Note that the accuracy of the error estimation for the last mesh is around 1.4%.

Table 4.10: Problem 3.b. K_{II} . $Q4h$. $\lambda = 1$. Values of the global effectivity index of the error estimator $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ for the GOA.

DoF	$Q(\mathbf{e})$	$\tilde{\mathcal{E}}_{\text{SPR-CD}}$	$\tilde{\vartheta}_{\text{SPR-CD}}$	$\tilde{\theta}_{\text{SPR-CD}}^{QoI}$
366	0.006394160	0.008200164	0.282445768	1.001806004
880	0.001387519	0.001846926	0.331099229	1.000459407
2,200	0.000569657	0.000602497	0.057648895	1.000032840
6,078	0.000288879	0.000284524	-0.015077564	0.999995644
17,458	0.000078596	0.000077508	-0.013849060	0.999998912
54,078	0.000022858	0.000022532	-0.014285924	0.999999673

Table 4.11 presents the results for the bounding techniques. For this QoI, K_{II} , the upper and lower bounds are not as tight as in previous cases, their values are in general between 10 and 20. Despite of that, the bounds evaluated for the QoI remains extremely accurate and for the last mesh the accuracy of both, the upper and the lower bound is within a 0.04%.

Figure 4.11 summarizes these results. At the left hand side we observe how the bounds of the error in the QoI are maintaining their bounding property

Table 4.11: Problem 3.b. K_{II} . $Q4h$. $\lambda = 1$. Values of the global effectivity index of the error estimation $\tilde{\vartheta}$ and the corrected value of the QoI $\tilde{\theta}^{QoI}$ of the bounding techniques for the GOA.

DoF	$\tilde{\vartheta}_{\text{SPR-CDUB}_1}$	$\tilde{\vartheta}_{\text{SPR-CDLB}_1}$	$\tilde{\theta}_{\text{SPR-CDUB}_1}^{QoI}$	$\tilde{\theta}_{\text{SPR-CDLB}_1}^{QoI}$
366	4.042138161	-4.657397682	1.025846080	0.970219852
880	22.151894229	-21.741740151	1.030736183	0.969832914
2,200	28.605125564	-28.885422686	1.016295108	0.983545218
6,078	17.150217589	-16.920352049	1.004954343	0.995112061
17,458	18.172459180	-18.151958982	1.001428290	0.998573321
54,078	14.421669967	-14.494149835	1.000329656	0.999668687

while tending to increase their accuracy for finer meshes. At the right hand side, we observe the rapid increase in the accuracy of the bound for the QoI. It is also worth mentioning that in this case as in all previous situations, the high accuracy of the error estimation, $\tilde{\mathcal{E}}_{\text{SPR-CD}}$. As mentioned before, the accuracy of the plain error estimator, $\tilde{\mathcal{E}}_{\text{SPR-CD}}$, is crucial for the proposed numerical bounding techniques ((4.50) and (4.51)) to yield these results. This is because the last term (the error estimator) has to be closer to the exact error than the difference between the terms in brackets.

4.6 Conclusions

In this Chapter we have summarized the main characteristics about the error estimation in QoI's and the corresponding h -adaptive process. Traditionally the error estimation in QoI was performed using residual-based error estimators whereas our objective with this contribution is to introduce enhanced recovery-based error estimators in this field. In this Chapter we have shown some numerical results using a ZZ-type error estimator specifically adapted to evaluate the error in QoI using the recovered fields provided by the SPR-CD technique for both, the primal and dual problems. These numerical results

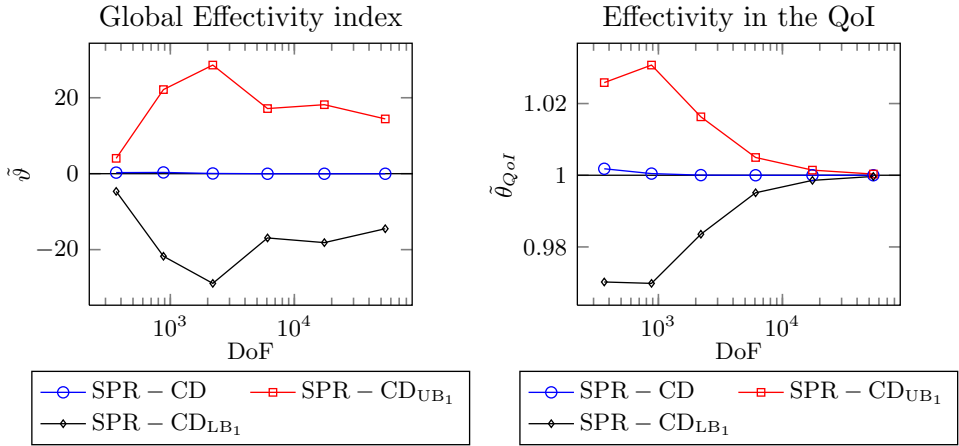


Figure 4.11: Problem 3.b. K_{II} . Q4h. $\lambda = 1$. Evolution of the effectivity index $\tilde{\vartheta}$ and the effectivity in the QoI $\tilde{\theta}^{QoI}$ for the error estimates in (4.7) and the error bounds (4.50) and (4.51) obtained with the SPR-CD technique.

show that this recovery-based error estimator is a clear alternative to the residual-based error estimators for QoI.

Additionally, we have also presented some numerical versions (and thus not guaranteed) of bounding techniques for of the error in QoI. The main advantage is that they are obtained with only one recovery process, one for the primal problem and one for the dual problem, and that all error bounds are obtained by only post-processing the recovered fields obtained with the SPR-CD technique. Thus, the computational cost for the bounding technique is small in comparison with traditional techniques. Despite of the fact that the proposed method is just a numerical version, it is able to provide sharp bounds in practice. Future work will be focused in adapting the technique that will be introduced in Section 5.3 to the QoI environment, providing guaranteed bounds with a reasonable computational cost.

Chapter 5

Upper error bounding techniques for the error in the energy norm

In this Chapter we have developed three methods to obtain guaranteed upper error bounds in energy norm, all of them are based on the use of recovery techniques. The first one, so-called FER (Fully Equilibrated Recovery), is based on a special recovery process that directly provides, into each element, an equilibrated stress field. This stress field is introduced in the ZZ error estimator to directly provides an upper error bound in energy norm. The second one, so-called RL (Recovery process and Local problems), is a mixed method which uses the advantages in accuracy of a recovery-based stress field and the bounding properties of the Neumann local problems. This approach is similar to the procedure introduced by Ladevèze *et. al.* [81]. In this case the recovery process yields a self-equilibrated recovered stress field into each element, then the stress projection to the element edges defines the local Neumann problem at each element. The statically admissible recovered stress field obtained will be used in the ZZ error estimator, directly yielding an upper

error bounds in energy norm. Finally, the third one, so-called FUB (Fast Upper Bound), is based on the ideas presented in [73] where the correction terms in (3.86) are bounded. For sake of simplicity, we will consider in this Chapter that the initial strains are zero, $\varepsilon_0 = 0$, and also the initial stresses, $\sigma_0 = 0$.

These three methods have been developed in collaboration with other researchers. We would like to thank their help provided for developing these techniques. More concretely, to J.P. Moitinho for the FER process, to P. Kerfriden for the RL process, and finally to P. Díez for the FUB process

5.1 A fully equilibrated recovery procedure. The FER technique

As previously mentioned, recovery-type error estimators were unable to directly provide error bounds in energy norm. Díez *et al.*[73] made a first attempt by adding some correction terms to the ZZ error estimator in order to ensure the upper bound property. In this Section we are going to introduce a scheme for a new recovery procedure that, directly using a ZZ-type error estimator, yields upper error bounds in energy norm.

When the recovered stress field used in the ZZ error estimator is statically admissible, then it yields an upper error bound, see Section 3.4. In this Section we intend to evaluate an statically admissible stress field for upper error bounding purposes. The recovery technique presented in this Section, so-called FER (FER stands for Fully Equilibrated Recovery), will directly provide a statically admissible stress field. The recovered stress field will fulfill the internal equilibrium equation, the boundary equilibrium equation and equilibrium of traction along the internal element edges. Note that the normal stress tangent to the boundary does not necessarily need to be continuous along the element edges. In case that the stress field were continuous, that is, all components

were continuous along element edges, and also equilibrated it will be the exact stress solution. Therefore, the FER technique relaxes the stress continuity of the SPR-CD that prevents statical admissibility.

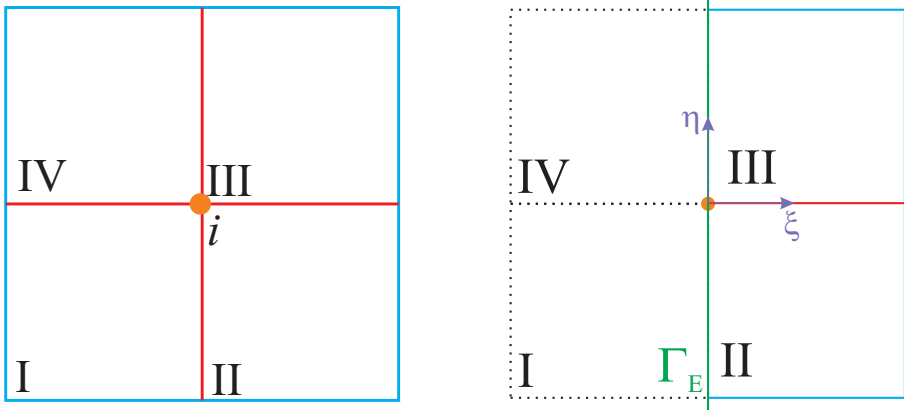
5.1.1 Recovery procedure

The FER technique, is based on the SPR technique developed by Zienkiewicz and Zhu [7]. In the FER, as in the SPR, we create patches of elements \mathcal{P}^i with the elements connected to the vertex nodes i , so-called patch assembly nodes, as in Section 3.3. There are two main differences between the traditional SPR and the FER:

- In the SPR each recovered stress component is represented by a single polynomial on each patch, while for the FER a polynomial surface is fitted for each stress component on each element k of the patch. Figure 5.1 shows two different patches for the case of Cartesian grids used in this thesis. A different polynomial surface for each stress component at elements $k = I, II, III, IV$ will be evaluated considering the appropriate constraints.
- The second difference is that the SPR technique builds up the global recovered field at each element by adding the contributions of each patch using the partition of unity concept. However, in the FER the global recovered field is obtained directly adding the contributions of all patches $\sigma_{\text{FER}}^* = \sum_{i=1}^{n_v} \hat{\sigma}_i^{*k}$ connected to one element k since the partition of unity is implicit in the functional (5.1). Note that when we apply the constraints for internal and boundary equilibrium the problem loads will be also affected by the shape functions N_i^k used in the partition of the unity.

For the statical admissibility condition, we add the constraints that are necessary to enforce the required continuity and equilibrium in the recovered so-

lution using a point collocation approach. The number of points will depend on the degree of the recovered field. This is obtained by adding continuity of traction along the internal edges (red edges). We enforce the recovered traction to zero along the external edges (blue edges) and finally we enforce the internal equilibrium equation at each element, separately. The recovery process will be described below in more detail.



(a) Patch of elements. Assembly node i represented in orange. (b) Patch in contact with a boundary (green line) of the domain.

Figure 5.1: Internal patch formed by 4 elements (left) and patch in contact with the boundary formed by 2 elements (right).

Let us assume a patch in a Cartesian grid composed by 4 elements of the same size around the node i , see Figure 5.1a. We will minimize the following functional on each of the k elements of the patch:

$$\Phi_{FER} := \int_{\mathcal{P}_k^i} \left(\hat{\boldsymbol{\sigma}}_k^*(\mathbf{x}) - N_i^k(\mathbf{x}) \boldsymbol{\sigma}^h(\mathbf{x}) \right)^2 d\Omega \quad k = I, II, III, IV \quad (5.1)$$

where $N_i^k(\mathbf{x})$ is the linear shape function of the node i , in element k , $\boldsymbol{\sigma}^h$ is the FE stress field and $\hat{\boldsymbol{\sigma}}_k^* = \mathbf{P}_k \mathbf{a}_k$ is the recovered stress field for the element k , where $\mathbf{a}_k = \{\mathbf{a}_k^{xx}, \mathbf{a}_k^{yy}, \mathbf{a}_k^{xy}\}^T$ are the coefficients for each stress component and $\mathbf{P}_k(\mathbf{x})$ is the matrix for the polynomial expansion $\mathbf{p}_k(\mathbf{x}) =$

$\{x^m y^n : m, n \leq q\}_k$, where q is the polynomial degree.

$$\mathbf{P}_k(\mathbf{x}) = \begin{bmatrix} \mathbf{p}_k(\mathbf{x}) & 0 & 0 \\ 0 & \mathbf{p}_k(\mathbf{x}) & 0 \\ 0 & 0 & \mathbf{p}_k(\mathbf{x}) \end{bmatrix} \quad (5.2)$$

For each element k , integrating numerically after the minimization of (5.1) we obtain the following expression:

$$\sum_j^{NIntP} \mathbf{P}_k^T(\mathbf{x}_j) \mathbf{P}_k(\mathbf{x}_j) |\mathbf{J}(\mathbf{x}_j)| \omega_j \mathbf{a}_k = \sum_j^{NIntP} \mathbf{P}_k^T(\mathbf{x}_j) N_i^k(\mathbf{x}_j) \boldsymbol{\sigma}^h(\mathbf{x}_j) |\mathbf{J}(\mathbf{x}_j)| \omega_j \quad (5.3)$$

where $|\mathbf{J}|$ is the Jacobian of the coordinates transformation, ω is the weight of each integration point and $NIntP$ is the number of integration points. This expression yields a linear system of equations for each element of the patch $\mathbf{M}_k \mathbf{a}_k = \mathbf{g}_k$. Due to the constraints we have imposed, we need some interaction between the different recovered stress fields. Thus, we assemble all four systems together and we obtain the following linear system for the patch:

$$\begin{bmatrix} \mathbf{M}_I & 0 & 0 & 0 \\ 0 & \mathbf{M}_{II} & 0 & 0 \\ 0 & 0 & \mathbf{M}_{III} & 0 \\ 0 & 0 & 0 & \mathbf{M}_{IV} \end{bmatrix} \begin{Bmatrix} \mathbf{a}_I \\ \mathbf{a}_{II} \\ \mathbf{a}_{III} \\ \mathbf{a}_{IV} \end{Bmatrix} = \begin{Bmatrix} \mathbf{g}_I \\ \mathbf{g}_{II} \\ \mathbf{g}_{III} \\ \mathbf{g}_{IV} \end{Bmatrix} \Rightarrow \mathbf{M} \mathbf{a} = \mathbf{g} \quad (5.4)$$

Internal equilibrium constraint

In contrast with the SPR-CD presented in Section 3.3, where the internal equilibrium constraint is considered by using (3.38), in the SPR-FE we have to take into account the partition of unity introduced in the functional (5.1). Therefore, it will affect to the body forces \mathbf{b} and also it will bring up a new term, first introduced in [88]: *the fictitious body forces*, $\mathbf{L}^T N_i^k \boldsymbol{\sigma}^h$. Their role is to ensure that the forces applied to each patch satisfy global equilibrium

for the isolated patch. If it were not considered, then the system of equations at the patch would generally have no solution. Nevertheless, when we sum up the contributions of the four patches of a single element k these terms will sum to zero, canceling their effect at a global level.

Then, the internal equilibrium equation to impose in this case is:

$$\mathbf{L}^T \hat{\boldsymbol{\sigma}}_i^{*k}(\mathbf{x}_j) + N_i^k \tilde{\mathbf{b}}(\mathbf{x}_j) = \mathbf{L}^T N_i^k \boldsymbol{\sigma}^h(\mathbf{x}_j) \quad \forall \mathbf{x}_j \in \mathcal{P}_k^i \quad (5.5)$$

at each element k . These constraints are independently enforced in all elements. This generates the internal equilibrium matrix for each element, \mathbf{C}_k^{IEE} , and the independent term \mathbf{h}_k^{IEE} shown in (5.6). This matrix is assembled to the previous system (5.4), yielding the new constrained system (5.7).

$$\mathbf{C}^{IEE} = \begin{bmatrix} \mathbf{C}_I^{IEE} & 0 & 0 & 0 \\ 0 & \mathbf{C}_{II}^{IEE} & 0 & 0 \\ 0 & 0 & \mathbf{C}_{III}^{IEE} & 0 \\ 0 & 0 & 0 & \mathbf{C}_{IV}^{IEE} \end{bmatrix} \quad \mathbf{h}_k^{IEE} = \begin{Bmatrix} \mathbf{h}_I^{IEE} \\ \mathbf{h}_{II}^{IEE} \\ \mathbf{h}_{III}^{IEE} \\ \mathbf{h}_{IV}^{IEE} \end{Bmatrix} \quad (5.6)$$

$$\begin{bmatrix} \mathbf{M} & (\mathbf{C}^{IEE})^T \\ \mathbf{C}^{IEE} & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \boldsymbol{\lambda}^{IEE} \end{Bmatrix} = \begin{Bmatrix} \mathbf{g} \\ \mathbf{h}^{IEE} \end{Bmatrix} \quad (5.7)$$

External patch edge constraint

The next step is to add the constraints along the external boundaries of the patch, that is, the constraints along the blue edges in Figure 5.1. These constraints will ensure tractions continuity when we sum up the contributions from the patches related to an element. Since the partition of unity function is zero at the external edges of the patch, the equation to be imposed is:

$$\mathbf{G} \hat{\boldsymbol{\sigma}}_i^{*k}(\mathbf{x}_j) = 0 \quad \forall \mathbf{x}_j \in \Gamma_N \cap \mathcal{P}_k^i \quad (5.8)$$

This generates for each element the matrix \mathbf{C}_k^{BE} (5.10) and the independent term $\mathbf{h}_k^{BE} = 0$.

$$\mathbf{C}^{EB} = \begin{bmatrix} \mathbf{C}_I^{EB} & 0 & 0 & 0 \\ 0 & \mathbf{C}_{II}^{EB} & 0 & 0 \\ 0 & 0 & \mathbf{C}_{III}^{EB} & 0 \\ 0 & 0 & 0 & \mathbf{C}_{IV}^{EB} \end{bmatrix} \quad (5.9)$$

The linear system can be rewritten as follows.

$$\begin{bmatrix} \mathbf{M} & (\mathbf{C}^{IEE})^T & (\mathbf{C}^{EB})^T \\ \mathbf{C}^{IEE} & 0 & 0 \\ \mathbf{C}^{EB} & 0 & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \lambda^{IEE} \\ \lambda^{EB} \end{Bmatrix} = \begin{Bmatrix} \mathbf{g} \\ \mathbf{h}^{IEE} \\ 0 \end{Bmatrix} \quad (5.10)$$

Equilibrium along internal edges of the patch

Finally, it is also necessary to add the constraints along the internal boundaries of the patch (red edges), *i.e.* the interfaces between elements. These are also used to ensure traction continuity along the element interface. The equation to be imposed is:

$$\mathbf{G}_k \hat{\boldsymbol{\sigma}}_i^{*k}(\mathbf{x}_j) + \mathbf{G}_l \hat{\boldsymbol{\sigma}}_i^{*l}(\mathbf{x}_j) = 0 \quad \forall \mathbf{x}_j \in \Gamma_N \cap \mathcal{P}_{k \cap l}^i \quad (5.11)$$

where $k \neq l$ and $\mathbf{G}_k = -\mathbf{G}_l$, generating \mathbf{C}^{IB} (5.12) and the corresponding r.h.s is again null, $\mathbf{h}^{IB} = 0$.

$$\mathbf{C}^{IB} = \begin{bmatrix} \mathbf{C}_I^{IB} & \mathbf{C}_{II}^{IB} & 0 & 0 \\ \mathbf{C}_I^{IB} & 0 & 0 & \mathbf{C}_{IV}^{IB} \\ 0 & \mathbf{C}_{II}^{IB} & \mathbf{C}_{III}^{IB} & 0 \\ 0 & 0 & \mathbf{C}_{III}^{IB} & \mathbf{C}_{IV}^{IB} \end{bmatrix} \quad (5.12)$$

A particular situation occurs when an internal edge coincides with a boundary where the tractions are prescribed. In Figure 5.1b we illustrate such a boundary (green line), which is internal to the patch. This is a typical situation

when the assembly nodes (orange point) are over the boundary. In this case, the equations to be imposed have to take into account the Neumann boundary conditions:

$$\mathbf{G}_k \hat{\boldsymbol{\sigma}}_i^{*k}(\mathbf{x}_j) = N_i^k(\mathbf{x}_j) \mathbf{t}(\mathbf{x}_j) \quad \forall \mathbf{x}_j \in \Gamma_N \cap \mathcal{P}_{k\cap l}^i \quad (5.13)$$

When the boundary condition is non-homogeneous the corresponding term in the r.h.s. is generally not null, $\mathbf{h}^{IB} \neq 0$.

5.1.2 Comments about the resolution of the system of equations

Adding all constraints to (5.4) we obtain the following linear system to solve at each patch:

$$\begin{bmatrix} \mathbf{M} & (\mathbf{C}^{IEE})^T & (\mathbf{C}^{EB})^T & (\mathbf{C}^{IB})^T \\ \mathbf{C}^{IEE} & 0 & 0 & 0 \\ \mathbf{C}^{EB} & 0 & 0 & 0 \\ \mathbf{C}^{IB} & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \boldsymbol{\lambda}^{IEE} \\ \boldsymbol{\lambda}^{EB} \\ \boldsymbol{\lambda}^{IB} \end{Bmatrix} = \begin{Bmatrix} \mathbf{g} \\ \mathbf{h}^{IEE} \\ 0 \\ \mathbf{h}_k^{IB} \end{Bmatrix} \quad (5.14)$$

Equation (5.14) can be rewritten as $\mathbf{M}_C \mathbf{a}_C = \mathbf{g}_C$, where C indicates that the constraints are included.

The basis \mathbf{p} for the stress field has to be able to represent all constraints to guarantee the statical admissibility property. Thus, we need to analyze the minimum degree required for the stress field to guarantee that the system of equations (5.14) is solvable. This procedure cannot be directly applied to bilinear FE elements since they do not guarantee the rotational equilibrium of the patch [123]. We will then consider bi-quadratic elements (Q8), therefore the FE stress field, $\boldsymbol{\sigma}^h$, will have quadratic terms. The term $N_i^k(\mathbf{x})$ in (5.1) is bilinear and its divergence will have linear terms. Analyzing the constraints:

- *Internal equilibrium constrain:* the virtual body forces, $\mathbf{L}^T N_i^k \boldsymbol{\sigma}^h$, has cubic terms and assuming that the body forces \mathbf{b} is linear the required degree for the recovered stress field is 4.
- *External patch edge constrain:* in this case there is not any requirement.
- *Internal patch edge constrain:* is this case for quadratic tractions over the boundary we require also a 4th order recovered stress representation.

Then, for linear body forces and quadratic traction over the boundary we need at least a 4th order polynomial interpolation for \mathbf{p} because of the fictitious body forces.

Table 5.1 shows the total size of the system to be solve at each node. We have to pay attention to the “effective” number of “free coefficients” (last column), which represents the difference between the number of “Coefficients” \mathbf{a} and the number of independent constraints. For degree 4 the number of constraints (192+80+40 = 312) is higher than the number of coefficients (300), therefore some constraints are linearly dependent. In fact there are always linearly dependent constraints, as indicated by the difference between the dimension of the system and its rank. For example, we obtain 16 effective free coefficients for degree 4.

Degree	Coeff	IEE	EB	IB	Sys size	Rank	Eff free coef
4	300	192	80	40	612	584	16
5	432	280	96	48	856	828	40
6	588	384	112	56	1140	1108	68

Table 5.1: Number of coefficients and constraints

To solve the system of equations in (5.14) we use the Singular Value Decomposition (SVD) technique. Using the SVD, any $m \times n$ matrix \mathbf{M}_C can be decomposed in three matrices \mathbf{U} , \mathbf{S} , and \mathbf{V} as represented in (5.15). \mathbf{U} is an $m \times m$ unitary matrix ($U^T U = I$, I the identity matrix), the matrix \mathbf{S} is an $m \times n$ diagonal matrix with nonnegative real numbers on the diagonal, and

the $n \times n$ unitary matrix \mathbf{V}^T denotes the conjugate transpose of \mathbf{V} . Then, the linear system could be solved by using (5.16), even if we have linear dependent constraints.

$$\mathbf{M}_C = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{5.15}$$

$$\mathbf{a}_C = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{g}_C \tag{5.16}$$

5.2 An error estimator combining a recovery-based procedure with equilibrated local problem resolution. The RL technique

In this Section we propose a methodology to obtain also guaranteed upper error bounds solving local equilibrated problems to obtain a statically admissible stress field. Traditionally, the main difficulty for local problem resolution has been the evaluation of a suitable traction field $\tilde{\mathbf{t}}_L$ for solving the local problem at element level. Similar problems are reported for the implicit residual error estimators and also for the CRE method, see Section 3.1. In this Section we propose to generate the element traction field $\tilde{\mathbf{t}}_L$ with a SPR-based recovery procedure which guarantees that the local problems at each element $K \in \mathcal{T}$, are solvable (self-equilibrated). The statically admissible stress field obtained from the solution of these local problems will be used in a ZZ-type error estimator equation to directly provides an upper error bound in energy norm.

The RL technique consists of two steps. In the first step an auxiliary recovered stress field σ_{aux}^* is evaluated. This field will be used to evaluate the traction field along the edges of each element defining the local Neumann problem. The second step will be to solve the local Neumann problem to obtain the statically admissible stress field σ_{RL}^* used in the ZZ error estimator to obtain an upper error bound.

5.2.1 Local problem formulation

Let us first define the local problem formulation, the second step of this error estimation process, to formally obtain the appropriate requirements that the auxiliary stress field $\boldsymbol{\sigma}_{\text{aux}}^*$ must satisfy. In order to evaluate an upper error bound in energy norm we obtain a statically admissible stress field as a solution of a local Neuman problem over each element of the mesh. The local problem at each element will be obtained integrating the weak form of the elasticity problem defined in (2.8) into the corresponding element domain Ω^K :

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u})^T \mathbf{D}^{-1} \boldsymbol{\sigma}(\mathbf{v}) d\Omega \\ &= \sum_K \int_{\Omega^K} \boldsymbol{\sigma}(\mathbf{u})^T \mathbf{D}^{-1} \boldsymbol{\sigma}(\mathbf{v}) d\Omega = \sum_K a_K(\mathbf{u}, \mathbf{v}) \quad \forall \mathbf{v} \in V \end{aligned} \quad (5.17)$$

integrating by parts the contribution of each single element we have:

$$a_K(\mathbf{u}, \mathbf{v}) = \int_{\partial\Omega^K} (\mathbf{G}\boldsymbol{\sigma}(\mathbf{u}))^T \mathbf{v} d\Gamma - \int_{\Omega^K} (\mathbf{L}^T \boldsymbol{\sigma}(\mathbf{u}))^T \mathbf{v} d\Omega \quad \forall \mathbf{v} \in V \quad (5.18)$$

Note that the projection of the exact stress field over the Neumann boundary, coincides with the Neumann conditions, that is $\mathbf{G}\boldsymbol{\sigma}(\mathbf{u}) = \mathbf{t}$. Furthermore, the body forces are data in the elasticity problem, hence we can consider $-\mathbf{L}^T \boldsymbol{\sigma}(\mathbf{u}) = \mathbf{b}$. Then we rewrite (5.18) as:

$$\begin{aligned} a_K(\mathbf{u}, \mathbf{v}) &= \int_{\partial\Omega^K \cap \Gamma_N} \mathbf{t}^T \mathbf{v} d\Gamma + \int_{\partial\Omega^K \setminus \Gamma_N} (\mathbf{G}\boldsymbol{\sigma}(\mathbf{u}))^T \mathbf{v} d\Gamma \\ &\quad + \int_{\Omega^K} \mathbf{b}^T \mathbf{v} d\Omega \quad \forall \mathbf{v} \in V \end{aligned} \quad (5.19)$$

The local problem defined in (5.19) would be solvable if the tractions over the non-Neumann element edges were known, but in practice they are unknown, then that problem is generally unsolvable. The idea here, as in Ladevèze *et al.* [81], is to generate an alternative Neumann local problem which reads:

$$\begin{aligned} a_K(\mathbf{u}_{\text{RL}}^*, \mathbf{v}) &= \int_{\partial\Omega^K \cap \Gamma_N} \mathbf{t}^T \mathbf{v} d\Gamma + \int_{\partial\Omega^K \setminus \Gamma_N} \tilde{\mathbf{t}}_L^T \mathbf{v} d\Gamma \\ &\quad + \int_{\Omega^K} \mathbf{b}^T \mathbf{v} d\Omega \quad \forall \mathbf{v} \in V \end{aligned} \quad (5.20)$$

where the traction field $\tilde{\mathbf{t}}_L^T$ over the non-Neumann element edges $\partial\Omega^K \setminus \Gamma_N$ will be obtained by the projection of the auxiliary recovered field over these element edges, that is $\tilde{\mathbf{t}}_L^T = \mathbf{G}\boldsymbol{\sigma}_{\text{aux}}^*$ on $\partial\Omega^K \setminus \Gamma_N$. The exact solution of the problem defined in (5.20) yields a statically admissible stress field per element, $\boldsymbol{\sigma}_{\text{RL}}^* = \mathbf{D}\mathbf{L}\mathbf{u}_{\text{RL}}^*$, which will be the stress field used in the ZZ-type error estimator to provide upper error bounds in energy norm.

To strictly guarantee the upper bounding property we need to solve the local problems exactly, or at least to solve them by using the stress-based formulation of the FEM. This is not desirable as it requires a second implementation of the FEM for this purpose. As an alternative, it is of common practice to use the standard FEM for the local problem but with a higher interpolation degree. In our particular implementation we have chosen $p + 2$, obtaining satisfactory results, being p the FE interpolation degree.

So far, we have presented a local Neumann problem whose stress solution, $\boldsymbol{\sigma}_{\text{RL}}^*$, is statically admissible. Now we have to consider the requirements to guarantee the solvability of the local problem (5.20). The condition is that the traction field, $\tilde{\mathbf{t}}_L^T$ over $\partial\Omega^K \setminus \Gamma_N$ has to be in equilibrium with the other loads in the element. Alternatively, applying the virtual work principle the condition can be seen as that the rigid body motions must not generate any work. This can be expressed in the following equation:

$$\int_{\partial\Omega^K \setminus \Gamma_N} (\mathbf{G}\boldsymbol{\sigma}_{\text{aux}}^*)^T \mathbf{v} \, d\Gamma + \int_{\partial\Omega^K \cap \Gamma_N} \mathbf{t}^T \mathbf{v} \, d\Gamma + \int_{\Omega^K} \mathbf{b}^T \mathbf{v} \, d\Omega = 0 \quad \forall \mathbf{v} \in V^{\text{RBM}} \quad (5.21)$$

where V^{RBM} is the Rigid Body Motion space. Then, the procedure to evaluate the auxiliary recovered field, $\boldsymbol{\sigma}_{\text{aux}}^*$, will impose (5.21) during the recovery process as it will be shown next.

5.2.2 The recovery procedure for the auxiliary stress field

The recovery procedure used for building up the auxiliary recovered field $\boldsymbol{\sigma}_{\text{aux}}^*$ is based on the SPR-C implementation. The initial non-constrained system (3.34) is built as usual, that is, minimizing and discretizing the functional defined in (3.35). The difference between this procedure and the SPR-C recovery process are the constraints added to the linear system of equations to solve at each patch i . This Section will be focused in developing the constraints needed to guarantee the solvability of the local problems.

In Figure 5.2 we present a patch with assembly node i . In cgFEM, all patches have square elements and at most four elements, $k = I, II, III, IV$, k indicating the relative position of the element with respect to the assembly node. The recovery procedure will provide a local stress field $\hat{\boldsymbol{\sigma}}_{\text{aux}}^{*i}$ associated to the patch i . To build up the global stress field $\boldsymbol{\sigma}_{\text{aux}}^*$ we use the Conjoint Polynomial enhancement:

$$\boldsymbol{\sigma}_{\text{aux}}^*(\mathbf{x}) = \sum_{i=1}^{n_v} N_i(\mathbf{x}) \hat{\boldsymbol{\sigma}}_{\text{aux}}^{*i}(\mathbf{x}) \quad (5.22)$$

To guarantee the solvability of the local problems we need all loads applied to the local problem to be in equilibrium, see (5.21). Substituting in (5.21) the definition of the auxiliary stress field shown in (5.22) and applying the partition of the unity to the other integrals, we obtain:

$$\begin{aligned} & \sum_{i=1}^{n_v} \int_{\partial\Omega^K \setminus \Gamma_N} N_i^k \mathbf{G} \hat{\boldsymbol{\sigma}}_{\text{aux}}^{*i} \mathbf{v} \, d\Gamma + \sum_{i=1}^{n_v} \int_{\partial\Omega^K \cap \Gamma_N} N_i^k \mathbf{t}^T \mathbf{v} \, d\Gamma \\ & + \sum_{i=1}^{n_v} \int_{\Omega^K} N_i^k \mathbf{b}^T \mathbf{v} \, d\Omega = \sum_{i=1}^{n_v} \int_{\Omega^K} (\mathbf{L}^T N_i^k \boldsymbol{\sigma}^h)^T \mathbf{v} \, d\Omega = 0 \quad \forall \mathbf{v} \in V^{\text{RBM}} \end{aligned} \quad (5.23)$$

We have a new term in (5.23), $\sum_{i=1}^{n_v} \int_{\Omega^K} (\mathbf{L}^T N_i^k \boldsymbol{\sigma}^h)^T \mathbf{v} \, d\Omega$. This term is the virtual work associated to the *fictitious body forces* already reported in 5.1. This term accounts for the local equilibrium when N_i^k is applied to the constraint. Note that when the patch contributions are added this term vanishes,

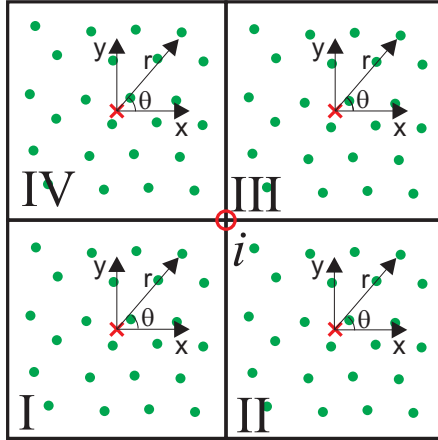


Figure 5.2: Local SdR and patch. The SdR is used to define the rotation virtual displacement

but it is important when the patch contributions are used independently. The next step consist in considering, independently, the contribution of each patch i to (5.23). This yields the constraint to impose into each patch i for each of the elements k of the patch:

$$\begin{aligned}
 \int_{\partial\Omega^K \setminus \Gamma_N} N_i^k (\mathbf{G}\boldsymbol{\sigma}_{\text{aux}}^{*i})^T \mathbf{v} \, d\Gamma + \int_{\partial\Omega^K \cap \Gamma_N} N_i^k \mathbf{t}^T \mathbf{v} \, d\Gamma + \int_{\Omega^K} N_i^k \mathbf{b}^T \mathbf{v} \, d\Omega \\
 = \int_{\Omega^K} (\mathbf{L}^T N_i^k \boldsymbol{\sigma}^h)^T \mathbf{v} \, d\Omega \quad \forall \mathbf{v} \in V^{\text{RBM}} \quad (5.24)
 \end{aligned}$$

Note that when the auxiliary recovered field is evaluated with (5.22) for a single element, the patch contributions are taken into account affected by the partition on the unity. The same will occur with the constraint defined in

(5.24), each of the integrals in (5.24) will be added, then:

$$\sum_{J=1}^{n_v} \int_{\partial\Omega^K \setminus \Gamma_N} N_J^k (\mathbf{G}\hat{\boldsymbol{\sigma}}_{\text{aux}}^{*J})^T \mathbf{v} \, d\Gamma = \int_{\partial\Omega^K \setminus \Gamma_N} (\mathbf{G}\boldsymbol{\sigma}_{\text{aux}}^*)^T \mathbf{v} \, d\Gamma \quad (5.25)$$

$$\sum_{J=1}^{n_v} \int_{\partial\Omega^K \cap \Gamma_N} N_J^k \mathbf{t}^T \mathbf{v} \, d\Gamma = \int_{\partial\Omega^K \cap \Gamma_N} \mathbf{t}^T \mathbf{v} \, d\Gamma \quad (5.26)$$

$$\sum_{J=1}^{n_v} \int_{\Omega^K} N_J^k \mathbf{b}^T \mathbf{v} \, d\Omega = \int_{\Omega^K} \mathbf{b}^T \mathbf{v} \, d\Omega \quad (5.27)$$

$$\sum_{J=1}^{n_v} \int_{\Omega^K} (\mathbf{L}^T N_J^k \boldsymbol{\sigma}^h)^T \mathbf{v} \, d\Omega = 0 \quad (5.28)$$

recovering the initial condition (5.21) for the solvability of the local problem at each element of the mesh.

During the recovery of the patch i we need to fulfill (5.24) at each element k of the patch for all rigid body motion space. For instance we took a basis of the rigid body motion space consisting of a displacement in x direction, another in y direction and a pure rotation over the element center:

$$\mathbf{v} = \begin{cases} [1, 0]^T \\ [0, 1]^T \\ [-r \cdot \sin(\theta), r \cdot \cos(\theta)]^T \end{cases} \quad (5.29)$$

where r and θ are defined in Figure 5.2 for each element of the patch. So, for each patch of four elements we have twelve constraints, $n_{\text{sole}} = 12$, added via the Lagrange Multipliers technique as in Section 3.2.6, to ensure solvability of the local problem. The constraint for any rigid body motion \mathbf{v} described in (5.29) reads:

$$\begin{aligned} c^{\text{sol}} : & \int_{\partial\Omega^K \setminus \Gamma_N} N_i^k (\mathbf{G}\hat{\boldsymbol{\sigma}}_{\text{aux}}^{*i})^T \mathbf{v} \, d\Gamma = - \int_{\partial\Omega^K \cap \Gamma_N} N_i^k \mathbf{t}^T \mathbf{v} \, d\Gamma \\ & - \int_{\Omega^K} N_i^k \mathbf{b}^T \mathbf{v} \, d\Omega + \int_{\Omega^K} (\mathbf{L}^T N_i^k \boldsymbol{\sigma}^h)^T \mathbf{v} \, d\Omega \quad \forall k \in \mathcal{P}^i \end{aligned} \quad (5.30)$$

Furthermore, in patches containing Neumann boundaries, we could also add the constraint equation (3.39) described in Section 3.2.6, to further increase

the quality of the recovered field over the boundary. Finally the functional to be minimized at each patch \mathcal{P}^i reads:

$$\Phi(\mathbf{A}, \boldsymbol{\lambda}) := \Phi'(\mathbf{A}) + \sum_j^{n_{bee}} \lambda_j (c^{\text{int}}(\mathbf{x}_j)) + \sum_j^{n_{sole}} \lambda_j (c^{\text{sol}}) \quad (5.31)$$

Considerations about the resolution of the system of equations

In this case, as in the SPR-C method, the information from the FE solution is considered using the functional described in (3.35), after numerical integration. This will yield matrix \mathbf{M} and the right hand side vector \mathbf{H} defined in (3.37). Then, we will add the constraint equations previously indicated. Finally, optimizing the functional (5.31) we obtain a linear system of equations for each patch of elements i , containing a constraint matrix \mathbf{C} and the corresponding right hand side vector $\boldsymbol{\Lambda}$ similar to the one described in (3.42).

Now we should proceed to analyze the set of equations playing a role in this case:

- Boundary equilibrium equation: the constrain equation is described in (3.39) and imposed as in the SPR-C procedure.
- Solvability constraint: the constraint to be imposed is described in (5.24). This equation has to be imposed once for each element of the patch and for each virtual displacement described in (5.29). That means a total of 12 equation per patch, independently of the degree of the recovered field. It is also important to point out that some of these equations are linearly dependent, increasing the difficulty for solving the patch system.

The system is solved with the SVD technique as for the FER method. However in this case, the size of the system is considerably smaller, yielding a better

performance. For instance, for a 2^{nd} order recovered field we have a total of 36 equations.

Note that, because of the lack of rotational equilibrium for the linear (Q4) FE solution [123], this method can be directly applied only for quadratic elements (Q8), as in the FER. For Q4 elements, it requires a post-processing of the FE solution in order to correct the lack of rotational equilibrium. See [123] for further information.

5.3 A fast SPR-based upper bounding technique. The FUB technique

This technique is based on the used of the SPR-CD technique and the correction terms presented in Section 3.4 that takes into account the lacks of internal and boundary equilibrium. These correction terms (5.32) require the exact displacement solution to be evaluated. Thus, in the previous numerical approach to obtain the upper error bound in energy norm presented in that Section, we pretended to estimate the correction terms using the recovered displacement field provided by the SPR-CD technique. In this Section we are not interested in estimating them but in bounding them, thus obtaining guaranteed upper error bounds in energy norm. As it will be shown through this Section the computational cost needed to bound these correction terms is negligible in comparison with the computational cost required to obtain the numerical solution of the elasticity problem, as a difference from the two previous techniques.

$$- 2 \int_{\Omega} (\mathbf{s}_{\sigma}^*)^T \mathbf{e} \, d\Omega \quad \text{and} \quad - 2 \int_{\Gamma_N} (\mathbf{r}_{\sigma}^*)^T \mathbf{e} \, d\Gamma \quad (5.32)$$

These correction terms are first bounded with the Cauchy-Schwarz inequality.

$$\begin{aligned} \left| \int_{\Omega} (\mathbf{s}_{\sigma}^*)^T \mathbf{e} \, d\Omega \right| &\leq \|\mathbf{s}_{\sigma}^*\|_{L^2(\Omega)} \|\mathbf{e}\|_{L^2(\Omega)} \\ \left| \int_{\Gamma_N} (\mathbf{r}_{\sigma}^*)^T \mathbf{e} \, d\Gamma \right| &\leq \|\mathbf{r}_{\sigma}^*\|_{L^2(\Gamma_N)} \|\mathbf{e}\|_{L^2(\Gamma_N)} \end{aligned} \quad (5.33)$$

In (5.33) we observe that the correction terms will be bounded if we obtain a bound of the L^2 -norm of the error in the displacement field. With the Aubin-Nitsche lemma [124], the L^2 -norm of the error in the displacement field can be bounded with the respective error in energy norm for all the problem domain, then:

$$\begin{aligned} \|\mathbf{e}\|_{L^2(\Omega)} &\leq C_{\Omega} h \|\mathbf{e}\|_{\Omega} \\ \|\mathbf{e}\|_{L^2(\Gamma_N)} &\leq C_{\Gamma} h^{\frac{1}{2}} \|\mathbf{e}\|_{\Omega} \end{aligned} \quad (5.34)$$

where h is a representative size of the mesh. In a h -uniform refinement process h is the size of the element. However for h -adaptive refinement processes we should take a representative value of the mesh size. In this case we have related it with the Number of Degrees of Freedom (NDoF), then have defined $h := \left(\frac{1}{NDoF}\right)^{\frac{1}{d}}$, where d is the dimension of the problem, $d = 2$ for 2D. C_{Ω} and C_{Γ} are constants dependent on the problem, but independent on the mesh size. Then we can bound the correction terms as follows:

$$\begin{aligned} \left| \int_{\Omega} (\mathbf{s}_{\sigma}^*)^T \mathbf{e} \, d\Omega \right| &\leq \|\mathbf{s}_{\sigma}^*\|_{L^2(\Omega)} \|\mathbf{e}\|_{L^2(\Omega)} \leq C_{\Omega} h \|\mathbf{s}_{\sigma}^*\|_{L^2(\Omega)} \|\mathbf{e}\|_{\Omega} \\ \left| \int_{\Gamma_N} (\mathbf{r}_{\sigma}^*)^T \mathbf{e} \, d\Gamma \right| &\leq \|\mathbf{r}_{\sigma}^*\|_{L^2(\Gamma_N)} \|\mathbf{e}\|_{L^2(\Gamma_N)} \leq C_{\Gamma} h^{\frac{1}{2}} \|\mathbf{r}_{\sigma}^*\|_{L^2(\Gamma_N)} \|\mathbf{e}\|_{\Omega} \end{aligned} \quad (5.35)$$

Note that correction terms in (5.32) can be taken as the residual of the recovered solution. Let us define the residual of the recovered stress field $\boldsymbol{\sigma}_{\sigma}^*$:

$$R^*(\mathbf{v}) := l(\mathbf{v}) - \bar{a}(\boldsymbol{\sigma}_{\sigma}^*, \boldsymbol{\sigma}(\mathbf{v})) \quad \forall \mathbf{v} \in V \quad (5.36)$$

this expression can be rewritten as follows:

$$\begin{aligned} R^*(\mathbf{v}) &= \sum_{K \in \mathcal{T}} \left(\int_{\Omega^K} \mathbf{b}^T \mathbf{v} \, d\Omega + \int_{\Gamma_N \cap \partial\Omega^K} \mathbf{t}^T \mathbf{v} \, d\Gamma \right) - \\ &\quad - \sum_{K \in \mathcal{T}} \int_{\Omega^K} (\boldsymbol{\sigma}_{\sigma}^*)^T \boldsymbol{\varepsilon}(\mathbf{v}) \, d\Omega \quad \forall \mathbf{v} \in V \end{aligned} \quad (5.37)$$

and integrating by parts:

$$\begin{aligned}
 R^*(\mathbf{v}) = & \sum_{K \in \mathcal{T}} \left(\int_{\Omega^K} \mathbf{b}^T \mathbf{v} \, d\Omega + \int_{\partial\Omega^K \cap \Gamma_N} \mathbf{t}^T \mathbf{v} \, d\Gamma \right) + \\
 & + \sum_{K \in \mathcal{T}} \left(\int_{\Omega^K} (\mathbf{L}^T \boldsymbol{\sigma}_\sigma^*)^T \mathbf{v} \, d\Omega - \int_{\partial\Omega^K} (\mathbf{G} \boldsymbol{\sigma}_\sigma^*)^T \mathbf{v} \, d\Gamma \right) \quad \forall \mathbf{v} \in V \quad (5.38)
 \end{aligned}$$

grouping terms and using the definition of $-\mathbf{s}_\sigma^* = \mathbf{L}^T \boldsymbol{\sigma}_\sigma^* + \mathbf{b}$ in K and $-\mathbf{r}_\sigma^* = \mathbf{t} - \mathbf{G} \boldsymbol{\sigma}_\sigma^*$ only over Γ_N due to the continuity of $\boldsymbol{\sigma}_\sigma^*$, we end up with the following expression:

$$R^*(\mathbf{v}) = - \sum_{K \in \mathcal{T}} \left(\int_{\Omega^K} (\mathbf{s}_\sigma^*)^T \mathbf{v} \, d\Omega + \int_{\partial\Omega^K \cap \Gamma_N} (\mathbf{r}_\sigma^*)^T \mathbf{v} \, d\Gamma \right) \quad \forall \mathbf{v} \in V \quad (5.39)$$

Adding the local contribution of all elements in the mesh and applying the Cauchy-Schwartz inequality we obtain an expression similar to the one that defines the upper bound of the correction terms.

$$|R^*(\mathbf{v})| \leq \|\mathbf{s}_\sigma^*\|_{L^2(\Omega)} \|\mathbf{v}\|_{L^2(\Omega)} + \|\mathbf{r}_\sigma^*\|_{L^2(\Gamma_N)} \|\mathbf{v}\|_{L^2(\Gamma_N)} \quad \forall \mathbf{v} \in V \quad (5.40)$$

Now, particularizing (5.40) for $\mathbf{v} = \mathbf{e}$ and applying the inequality defined in (5.35).

$$|R^*(\mathbf{e})| \leq C_\Omega h \|\mathbf{s}_\sigma^*\|_{L^2(\Omega)} \|\mathbf{e}\|_\Omega + C_\Gamma h^{\frac{1}{2}} \|\mathbf{r}_\sigma^*\|_{L^2(\Gamma_N)} \|\mathbf{e}\|_\Omega \quad (5.41)$$

Because of the use of the SPR-CD recovery technique, the lack of equilibrium along the boundary is negligible, thus we can use the assumption that $C_\Gamma = C_\Omega$ without loss in accuracy. Finally we square the contribution of the lacks of equilibrium in order to further decrease the importance of the lack of equilibrium along the boundary:

$$|R^*(\mathbf{e})| \leq C \|\mathbf{e}\|_\Omega \left\{ h^2 \|\mathbf{s}_\sigma^*\|_{L^2(\Omega)}^2 + h \|\mathbf{r}_\sigma^*\|_{L^2(\Gamma_N)}^2 \right\}^{\frac{1}{2}} \quad (5.42)$$

where $C = \sqrt{2} C_\Omega^1$. Now, we rewrite expression (3.86) in the following manner:

$$\begin{aligned}
 \|\mathbf{e}\|_\Omega^2 \leq & \int_{\Omega} \left(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h \right)^T \mathbf{D}^{-1} \left(\boldsymbol{\sigma}_\sigma^* - \boldsymbol{\sigma}^h \right) \, d\Omega + 2R^*(\mathbf{e}) = \\
 & \mathcal{E}_{\text{SPR-CD}}^2 + 2R^*(\mathbf{e})
 \end{aligned} \quad (5.43)$$

¹Note that $\sqrt{2}$ is obtained when the Cauchy-Schwarz inequality is applied to (5.41).
 $\sum_i^n a_i \leq (n^2 \sum_i^n a_i^2)^{\frac{1}{2}}$.

substituting (5.42) in (5.43) we obtain:

$$\begin{aligned} |||\mathbf{e}|||_{\Omega}^2 &\leq \mathcal{E}_{\text{SPR-CD}}^2 + 2C |||\mathbf{e}|||_{\Omega} \left\{ h^2 \|\mathbf{s}_{\sigma}^*\|_{L^2(\Omega)}^2 + h \|\mathbf{r}_{\sigma}^*\|_{L^2(\Gamma_N)}^2 \right\}^{\frac{1}{2}} \\ &= \mathcal{E}_{\text{SPR-CD}}^2 + \Xi |||\mathbf{e}|||_{\Omega} \end{aligned} \quad (5.44)$$

Expression (5.44) is a second order degree polynomial in $|||\mathbf{e}|||_{\Omega}$. The most conservative root is providing the upper bound in energy norm up to a constant C , see [73].

$$|||\mathbf{e}|||_{\Omega} \leq \mathcal{E}_{\text{FUBSPR-CD}} := \frac{\Xi + \sqrt{\Xi^2 + 4\mathcal{E}_{\text{SPR-CD}}^2}}{2} \quad (5.45)$$

Now we have to investigate the convergence rate of each term in (5.44).

- $\mathcal{E}_{\text{SPR-CD}}$: this term is considered *asymptotically exact* because the recovery technique has a higher convergence rate than the FE solution [67]. Thus its convergence rate could be considered the same as that of the convergence rate of the error in energy norm, p , being p the order of the FE interpolation.
- $|||\mathbf{e}|||_{\Omega}$: obviously the convergence rate of this term is p .
- Ξ : the convergence rate of this term is not obvious. In our case, it will depend on the convergence rate of $\|\mathbf{s}_{\sigma}^*\|_{L^2(\Omega)}$ and $\|\mathbf{r}_{\sigma}^*\|_{L^2(\Gamma_N)}$. Assuming that the recovered field has a convergence rate $p + q$, $q > 0$, higher than that for the FE solution, p , then the convergence of $\|\mathbf{s}_{\sigma}^*\|_{L^2(\Omega)}$ would be $p + q - 1$ and the convergence rate of $\|\mathbf{r}_{\sigma}^*\|_{L^2(\Gamma_N)}$ would be $p + q - \frac{1}{2}$. Then, under this situation the convergence rate of Ξ could be considered as $p + q$.

In general we cannot confirm that $q = 1$. However we know that the recovered field converges faster than the FE solution, $q > 0$. This means that these correction terms will disappear during the refinement process. Therefore, the

plain ZZ error estimator with the SPR-CD technique will provide asymptotically guaranteed upper error bounds.

Finally, we can conclude that if the recovered field converges faster than the FE solution, we will obtain a stable upper bound, that is *asymptotically effective*. Regarding to the constant C , some authors take the value $C = 1$ for error estimation [62]. However this approach could be in some cases inaccurate and in others it will provide under estimations of the true error. In the following Section we propose a methodology to numerically compute this constant for each problem.

5.3.1 Numerical evaluation of the constant C_Ω

Expression (5.45) is an upper bound of the error in energy norm, but requires the evaluation of a constant C which is specific for each problem and also for each discretization type. This constant appears in the *a priori* error estimator (5.34). This constant relates the L^2 -norm of the error in displacements with the respective error in energy norm.

Prof. Stein's group at Leibniz University is actively working in explicit residual-type error estimators for the elasticity problem [62, 125]. These error estimators also requires the evaluation of a constant with similar characteristics than the constant presented in (5.34), but arising from the Korn inequality. Recently, Prof. Stein presented a value for the constant they use in their explicit residual type error estimator [63] only valid for linear triangular elements.

In this work we propose a methodology to numerically estimate the value of the constant for any mesh type and also for any problem domain. This methodology is based on the numerical evaluation of the constant C_Ω in (5.34) under a h -adaptive refinement process. Let H and h be the representative sizes of two meshes such that $H \ll h$, \mathbf{u}^H be the FE solution of the finer mesh. We can consider that \mathbf{u}^H is a good approximation to \mathbf{u} in comparison with \mathbf{u}^h .

Considering the Richardson extrapolation the following relations will hold.

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}^h\|_{L^2(\Omega)}^2 &\approx \frac{\|\mathbf{u}^H - \mathbf{u}^h\|_{L^2(\Omega)}^2}{1 + \left(\frac{H}{h}\right)^{2p+2}} \\ \|\|\mathbf{u} - \mathbf{u}^h\|\|_{\Omega}^2 &\approx \frac{\|\|\mathbf{u}^H - \mathbf{u}^h\|\|_{\Omega}^2}{1 + \left(\frac{H}{h}\right)^{2p}} \end{aligned} \quad (5.46)$$

where p is the degree of the FE solution. Under these assumptions, it immediately follows the evaluation of the numerical approximation to C_{Ω} , C_{Ω}^* for the mesh h .

$$C_{\Omega} \approx C_{\Omega}^* = \sqrt{\frac{\|\mathbf{u}^H - \mathbf{u}^h\|_{L^2(\Omega)}^2 \left(1 + \left(\frac{H}{h}\right)^{2p}\right)}{h^2 \|\|\mathbf{u}^H - \mathbf{u}^h\|\|_{\Omega}^2 \left(1 + \left(\frac{H}{h}\right)^{2p+2}\right)}} \quad (5.47)$$

Then constant C will be also approximated by $C^* = \sqrt{2}C_{\Omega}^*$. Note that we are also assuming that $C_{\Gamma} \approx C_{\Omega}^*$. In this case this is reasonable because the SPR-CD technique provides a recovered stress field with a negligible lack of boundary equilibrium, $\|\mathbf{r}_{\sigma}^*\|_{L^2(\Gamma_N)} \ll \|\mathbf{s}_{\sigma}^*\|_{L^2(\Omega)}$, as indicated before. For other recovery processes, where the boundary equilibrium will not be fulfilled, such as the plain SPR, this assumption does not hold.

Finally the upper bound of the error in energy norm reads:

$$\|\|\mathbf{e}\|\|_{\Omega} \lesssim \mathcal{E}_{\text{FUBSPR-CD}}^* := \frac{\Xi^* + \sqrt{(\Xi^*)^2 + 4\mathcal{E}_{\text{SPR-CD}}^2}}{2} \quad (5.48)$$

where:

$$\Xi^* = 2C^* \left\{ h^2 \|\mathbf{s}_{\sigma}^*\|_{L^2(\Omega)}^2 + h \|\mathbf{r}_{\sigma}^*\|_{L^2(\Gamma)}^2 \right\}^{\frac{1}{2}} \quad (5.49)$$

Note that with the described process, a estimation of the constant is not available for the first iteration. For the evaluation of the constant C_{Ω}^* for mesh n , $n > 2$, we take, in general a difference of two meshes, $u^H = u^n$ and $u^h = u^{n-2}$. But, for the first one ($n = 1$) we set $C_{\Omega}^* = 1$ and for the second one ($n = 2$) we take $\mathbf{u}^h = u^1$ and u^H as the solution of the current mesh.

5.4 Numerical results

In this Section we compare the three bounding techniques presented before. The upper error bound for the FUB technique is obtained with expression (5.45), whereas the FER and the LM techniques evaluate the upper error bound directly using the ZZ error estimator:

$$\|e\|_{\Omega}^2 \leq \mathcal{E}_{\text{FER}}^2 = \int_{\Omega} \left(\boldsymbol{\sigma}_{\text{FE}}^* - \boldsymbol{\sigma}^h \right)^T \mathbf{D}^{-1} \left(\boldsymbol{\sigma}_{\text{FE}}^* - \boldsymbol{\sigma}^h \right) d\Omega \quad (5.50)$$

$$\|e\|_{\Omega}^2 \leq \mathcal{E}_{\text{RL}}^2 = \int_{\Omega} \left(\boldsymbol{\sigma}_{\text{RL}}^* - \boldsymbol{\sigma}^h \right)^T \mathbf{D}^{-1} \left(\boldsymbol{\sigma}_{\text{RL}}^* - \boldsymbol{\sigma}^h \right) d\Omega \quad (5.51)$$

Recall that the fields $\boldsymbol{\sigma}_{\text{FE}}^*$ and $\boldsymbol{\sigma}_{\text{RL}}^*$ are statically admissible. Thus the ZZ error estimator directly provides upper error bounds as explained in Section 3.4.

In this Section we will compare the results obtained with these three techniques in terms of accuracy and performance. In all problems, plane strain and bi-quadratic (Q8) elements will be considered for all analysis. In all problems a 2×2 square portion of the domain has been selected and the corresponding Neumann boundary conditions applied. Note that 2×2 square domains have been used in order to force that the nodes of the mesh coincides with the sides of the domain. The general case in which the boundary of the domain intersects the mesh is under development.

5.4.1 Problem 1a: 3rd order solution in a 2×2 square

This problem has an analytical cubic solution in displacements with body forces. The problem model, material properties and exact solution are represented in Figure 3.11. Table 5.2 shows the global effectivity index obtained with the different upper error bounding techniques presented in this Section. The results obtained with the SPR-CD are also shown for comparison. We also observe that there are two columns referring to the FUB technique, each

one with a different constant C as indicated. The results indicate that all the bounding techniques are providing upper error bounds. The most accurate one is the RL procedure while the $\text{FUB}_{\text{SPR-CD}}^{C=1}$ is obtaining the worst results, in general. However we should recall that this last technique is constant dependent, and as observed, as the value of the constant changes, the results also change. When we estimate the value of the constant as described in Section 5.3.1, the results improve considerably. Table 5.2 also shows the value of the constant C numerically evaluated as described in Section 5.3.1. We clearly observe that the numerical approximation of C rapidly stabilizes.

Table 5.2: Problem 1a. Q8 uniform refinement. Values of the global effectivity index θ for different bounding techniques and C^* for the FUB technique.

DoF	SPR-CD	C^*	$\text{FUB}_{\text{SPR-CD}}^{C^*}$	$\text{FUB}_{\text{SPR-CD}}^{C=1}$	FER	RL
130	1.0031653	1.0000000	3.5674100	3.5674100	1.1408109	1.0056877
450	1.0051082	0.0244572	1.0317116	2.5444876	1.1726613	1.0035760
1,666	1.0061665	0.0244476	1.0243175	1.9824101	1.2015884	1.0020300
6,402	1.0067316	0.0244476	1.0196603	1.6611042	1.1948981	1.0011346
25,090	1.0070231	0.0244476	1.0166475	1.4723679	1.2003460	1.0006565

In Figure 5.3, the left hand side graph is a plot of the results shown in Table 5.2 for the different bounding techniques. On the right hand side, we plot the computational cost of those techniques where the most remarkable fact is that the $\text{FUB}_{\text{SPR-CD}}$ is almost inexpensive in comparison with the plain SPR-CD technique as it only requires an additional cheap post process.

5.4.2 Problem 1c. 2×2 square without body forces

In this case we present a third load set for the problem 1, see Section 3.5.1. This problem has an analytical cubic solution in displacements with null body forces. The problem model, material properties and exact solution are represented in Figure 5.4.

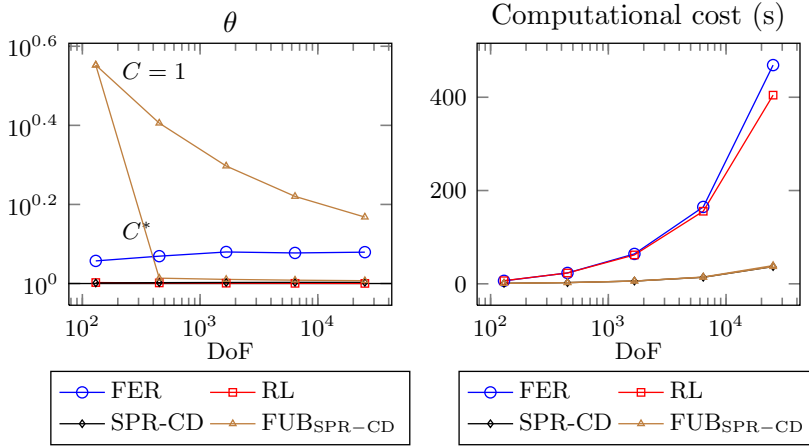


Figure 5.3: Problem 1a. Q8 uniform refinement. Global effectivity index θ and overall computational cost.

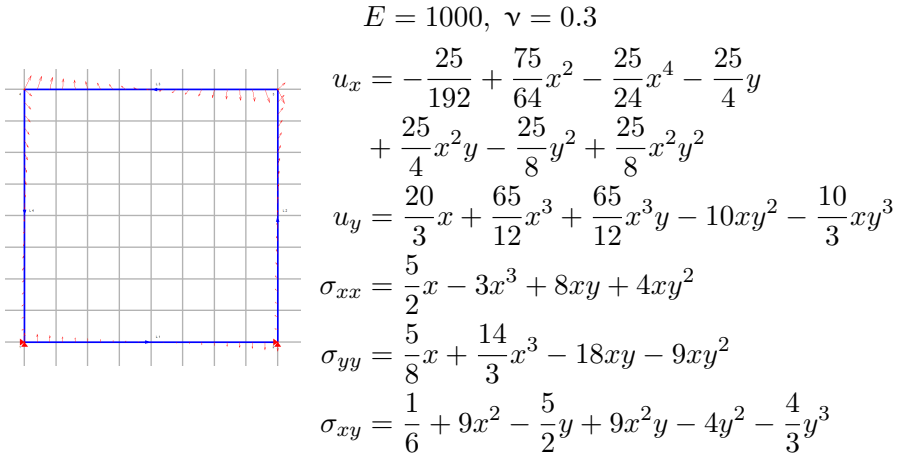


Figure 5.4: Problem 1c. Model, material and analytical solution.

As in the previous example, Table 5.3 presents the results obtained for the different bounding techniques. These results show the same tendencies for the bounding techniques than in the previous example. The main difference is that the effectivity index for the plain SPR-CD goes below 1. This is normal

because the SPR-CD technique does not guarantee upper bounds. Despite of this the rest of techniques are able to provide the upper bounds. Figure 5.5 represents the results presented in the Table 5.3. The effectivity index for the FUB_{SPR-CD} technique goes smoothly towards $\theta = 1$ in both situations, when $C = 1$ and also for C^* . The FER seems not to be converging to one, losing the asymptotically exactness of the SPR-based technique, while the RL seems to be converging. Regarding to the computational cost, we have obtained the same results as before, as expected.

Table 5.3: Problem 1c. Q8 uniform refinement. Values of the global effectivity index θ for different bounding techniques and C^* for the FUB technique.

DoF	SPR-CD	C^*	$FUB_{SPR-CD}^{C^*}$	$FUB_{SPR-CD}^{C=1}$	FER	RL
130	1.0220357	1.0000000	1.2404561	1.2467417	1.0317717	1.0205229
450	0.9987000	1.4910874	1.1681756	1.1728618	1.0498760	1.0084493
1,666	0.9935364	1.5482592	1.0936909	1.0969724	1.0825020	1.0039569
6,402	0.9928568	1.3584752	1.0511103	1.0530188	1.0677168	1.0019774
25,090	0.9930377	1.2549370	1.0353162	1.0364043	1.0708363	1.0010347

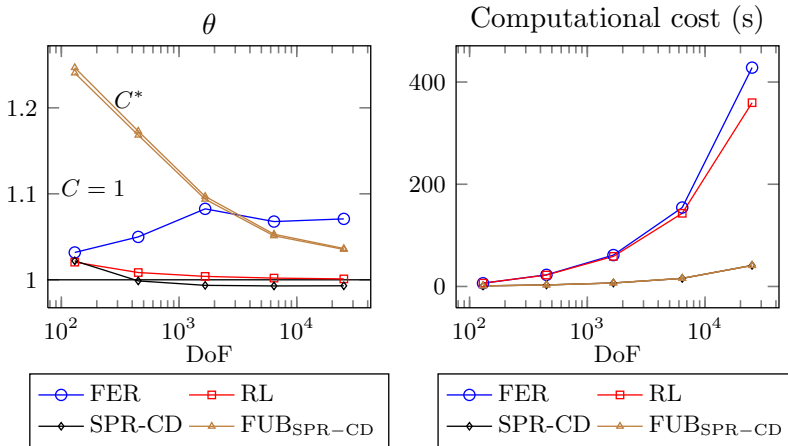


Figure 5.5: Problem 1c. Q8 uniform refinement. Global effectivity index θ and overall computational cost.

Finally, Figure 5.6 shows the behavior of the FUB_{SPR-CD} technique for different values of the constant C defined by the user. The value of the constant does not affect to the asymptotic convergence of the technique to $\theta = 1$. It

affects to the accuracy of the estimate and also to the upper bound property as very low values of C could lead to underestimations of the true error. For instance, blue and red lines differ on the accuracy while maintaining the upper bound property, however black line for $C = 0.1$ lose the upper bound property. We should recall that the implemented method to numerically evaluate C^* is always providing upper error bounds.

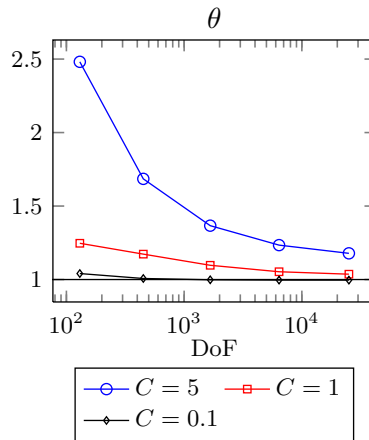


Figure 5.6: Problem 1c. Q8 uniform refinement. Global effectivity index θ for the $\text{FUB}_{\text{SPR-CD}}$ with different values of the constant C . Note that for $C = 0.1$ for the finer mesh the effectivity is $\theta = 0.99595488728704$.

5.4.3 Problem 1d. 2×2 square. Thick-wall cylinder subjected to internal pressure

This problem corresponds to the geometry of the problem 1 with the load set corresponding to a thick-wall cylinder under internal pressure. The analytical solution corresponds to problem 2, see Section 3.5.1. We have extracted from the domain of the problem 2 a 2×2 area (green area in Figure 5.7) for the analysis. This square area is the problem domain for the FE analysis where we apply the corresponding Neumann boundary conditions. We have also

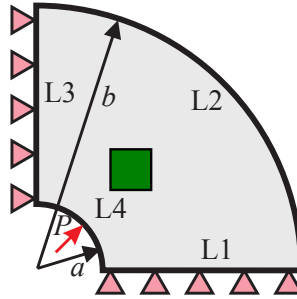


Figure 5.7: Problem 1d: The model of the problem corresponds to the green square area extracted from the thick-wall cylinder geometry which corresponds to the geometry of the problem 2.

constrained the rigid body motions. The problem model is represented in Figure 5.7.

Table 5.4 presents the global effectivity index for the different error estimators. As in the previous occasions, the bounding techniques provides upper error bounds. For this particular case, the SPR-CD technique is also providing over estimations of the true error in energy norm. Continuing with the format used in previous problems, in Figure 5.8 we plot the effectivity evolution for the techniques analyzed. We observe the smooth convergence of the FUB_{SPR-CD} technique in both situations, when $C = 1$ and for C^* . In this case it is easier to observe that the FER, obtaining an accurate upper error bound, is not converging to one but it stabilizes to a higher value. It is worthy to mention that, as in the previous cases, the accuracy obtained with the RL technique is remarkable.

5.5 Conclusions

In this Chapter we have presented some novel techniques that are able to provide upper error bounds in energy norm. The FUB_{SPR-CD} and the FER

Table 5.4: Problem 1d. Q8 uniform refinement. Values of the global effectivity index θ for different bounding techniques and C^* for the FUB technique.

DoF	SPR-CD	C^*	$\text{FUB}_{\text{SPR-CD}}^{C^*}$	$\text{FUB}_{\text{SPR-CD}}^{C=1}$	FER	RL
130	1.0029377	1.0000000	1.1116927	1.1116927	1.0931193	1.0062968
450	1.0042645	1.1974190	1.0737540	1.0619881	1.1185186	1.0030439
1,666	1.0049761	1.2602164	1.0464060	1.0377170	1.1355020	1.0015428
6,402	1.0054346	1.0741445	1.0271042	1.0255938	1.1379648	1.0008145
25,090	1.0057159	0.9800008	1.0191599	1.0194361	1.1409429	1.0004522

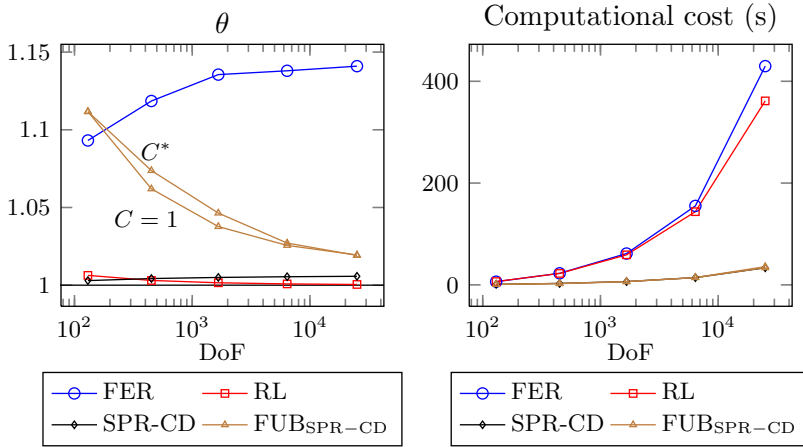


Figure 5.8: Problem 1d. Q8 uniform refinement. Global effectivity index θ and overall computational cost.

techniques are based on recovery procedures, while the RL also includes solving local Neumann problems.

The $\text{FUB}_{\text{SPR-CD}}$ is a SPR-based upper error bounding method for the linear elasticity problem. This method consists of two steps, the first one uses a very accurate and locally equilibrated recovery process, the SPR-CD. It usually provides non-guaranteed upper error bounds. In order to guarantee the upper bound property some correction terms introduced in [73] have to be taken into consideration. The second step consists in evaluating upper bounds of the correction terms. To bound these terms we use a constant dependent a

priori error estimator. Additionally we also propose a method to numerically evaluate the constant. The numerical results show the accuracy of the proposed technique, and also its convergence from above to the exact solution. The procedure for the evaluation of C^* has also shown to be effective.

For the FER technique the results showed that the computational cost to obtain the recovered field is quite high in comparison with standard recovery procedures or the other bounding techniques, specially when compared to $FUB_{\text{SPR-CD}}$. The main advantage of the FER technique is that, in contrast with other recovery procedures such as the SPR-CD, it is able to provide guaranteed error bounds without any correction terms. Nevertheless, the SPR-CD has provided for these examples numerical upper bounds and error estimates close to one. We are currently working to improve the computational cost associated to the FER technique to make it competitive with traditional error bounding techniques.

Finally the RL consist of two steps, the first one is to build a weakly equilibrated traction field at each element by means of an SPR-based recovery procedure, and the second one is to solve a local problem. The results show the high accuracy of the method, however we have also obtained a high computational cost.

The $FUB_{\text{SPR-CD}}$, even when we need to evaluate a constant, is providing the best results if we take into account both accuracy and efficiency. The $FUB_{\text{SPR-CD}}$ is therefore a very competitive technique to provide upper error bounds in energy norm for practical applications. In the future we pretend to extend this method to GOA as indicated in Chapter 4.

Chapter 6

Error estimation in the recovered solution field

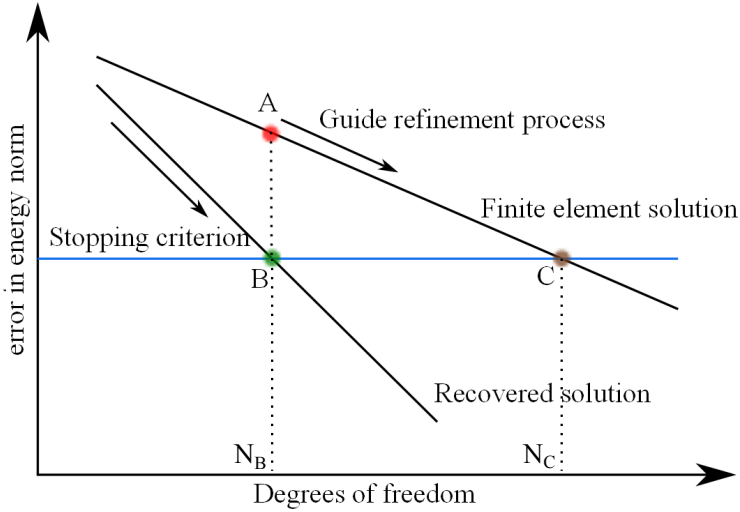
6.1 Introduction

For the recovery-based error estimators we usually compute an enhanced stress solution which is compared with the raw FE stress solution. As explained in Chapter 3 the enhanced solution has better properties, in terms of continuity and equilibrium. Since we have evaluated an enhanced solution, obviously we are interested in using it. However, in order to use it as the output of our FE code we need a method to assess the quality of this magnitude, i. e., we need to obtain an error estimate for the enhanced solution. Some authors [111] evaluate the error of the plain SPR recovered field by comparing it with some enhancements of the SPR solution, such as adding equilibrium and information of the elasticity problem. Other authors [126] also use a recovered (recycled) solution from the recovered solution.

This Chapter will show a technique to estimate the error, for the enhanced SPR-CD technique presented in Section 3.3. Recovery error estimation techniques are based on the assumption that the recovered solution is more accurate than the FE solution. A sufficiently accurate estimation of the error in energy norm of the recovered solution could lead to refinement processes based on the accuracy of the recovered solution instead of that of the FE solution. This could result in a considerable reduction of the computational cost that would be particularly interesting, for example, in optimization processes whose efficiency would be significantly increased as the required level of accuracy would be achieved with a considerable lower number of degrees of freedom.

Figure 6.1 shows an scheme of the evolution of the exact errors in energy norm of the finite element and recovered solutions. Note that in the SPR-CD technique 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 reach a prescribed accuracy level defined by blue horizontal line in Figure 6.1 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 so that the refinement process finishes when the error in the recovered solution is smaller than the prescribed value.

In this Chapter we will first show the initial developments that finally led to an heuristic expression that can be used to efficiently estimate the error in energy norm of the recovered solution σ_σ^* , initially evaluated to estimate the error of the FE solution \mathbf{u}^h . We also show some numerical results accompanied with the proposed h -adaptive technique. The results show a good accuracy in the error estimation and also an important reduction in computational cost in comparison with standard h -adaptive refinement procedures based on the error in energy norm of the FE solution.

Figure 6.1: h -adaptive refinement scheme.

6.2 Error norm representation for the recovered solution

The purpose of this Section is to introduce an error measure for the SPR-CD stress recovered field. We have developed an expression which allows us to evaluate the error of the nearly-statically recovered stress field σ_σ^* .

The SPR-CD procedure yields a post-processed stress field σ_σ^* which is taken as an enhanced approximation to the exact stresses, σ , more accurate than σ^h . As explained in Section 3.3, the recovered stress σ_σ^* is continuous but it has a lack of internal equilibrium \mathbf{s}_σ^* and also a lack of boundary equilibrium \mathbf{r}_σ^* .

Theorem 6.1. *Under the assumptions presented so far, the following expression evaluates the error in energy norm of the recovered stress field.*

$$\begin{aligned} ||| \mathbf{e}_\sigma^* |||_\Omega^2 &= ||| \mathbf{u} - \mathbf{u}_\sigma^* |||_\Omega^2 = \bar{a}(\boldsymbol{\sigma} - \boldsymbol{\sigma}_\sigma^*, \boldsymbol{\sigma} - \boldsymbol{\sigma}_\sigma^*) \\ &= - \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{e}_\sigma^* \, d\Omega - \int_{\Gamma_N} (\mathbf{r}_\sigma^*)^T \mathbf{e}_\sigma^* \, d\Gamma \end{aligned} \quad (6.1)$$

being \mathbf{e}_σ^* the error in displacements corresponding to the recovered solution $\boldsymbol{\sigma}_\sigma^*$ such that $\boldsymbol{\sigma}_\sigma^* = \boldsymbol{\sigma}(\mathbf{u}_\sigma^*) = \boldsymbol{\sigma}(\mathbf{u} - \mathbf{e}_\sigma^*)$. Note that \mathbf{u}_σ^* and \mathbf{e}_σ^* will not be explicitly evaluated.

Proof.

$$\begin{aligned} & - \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{e}_\sigma^* \, d\Omega - \int_{\Gamma_N} (\mathbf{r}_\sigma^*)^T \mathbf{e}_\sigma^* \, d\Gamma \\ &= \int_\Omega (\mathbf{L}^T \boldsymbol{\sigma}_\sigma^* + \mathbf{b})^T \mathbf{e}_\sigma^* \, d\Omega - \int_{\Gamma_N} (\mathbf{G} \boldsymbol{\sigma}_\sigma^* - \mathbf{t})^T \mathbf{e}_\sigma^* \, d\Gamma \\ &= \int_\Omega (\mathbf{L}^T \boldsymbol{\sigma}(\mathbf{u}_\sigma^*) - \mathbf{L}^T \boldsymbol{\sigma}(\mathbf{u}))^T \mathbf{e}_\sigma^* \, d\Omega - \int_{\Gamma_N} (\mathbf{G} \boldsymbol{\sigma}(\mathbf{u}_\sigma^*) - \mathbf{G} \boldsymbol{\sigma}(\mathbf{u}))^T \mathbf{e}_\sigma^* \, d\Gamma \\ &= \int_\Omega (-\mathbf{L}^T \boldsymbol{\sigma}(\mathbf{e}_\sigma^*))^T \mathbf{e}_\sigma^* \, d\Omega + \int_{\Gamma_N} (\mathbf{G} \boldsymbol{\sigma}(\mathbf{e}_\sigma^*))^T \mathbf{e}_\sigma^* \, d\Gamma \\ &= \int_\Omega \boldsymbol{\sigma}(\mathbf{e}_\sigma^*)^T \boldsymbol{\varepsilon}(\mathbf{e}_\sigma^*) \, d\Omega - \int_{\Gamma_N} (\mathbf{G} \boldsymbol{\sigma}(\mathbf{e}_\sigma^*))^T \mathbf{e}_\sigma^* \, d\Gamma + \int_{\Gamma_N} (\mathbf{G} \boldsymbol{\sigma}(\mathbf{e}_\sigma^*))^T \mathbf{e}_\sigma^* \, d\Gamma \\ &= \int_\Omega \boldsymbol{\sigma}(\mathbf{e}_\sigma^*)^T \boldsymbol{\varepsilon}(\mathbf{e}_\sigma^*) \, d\Omega \\ &= ||| \mathbf{e}_\sigma^* |||_\Omega^2 \end{aligned} \quad (6.2)$$

□

In the following we will show how to evaluate an upper error bound of $||| \mathbf{e}_\sigma^* |||_\Omega^2$ and also several heuristic error estimators making use of the recovered solution already evaluated with the SPR-CD technique. Regarding the evaluation of an upper bound, operating with (6.1) considering Cauchy-Schwartz inequality

we will have:

$$\begin{aligned} \|\mathbf{e}_\sigma^*\|_\Omega^2 &= - \int_\Omega (\mathbf{s}_\sigma^*)^T \mathbf{e}_\sigma^* \, d\Omega - \int_{\Gamma_N} (\mathbf{r}_\sigma^*)^T \mathbf{e}_\sigma^* \, d\Gamma \\ &\leq (\|\mathbf{s}_\sigma^*\|_{L^2(\Omega)} + \|\mathbf{r}_\sigma^*\|_{L^2(\Omega)}) \|\mathbf{e}_\sigma^*\|_{L^2(\Omega)} \end{aligned} \quad (6.3)$$

During the SPR-CD recovery process, the functional (3.46) tries to minimize $\|\mathbf{u}_u^* - \mathbf{u}^h\|_{L^2(\mathcal{P}^i)}$, locally. In addition, as \mathbf{u}_σ^* would be a recovered displacements field, laying in a so-called 'broken space' richer than V^h , we could assume that the L^2 -norm of the error of the recovered solution is smaller than the L^2 norm of the error of the FE solution:

$$\|\mathbf{e}_\sigma^*\|_{L^2(\Omega)} \lesssim \|\mathbf{e}\|_{L^2(\Omega)} \quad (6.4)$$

Considering the assumption in (6.4) in (6.3) we would obtain an upper error bound of the recovered solution:

$$\|\mathbf{e}_\sigma^*\|_\Omega \lesssim (\|\mathbf{s}_\sigma^*\|_{L^2(\Omega)} + \|\mathbf{r}_\sigma^*\|_{L^2(\Omega)}) \|\mathbf{e}\|_{L^2(\Omega)} \quad (6.5)$$

Note that $\|\mathbf{e}\|_{L^2(\Omega)}$ is unknown in general. One possibility is to replace it by $\|\mathbf{e}_u\|_{L^2(\Omega)}$, recall that $\mathbf{e}_u = \mathbf{u}_u^* - \mathbf{u}^h$ obtaining a computational version of the upper bound. The implicit idea is to replace $\|\mathbf{e}_\sigma^*\|_{L^2(\Omega)}$ by $\|\mathbf{e}_u\|_{L^2(\Omega)}$, *i.e.* we have replaced the error in the recovered solution \mathbf{u}_σ^* by the estimated error of the FE solution \mathbf{u}^h , to obtain a bound of the error in the recovered solution.

Following this idea we can also derive expressions for the error estimator. We can replace $\mathbf{e}_\sigma^* = \mathbf{u} - \mathbf{u}_\sigma^*$ by $\mathbf{e}_u = \mathbf{u}_u^* - \mathbf{u}^h$ in (6.1), and we define the following error estimator \mathcal{E}_1^* in (6.6a) to check if it could provide an indication of the error level in energy norm of the recovered solution σ_σ^* . We also defined the error indicators \mathcal{E}_2^* and \mathcal{E}_3^* as described in (6.6b) and (6.6c) to force the result

to be positive:

$$\mathcal{E}_1^* = \sqrt{-\int_{\Omega} (\mathbf{s}_{\sigma}^*)^T \mathbf{e}_u \, d\Omega - \int_{\Gamma} (\mathbf{r}_{\sigma}^*)^T \mathbf{e}_u \, d\Gamma} \quad (6.6a)$$

$$\mathcal{E}_2^* = \sqrt{\sum_K \left(\left| \int_K (\mathbf{s}_{\sigma}^*)^T \mathbf{e}_u \, d\Omega \right| + \left| \int_{\partial K \cup \Gamma_N} (\mathbf{r}_{\sigma}^*)^T \mathbf{e}_u \, d\Gamma \right| \right)} \quad (6.6b)$$

$$\mathcal{E}_3^* = \sqrt{\int_{\Omega} |(\mathbf{s}_{\sigma}^*)^T \mathbf{e}_u| \, d\Omega + \int_{\Gamma} |(\mathbf{r}_{\sigma}^*)^T \mathbf{e}_u| \, d\Gamma} \quad (6.6c)$$

In (6.6b) the value of the integrals at each element are forced to be positive. In (6.6c) the integrands themselves are forced to be positive. Note that this is a reasonable assumption. For example, if we assume $\mathbf{r}_{\sigma}^* = 0$, see (6.2).

$$0 \leq \boldsymbol{\sigma}(\mathbf{e}_{\sigma}^*) \boldsymbol{\varepsilon}(\mathbf{e}_{\sigma}^*) = -(\mathbf{s}_{\sigma}^*)^T \mathbf{e}_{\sigma}^* = |(\mathbf{s}_{\sigma}^*)^T \mathbf{e}_{\sigma}^*| \quad (6.7)$$

As \mathbf{s}_{σ}^* and \mathbf{e}_{σ}^* are consistent (\mathbf{s}_{σ}^* would be the defaults of equilibrium corresponding to \mathbf{u}_{σ}^* , whose associated error is \mathbf{e}_{σ}^*) then $0 \leq -(\mathbf{s}_{\sigma}^*)^T \mathbf{e}_{\sigma}^*$. However, in (6.6c) \mathbf{e}_{σ}^* has been substituted by \mathbf{e}_u . The terms \mathbf{s}_{σ}^* and \mathbf{e}_u are non-consistent and as a result $-(\mathbf{s}_{\sigma}^*)^T \mathbf{e}_u$ could be negative. This suggests the use of the approximation in (6.6c), $-(\mathbf{s}_{\sigma}^*)^T \mathbf{e}_{\sigma}^* \approx |(\mathbf{s}_{\sigma}^*)^T \mathbf{e}_u|$.

6.3 h -adaptive refinement process

In the previous Section we have presented several methods to estimate the error of the recovered solution. The numerical results will indicate that the estimator \mathcal{E}_3^* provides very accurate results with an excellent global effectivity index. Now, we are going to show how to use this error estimator of the recovered solution to define a h -adaptive refinement processes.

During a h -adaptive refinement process, using a standard FE compilation, the process reads as follows:

1. Generate a FE mesh.
2. Solve the FE problem.
3. Estimate the error of the FE solution (locally and globally).
4. If target error is smaller than the estimated error of the FE solution continue to step 5, else stop the process.
5. Generate a h -adapted mesh using the local FE error estimation
6. Go to step 2.

In this classical situation we are estimating the error of the raw FE solution, then we are using the raw FE solution $(\mathbf{u}^h, \boldsymbol{\sigma}^h)$ as output. However, when we use our recovery procedure we have an improved solution $(\mathbf{u}_u^*, \boldsymbol{\sigma}_\sigma^*)$ available. So far, we were unable to estimate the error of this last solution, therefore this output was not reliable. However, with the contribution presented in this Chapter we have a methodology to obtain an accurate estimation of the error of this recovered solution, \mathcal{E}_3^* , thus we can use $(\mathbf{u}_u^*, \boldsymbol{\sigma}_\sigma^*)$ as the output for the analysis. The information about the error estimation in the recovered solution could then be used in the h -adaptive refinement process to obtain a solution $(\mathbf{u}_u^*, \boldsymbol{\sigma}_\sigma^*)$ with the required accuracy.

The h -adaptive procedure proposed in this Section is guided by the well-established techniques based on the error estimation of the FE solution. As a consequence of the refinement process the error of the FE solution and the error of the recovered solution will simultaneously decrease. An scheme of the proposed h -adaptive process is in Figure 6.1. The main difference between the traditional refinement process and the proposed one is, simply, the stopping criterion. We simply propose to stop the h -adaptive refinement process when the estimated error of the recovered solution is smaller than the target error. As the recovered solution reaches the prescribed error level with less degrees of freedom than the FE solution, this method produces important savings in the total computational cost of the analysis. The process would then be as follows:

1. Generate a FE mesh.
2. Solve the FE problem.
3. Evaluate the local error estimate of FE solution $(\mathbf{u}^h, \boldsymbol{\sigma}^h)$.
4. Evaluate the global value of the error estimate of the recovered solution $(\mathbf{u}_u^*, \boldsymbol{\sigma}_\sigma^*)$ (costless procedure).
5. If target error is smaller than the error of the recovered solution continue to step 6, else stop the process.
6. Generate a h -adapted mesh using the local FE error estimation
7. Go to step 2.

6.4 Numerical Examples

This section will in the first place show the accuracy of the error estimator for the recovered solution, both at global and local levels. After that we will present the results obtained with the new h -adaptive process previously described.

6.4.1 Accuracy of the error estimator for the recovered solution

In this Section we have two objectives regarding the error estimation of the recovered stress field. First we will check if the error indicators \mathcal{E}_i^* 's provide a rough idea about the error level of the recovered solution. Then we will also check if the local evaluations, at element level, of \mathcal{E}_i^* 's could roughly describe the distribution of the error of the recovered solution. As far as we know, there is no any strict mathematical relation between \mathcal{E}^* and the error indicators

\mathcal{E}_i^* that could make them provide accurate evaluations of \mathcal{E}^* . However, the numerical results obtained show that the \mathcal{E}_i^* 's capture the order of magnitude of \mathcal{E}^* . In particular, \mathcal{E}_3^* provides an accurate evaluation of \mathcal{E}^* ($\mathcal{E}_3^* \approx \mathcal{E}^*$) and very similar error distributions.

Finally we will also introduce the h -adaptive refinement process based on the error estimation of the recovered solution. The results will show the importance of this new process in computational cost saving while maintaining the accuracy of the solution.

Problem 1b: 2×2 plate with 4^{rd} order solution

This problem considers a 4^{th} order displacements field over an infinite domain. A 2×2 square portion has been modeled. The corresponding body loads and Neumann conditions have been imposed. For further description of the problem see Section 3.5.1.

Figure 6.2 shows the evolution of the global effectivity index θ considering the error estimates \mathcal{E}_i^* defined in (6.6). Note that in this case we have considered the lack of equilibrium both, in the internal and boundary equilibrium equations.

The results show that the error estimators \mathcal{E}_1^* , \mathcal{E}_2^* and \mathcal{E}_3^* capture the order of magnitude of the exact error in energy norm of the recovered solution. In particular, the results obtained with \mathcal{E}_3^* provide error effectivity indexes very close to one, the desired value.

Moreover, in Figure 6.3 we show the local error evaluated by $\mathcal{E}_3^*|_K$ and the exact error of the recovered field $|||\mathbf{e}_\sigma^*|||_K$ for a sequence of h -adapted meshes. Index K indicates that the quantities are evaluated at element level. For this

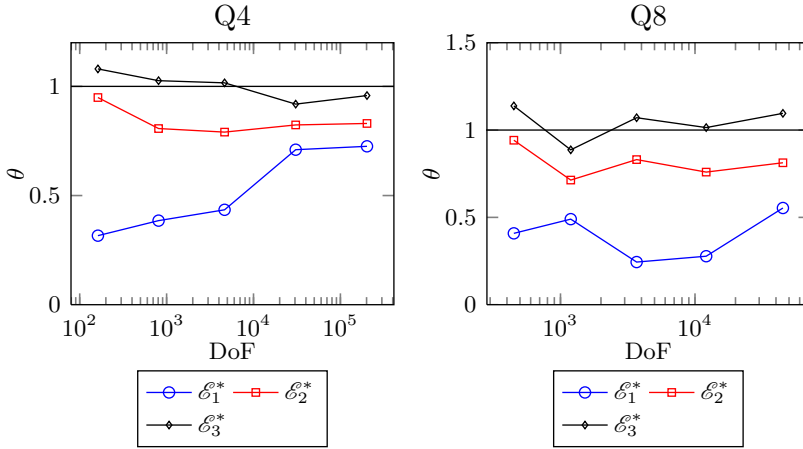


Figure 6.2: Problem 1b. Global effectivity index during the h -adaptive refinement process.

problem both results are quite similar, thus $\mathcal{E}_3^*|_K$ is a good indicator at local level of the error of the recovered solution.

Problem 2: Thick-wall cylinder

Let us now consider a pipe under internal pressure. For further description of the problem see Section 3.5.1. Figure 6.4 shows the evolution of the effectivity index for the different error estimates.

For this problem we have also evaluated the distribution of the exact energy norm of the recovered solution $|||e^*|||_K$ and the distribution of the error estimate $\mathcal{E}_3^*|_K$. In Figure 6.5 we have represented these results for a sequence of h -adapted meshes. We can observe that both error distributions are quite similar. These results show that the error estimator for the recovered solutions \mathcal{E}_3^* has a good behavior at global level, but also at local level.

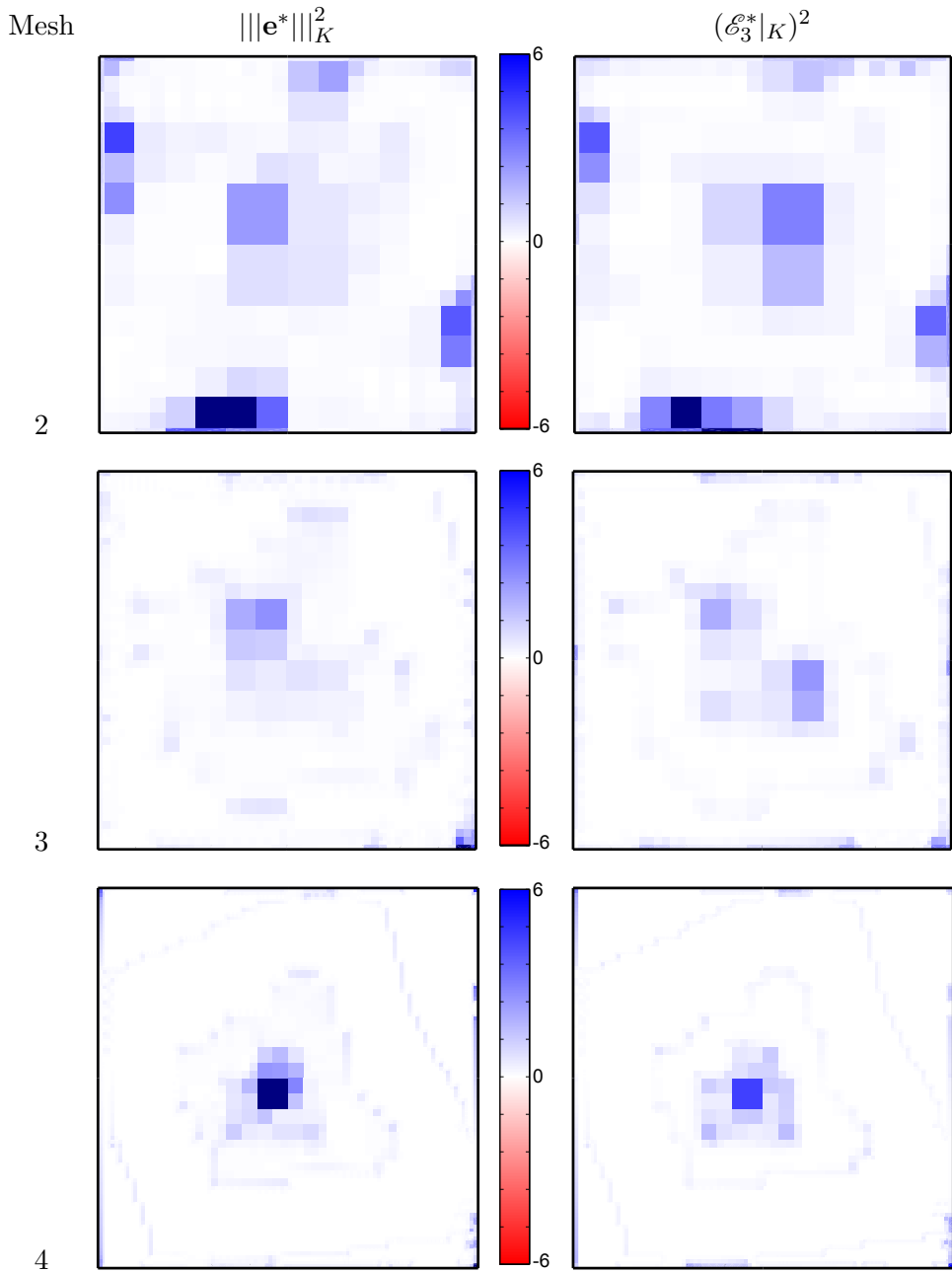


Figure 6.3: Problem 1b. Q4. Local exact error of the recovered solution (left), local error estimates using \mathcal{E}_3^* estimator (right).

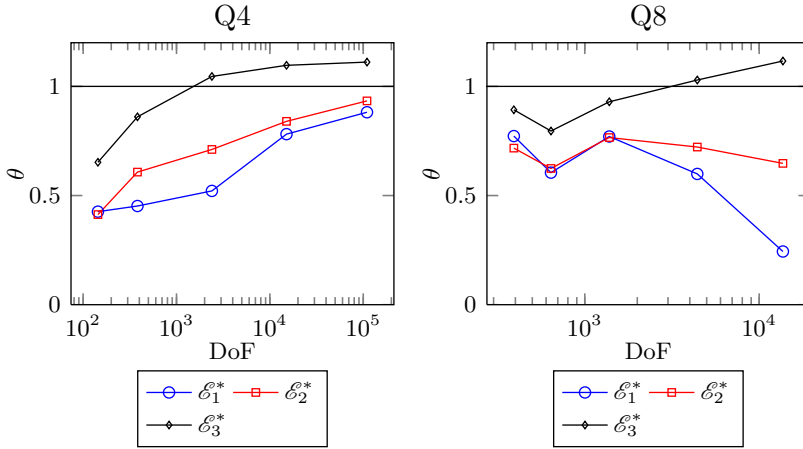


Figure 6.4: Problem 2. Global effectivity index during the h -adaptive refinement process.

Problem 3: L-Shape plate

Consider the problem of an infinite plate with a V-notch subjected to tractions. We have considered the Mode I loading condition. The model of the problem and further information can be found in Section 3.5.1.

Figure 6.6 shows again that \mathcal{E}_3^* exhibits the best results for linear and quadratic elements. In the last meshes of the analysis we always obtain a decrease of the effectivity index. This is because for the firsts meshes of each analysis the mesh increases its element density around the reentrant corner increasing the refinement level as we get closer to the singular point. The FE code can reach up to 22 refinement levels. In the last mesh the refinement level around the singularity requires higher refinement levels. The result is that, as higher refinement levels cannot be reached, we obtain an area around the singularity with elements of uniform size as opposite to graded meshes towards the singularity. This produces pollution errors and a decrease in the accuracy of the error estimation that leads to worse effectivity indexes.

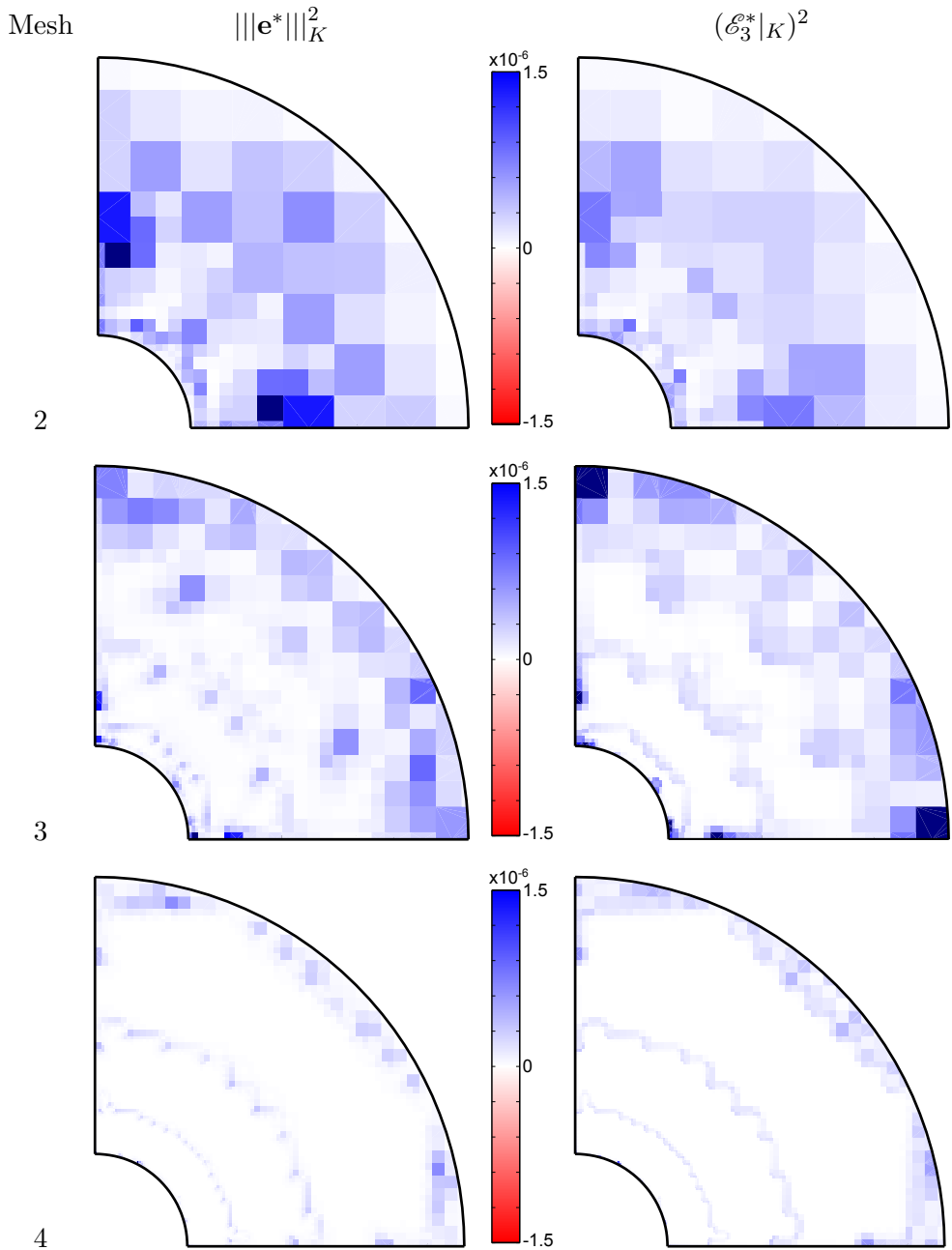


Figure 6.5: Problem 2. Q4. Local exact error of the recovered solution (left), local error estimates using \mathcal{E}_3^* estimator (right).

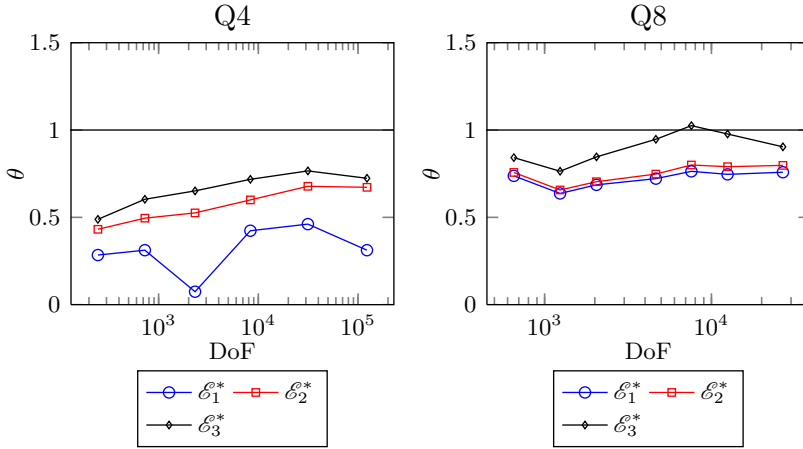


Figure 6.6: Problem 3. Global effectivity index during the h -adaptive refinement process.

We show the local error index $\mathcal{E}_3^*|_K$ and the exact error of the recovered solution, $|||\mathbf{e}^*|||_K$, in Figure 6.7. Observe that even for this singular problem the results are quite similar, concluding again that $\mathcal{E}_3^*|_K$ is a good error indicator.

6.4.2 h -adaptive process

The following results compare the traditional h -adaptive process with the proposed one that uses the recovered solution. Figures 6.8 and 6.9 show the evolution of the errors during the h -adaptive refinement process and the computational cost to obtain a certain accuracy level, for Problem 2 for Q4 and Q8 elements, see details in Section 3.5.1.

The horizontal black lines in Figures 6.8 and 6.9 represent the error level of the solution prescribed by the analyst. Red and brown lines represent the error (exact and estimated) of the standard FE output $(\mathbf{u}^h, \boldsymbol{\sigma}^h)$. Blue and black lines represent the error (exact and estimated) of the recovered

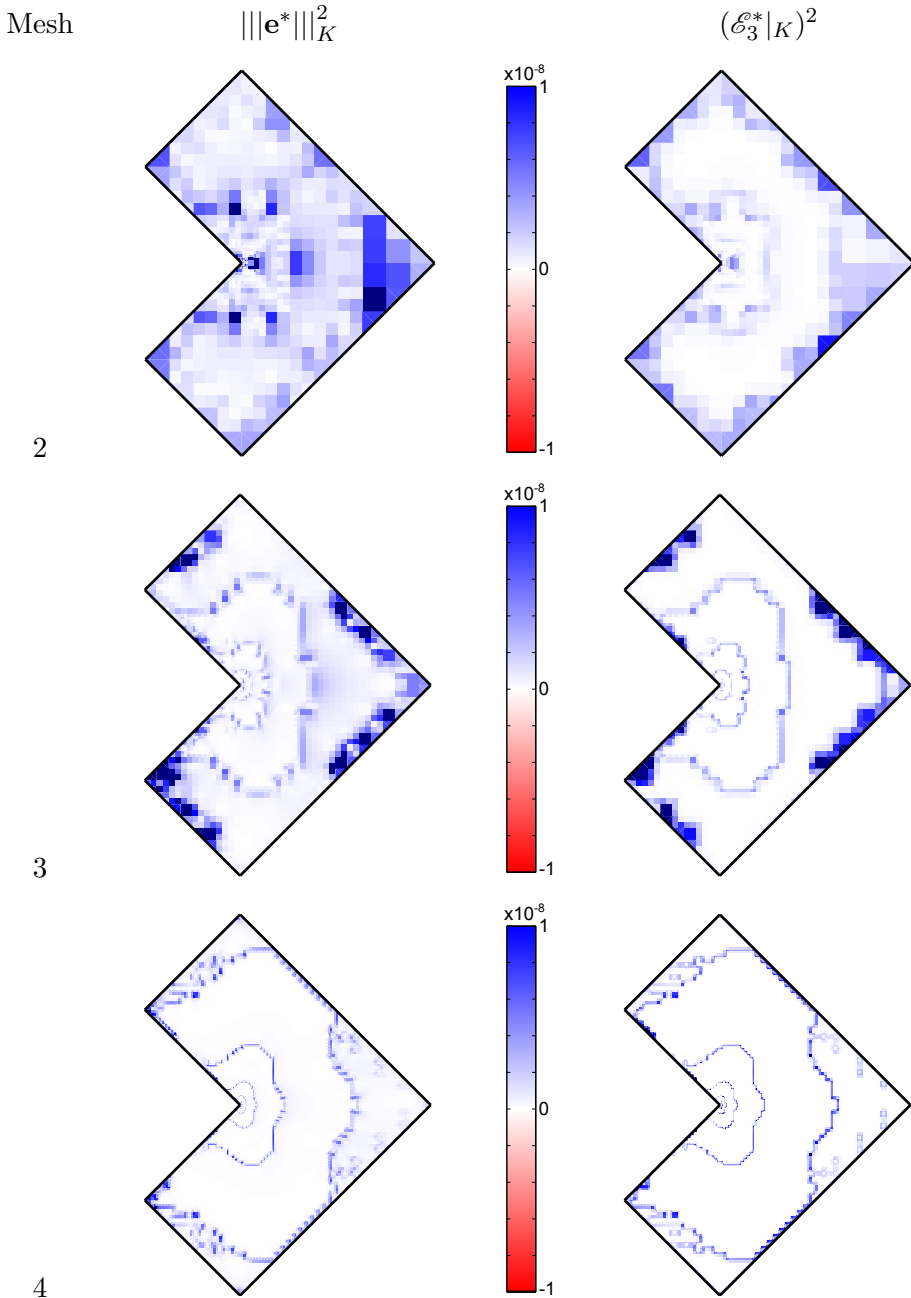
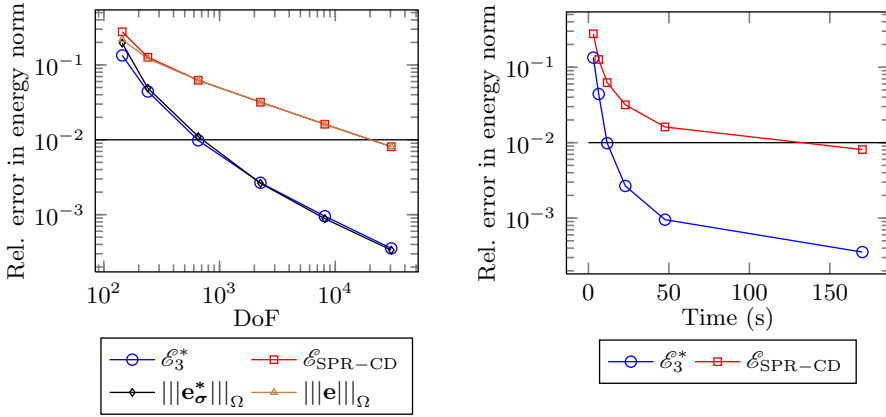


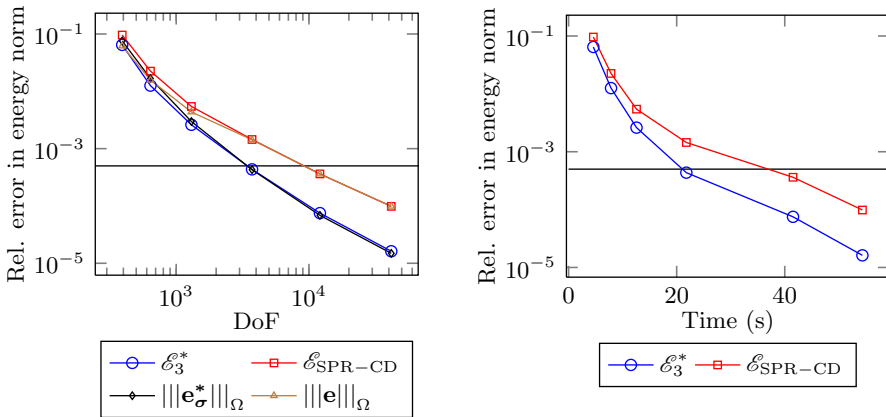
Figure 6.7: Problem 3. Q4. Local exact error of the recovered solution (left), local error estimates using \mathcal{E}_3^* estimator (right).



(a) Convergence analysis. Relative error in energy norm *versus* degree of freedom

(b) Computational cost analysis. Relative error in energy norm *versus* time.

Figure 6.8: Problem 2. h -adaptive analysis with Q4 elements. The black line represents the prescribed relative error in energy norm (1%).



(a) Convergence analysis. Relative error in energy norm *versus* degree of freedom

(b) Computational cost analysis. Relative error in energy norm *versus* time.

Figure 6.9: Problem 2. h -adaptive analysis with Q8 elements. The black line represents the prescribed relative error in energy norm (0.05%).

field $(\mathbf{u}_u^*, \boldsymbol{\sigma}_\sigma^*)$. We can observe that the error estimation of recovered solution accurately represents the exact error for both linear (Q4) and quadratic (Q8) elements. The Figures show a considerable improvement of the h -adaptive process to reach a prescribed error level. Note that, for Q4 elements, to reach the prescribed relative error in energy norm (1%), the standard h -adaptive strategy based on the accuracy of the FE solution would stop the process after an analysis with 30492 degrees of freedom and 170.2 s whereas with the proposed h -adaptive method based on the accuracy of the recovered solution the process would stop with only 654 degrees of freedom and 11.7 s. With Q8 elements reaching a 0.05% prescribed error, the standard h -adaptive procedure would stop the process after an analysis with 12102 degrees of freedom and 41.53 s, whereas the proposed method would only require 3728 degrees of freedom and 21.8 s.

6.5 Conclusions

In this Chapter we have presented several heuristic methods to estimate the error of the recovered solution, \mathcal{E}_1^* , \mathcal{E}_2^* and \mathcal{E}_3^* . We have validated them using several benchmark problems with satisfactory results. The best performance in all situations have been obtained by \mathcal{E}_3^* both, globally and locally. \mathcal{E}_3^* has been selected as the error estimator for the recovered solution. It will be used in Chapter 7 for shape optimization problems.

As mentioned before this is an heuristic method to evaluate the error of the recovered solution obtained with the SPR-CD process, however, the results obtained are very promising. This kind of techniques could represent a new paradigm leading to substantially more efficient FE analysis. Evidently this will require further studies to provide mathematical support to the results presented in this Chapter.

Chapter 7

Structural Shape Optimization

7.1 Introduction and motivation

As mentioned in Chapter 1 the final objective of this thesis is to obtain an efficient and also accurate FE code to be used as the lower level in a shape optimization process. In this Chapter we will use the cgFEM code developed in this thesis as the lower level of the optimization process. We will also compare the cgFEM implementation with a commercial code in order to check the performance of the proposed implementation. The cgFEM code includes the improvements with respect to traditional FE implementations described along the thesis, such as the sharing information procedures and the solving improvements described in Chapter 2. We also benefit from the advantages of the accuracy of the recovered solution for which we have developed an error estimator in Chapter 6. All these ingredients make the cgFEM code a serious alternative for structural shape optimizations processes.

The optimization processes used in this Section is based on Genetic Algorithms and requires a high number of evaluations of the objective function. In the particular case of structural shape optimization, each evaluation of the objective function requires to solve a full elasticity problem that it is highly time consuming, hence the importance in the efficiency of the cgFEM solver.

The discretization error of each individual must be controlled to ensure the convergence of the optimization process [1, 2]. This justifies the use of adaptive procedures to ensure that the prescribed accuracy level is obtained with the minimum computational cost. In any case, adaptive analysis involves a considerable computational cost. It is thus necessary to develop new methodologies with a higher computational efficiency and accuracy. The SPR-CD technique, introduced in this thesis, produces highly accurate recovered stress fields with a reduced computational cost because of the mesh structure, as indicated in Section 3.3. This recovery technique in combination the ZZ error estimator yields very accurate error estimations which is used for the h -adaptive refinement process based on element splitting. This element splitting procedure, and thus the regeneration of the meshes for the adaptive analysis, is very efficient thanks to the Cartesian grid. The numerical results will provide the evidence in the decrease of the total computational cost.

In an optimization process some parts of the geometry do not change along the process. The cgFEM code has been used to easily share information between individuals, reusing information previously evaluated in other individuals, as introduced in Section 2.3.8. This means that a significant part of the FE analysis of the current individual has been evaluated before (for other individuals) and stored in memory. Then, for the current individual the pre-evaluated data will not be computed again, it will only be loaded from memory, thus reducing the total amount of calculations to be performed.

The objective of this Chapter is to use the cgFEM code for structural shape optimization in order to check its performance, robustness and reliability. We

will also compare the cgFEM code with a commercial version of the traditional FEM code in order to check the behavior of the cgFEM technology.

7.2 The optimization problem

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

$$\begin{aligned} & \min f(\mathbf{x}) \\ & \text{where } \mathbf{x} = \{x_i\} \quad i = 1, 2, \dots, n \\ & \text{under } \mathbf{g}(\mathbf{x}) \leq \{g_j(\mathbf{x})\} \quad j = 1, 2, \dots, m \\ & \quad \quad \quad a_i \leq x_i \leq b_i \quad i = 1, 2, \dots, n \end{aligned} \tag{7.1}$$

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 (for example 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 cgFEM methodology can be obtained with any optimization algorithm. In our case, in the numerical examples, we have considered the use of the genetic algorithm (GA) proposed by Storn and Price [127]. More precisely, we use the *Differential Evolution* (DE) algorithm, version DE1.

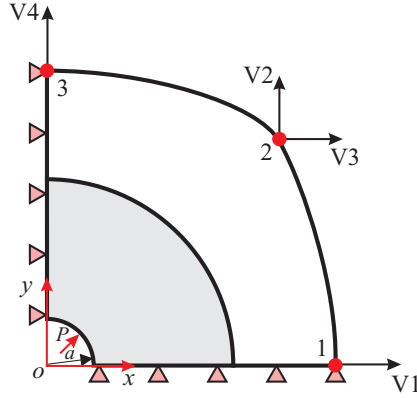
7.3 Numerical Results

In this Section we first present the results of a first problem with known exact geometry in order to test the optimization algorithm and also the cgFEM implementation coupled with the optimization software. The second problem does not have analytical solution but it has a more complex geometry. The results we will measure are the evolution of the computational cost during the optimization process, the discretization error in energy norm achieved at each individual and also the evolution of the area during the process.

For these numerical examples we have used a PC DELL PE1950 equipped with two processors Intel Xeon E5430 with 32Gb of memory. The operative system is Windows Server 2003 Enterprise x64 SP2. For the FE analysis we have used Q4 and Q8 elements, always under a h -adaptive refinement process.

7.3.1 Optimization problem 1. Cross section under pressure over internal circular shape

In order to observe the performance of the proposed solver, we compare it with a commercial code such as ANSYS[®] 12.1. We also compare the different performance when the raw FE solution is used as output or, instead, the output is the recovered solution. The data and the design variables are shown in Figure 7.1. The external boundary is represented by using a cubic spline, defined by points 1, 2, and 3. The optimal shape for this problem corresponds to a thick-wall cylinder under internal pressure. The analytical solution corresponds with problem 2, see Section 3.5.1. Considering a yield stress $S_y = 2 \cdot 10^6$, the analytical optimal shape is a thick-wall cylinder with external circular surface of radius $R_0 = 10.67033824461$. The optimal area is $A_{opt} = 69.787307715081$. Analytically, the optimal solution is obtained minimizing the external radius in order to obtain a Von Mises stress value, at the internal radius, equal to the yield stress.



$$E = 10.1 \cdot 10^6, \nu = 0.3, a = 5, P = 0.9 \cdot 10^6$$

Figure 7.1: Optimization problem 1. Model and data. The optimal shape corresponds to a thick-wall cylinder under internal pressure as represented in gray.

Table 7.1 shows the constraints imposed over the design variables and their range. The number of individuals at each iteration is 30 and the number of iterations is 150. We have performed analyses with Q4 and Q8 elements for different prescribed error levels, γ , to study the influence of the error in energy norm on the computational cost and on 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.

- *Strategy a* σ^h . We have used the FE solution σ^h as the output of the analysis together with a criterion to stop the adaptive process based on $\mathcal{E}_{\text{SPR-CD}}$, see (3.80).
- *Strategy b* σ^h . We have used the recovered solution σ_σ^* and a criterion to stop the adaptive process based on the error estimator for this solution \mathcal{E}_3^* , see (6.6c). Details of the refinement process can be found in Section 6.4.2.

Variables	Description	Range	Constrain	Reference solution
V1	x point 1	[5, 24]		20
V2	y point 2	[5, 23]		19
V3	x point 2	[5, 23]	$V3 < V1 - 0.5$	19
V4	y point 3	[5, 24]	$V4 > V2 + 0.5$	20

Table 7.1: Optimization problem 1. Design variables constraints and reference solution.

First we present the results referring to Q4 elements to evaluate the time reduction for the *strategy b* in comparison with the *strategy a*. Later, we will compare the results obtained with the cgFEM code with those obtained with ANSYS® 12.1 as a solver. Figure 7.2 shows at the left, the evolution of the exact error in area through the optimization process. First we can observe that the final error in area of the optimization process strongly depends on the prescribed error in energy norm during the FE analysis. These results are in agreement with [2, 4] where the importance of the error control for optimization processes was emphasized. The second aspect that we can observe is that for the same prescribed error ($\gamma = 5\%$ or $\gamma = 2\%$) we observe a higher accuracy when *strategy b* used. This is because of the higher accuracy of the recovered field along the boundaries, which is usually the critical area for design. The right graph in Figure 7.2 shows that the discretization error obtained during the optimization process, is below the prescribed one. The discretization error at each generation is evaluated as the mean discretization error of the individuals in that generation.

Figure 7.3 shows a graph comparing the accumulative computational cost obtained for both *strategies* and for the different prescribed error levels. Comparing the continuous lines for $\gamma \leq 5\%$ we observe a reduction in computational cost, but it is more interesting for $\gamma \leq 2\%$ where the reduction in computational cost is around 50% for the same level of accuracy. Hence the importance of using the *strategy b* for optimization processes.

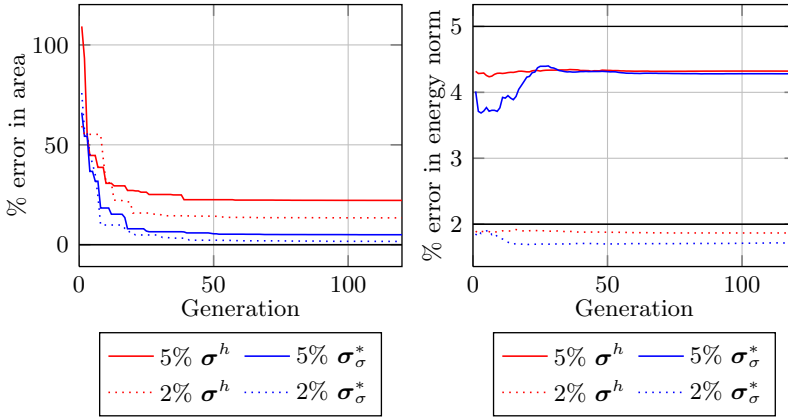


Figure 7.2: Optimization problem 1. Q4. Evolution of cylinder optimization considering the error estimation of the FE solution (σ^h) or the recovered one (σ_σ^*), for different prescribed errors levels γ . *Left*: Evolution of exact error in area with respect to the optimal analytical solution. *Right*: Real error in energy norm.

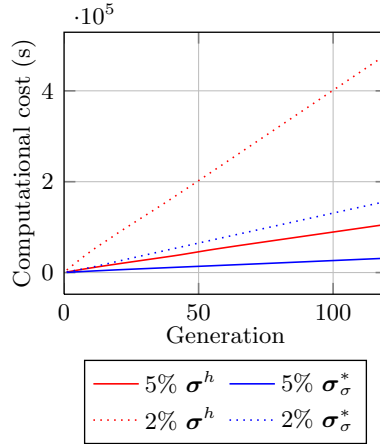


Figure 7.3: Optimization problem 1. Q4. Evolution of the optimization process considering the error estimation of the FE solution (σ^h) or the recovered one (σ_σ^*), for different prescribed errors levels γ . Evolution of the accumulative computational cost.

We will now present the results concerning Q8 elements. We will compare these results, obtained with the cgFEM solver and *strategy b*, with those obtained

with ANSYS® 12.1. The analysis in ANSYS® 12.1 have been carried out with the element *PLANE 82*, plane strain configuration. The *h*-adaptive refinement process has been carried out using the *ADAPT* function implemented with the default configuration. The left hand side of Figure 7.4 shows that both solvers are giving the similar results. The graph on the right shows the mean error in energy norm at each iteration. In this case we have chosen the criterion $\gamma \leq 1$ (black line). We observe that the error obtained with cgFEM solver is always below the prescribed one, however when ANSYS® 12.1 is the solver, it can not achieve the objective for the firsts iterations. This is not critical for the optimization processes because a higher error at the beginning of the process does not affect to the final result, but shows that cgFEM has been more robust than ANSYS® 12.1, always providing FE models with the prescribed accuracy level.

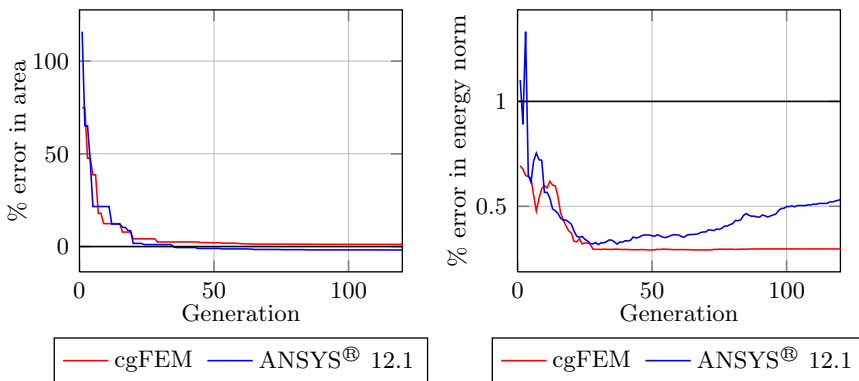


Figure 7.4: Optimization problem 1. Q8. Evolution of cylinder optimization considering the cgFEM as a solver (red line) or ANSYS® 12.1 (blue line) for $\gamma \leq 1$. *Left*: Evolution of exact error in area with respect to the optimal analytical solution. *Right*: Real error in energy norm.

In Figure 7.5 we plot the accumulated computational cost of the optimization process for the cgFEM and also ANSYS® 12.1. As it can be appreciated, the cgFEM solver is around a 30% faster than the commercial code. It is important to mention that the cgFEM solver is fully implemented in Matlab® 2010b, without any compiled routine. This reduction in computational cost is due to the hierarchical data structure of the cgFEM method that allows a fast

data sharing as indicated in Section 2.3.8. Additionally the use of the *strategy b* also improves considerably the computational cost, making it competitive with commercial codes.

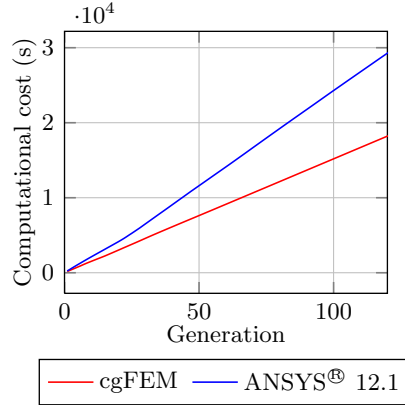


Figure 7.5: Optimization problem 1. Q8. Evolution of the optimization process considering the cgFEM as a solver (red line) or ANSYS[®] 12.1 (blue line) for prescribed relative error $\gamma \leq 1$. Evolution of the accumulative computational cost.

7.3.2 Optimization problem 2. Gravity dam

This problem consist in the structural optimization of the section of a gravity dam. The optimization process will change the shape of the internal surface in order to decrease the dam volume. Figure 7.6a shows the global problem model and the material data. The optimization process consists in 100 iterations with 30 individuals in each generation. Figure 7.7a shows a detail of the internal hole defined by a spline (points 22, 23, 24, 25, 26, 27 and 21) and by a straight line (points 21 and 22). Material properties are defined in Figure 7.7b. Table 7.1 shows the input constraints of the design variables and its range. We have performed several analysis with Q4 and Q8 elements using a h -adaptive refinement process for this problem.

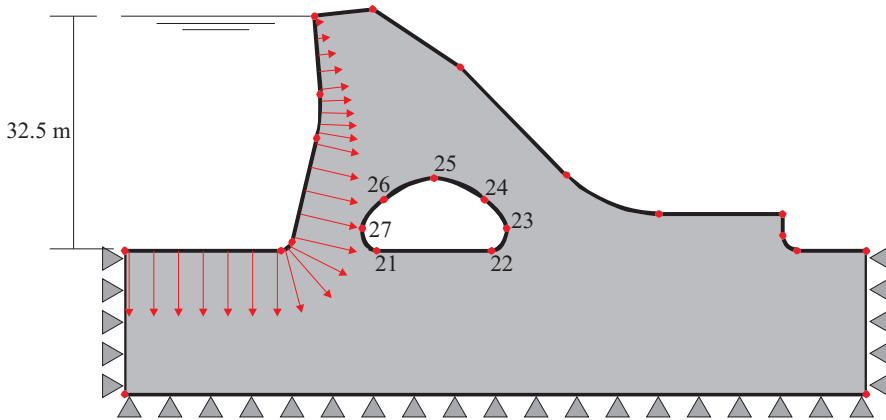


Figure 7.6: Optimization problem 2. Gravity dam. Model of the problem.

The left graph in Figure 7.9 represents the evolution of the area through the optimization process for the *strategy a* (red line) and also for *strategy b*. The convergence in both methods is similar, as expected. For this problem, we have implemented a variation of the prescribed error (γ) along the optimization process as shown by the in black line on the right graph in Figure 7.9. Initially we allow a 20% of error in energy norm and then we gradually decrease it to 2.5% in 20 iterations. Afterward it will remain constant till the end of the optimization process. This allows to increase the speed of the calculations at the initial iterations of the optimization process without affecting the final results. Further research should be addressed in this sense for further increase the efficiency of the optimization process. Both techniques always achieve an error smaller than the prescribed one.

As shown in 7.10 when solving this problem with cgFEM and *strategy a* (red line) the computational cost becomes unaffordable for the prescribed evolution of the error in energy norm presented in Figure 7.9. Thus, we had to stop the process after 30 generations. However, when *strategy b* is used, the computational cost decreases considerably, making possible to solve the whole optimization process, that is, reaching 100 generations, with a reasonable computational cost.

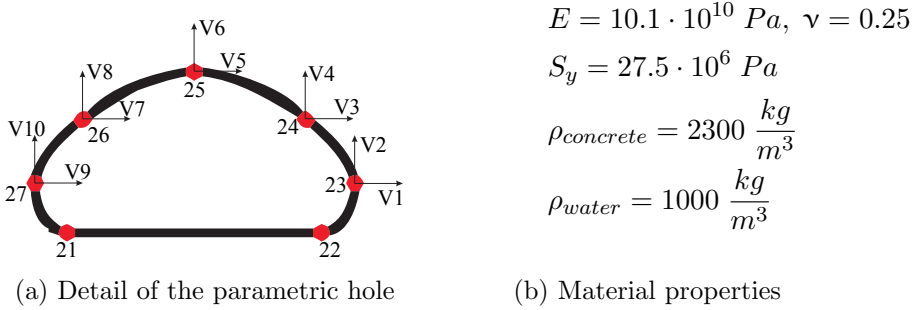


Figure 7.8: Optimization problem 2. Gravity dam. Detail of the parametric hole and material properties.

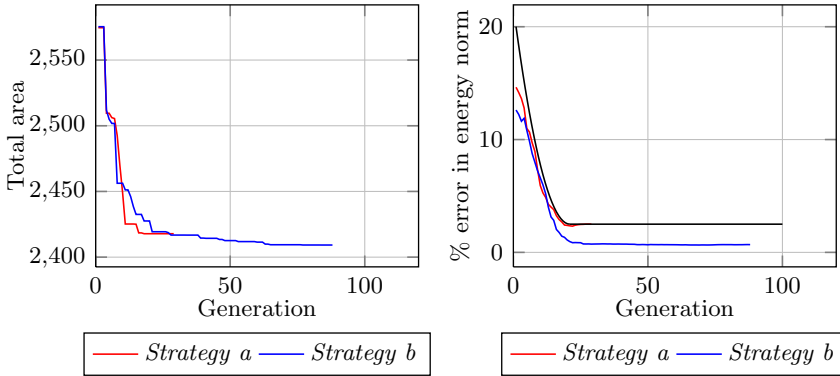


Figure 7.9: Optimization problem 2. Gravity dam. Q4. Evolution of the dam optimization considering the cgFEM as a solver. Using *strategy a* (red line) or *strategy b* (blue line) for γ variable. *Left*: Evolution of the area. *Right*: Evolution of the estimated error in energy norm.

As for the previous example, we compare the results for Q8 obtained with cgFEM with *strategy b* with those obtained with the commercial code ANSYS[®] 12.1. Figure 7.11 shows on the left, the evolution of the area of the optimization process. In this example it seems, because of the scale ranges, that they stabilize around a different value of area, however their values are actually similar. For the cgFEM the optimal value is 2486.252888 and for ANSYS[®] 12.1 it is 2435.90375. This difference could be due to the randomness of the

Variables	Description	Range	Constrain	Reference solution
V1	x point 23	[50, 60]		50
V2	y point 23	[2, 10]		30
V3	x point 24	[41, 50]	$V3 < V1 - 2$	47
V4	y point 24	[3, 20]	$V4 > V2 + 3$	7
V5	x point 25	[25, 48]	$V5 < V3 - 3$	40
V6	y point 25	[3, 25]	$V6 > V4 + 2$	10
V7	x point 26	[25, 39]	$V7 < V5 - 3$	33
V8	y point 26	[3, 20]	$V8 < V6 - 2$	7
V9	y point 27	[23, 30]	$V9 < V7 - 2$	30
V10	y point 27	[2, 10]	$V10 < V8 - 3$	3

Table 7.2: Optimization problem 2. Gravity dam. Design variables constraints and reference solution.

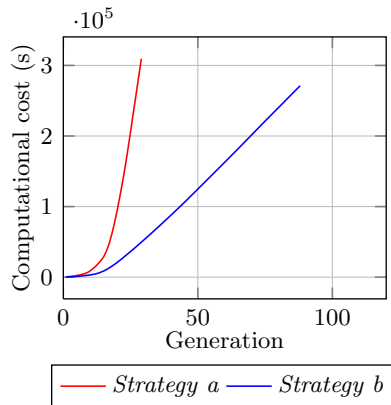


Figure 7.10: Optimization problem 2. Q4. Evolution of the dam optimization considering the cgFEM as a solver. Using *strategy a* (red line) or *strategy b* (blue line) for γ variable. Evolution of the accumulative computational cost.

GA used and also because of the Neumann boundary conditions which in ANSYS[®] 12.1 are limited to a linear evolution in the curves of the wall of the dam where the hydrostatic pressure is applied. On the right hand side of the Figure, we observe that both approaches provide an error in energy norm smaller than the prescribed one, as expected.

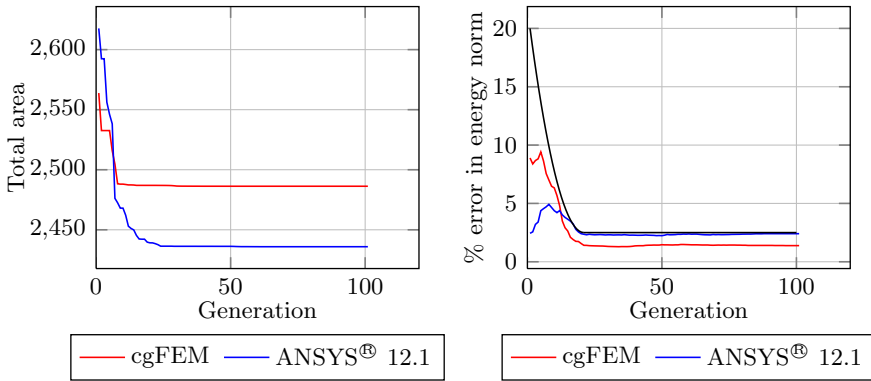


Figure 7.11: Optimization problem 2. Q8. Evolution of optimization process considering the cgFEM with *strategy b* as a solver (red line) or ANSYS® 12.1 (blue line) for variable prescribed error. *Left*: Evolution of the total area. *Right*: Estimated error in energy norm.

Finally, Figure 7.12 shows the evolution of the computational cost for solving the optimization process with the commercial code and the cgFEM technology proposed in this work. We observe that for this case the computational cost is similar in both analysis, however the cgFEM remains competitive when we consider that is fully implemented in Matlab® 2010b. Future compiled versions of the cgFEM technology would considerably improve its performance.

7.4 Conclusions

In this Chapter we have presented the cgFEM code working as the lower level of a structural shape optimization process based on genetic algorithms. We have used all the developments described along this thesis in order to improve the performance of the cgFEM technology obtaining satisfactory results. The results show that, even when the cgFEM is fully implemented in Matlab® 2010b, the overall computational cost of the optimization process is compara-

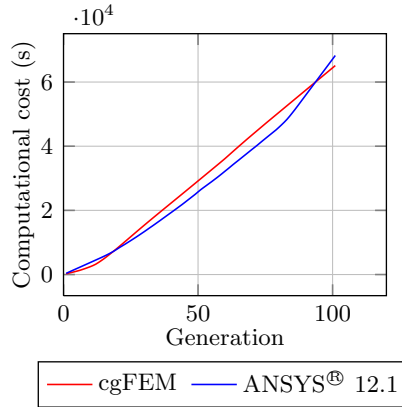


Figure 7.12: Optimization problem 2. Q8. Evolution of optimization process considering the cgFEM *strategy b* as a solver (red line) or ANSYS[®] 12.1 (blue line) for variable prescribed error. Evolution of the accumulative computational cost.

ble to the one obtained with commercial codes. Future compiled versions of the cgFEM technology will considerably improve its performance.

The results presented in this Chapter also show the excellent behavior of the *h*-adaptive refinement process and of both, the error estimation of the FEM solution and the error estimation of the recovered solution, providing always results at the prescribed level of accuracy. We should also remark the improvement in computational cost obtained when the *strategy b* is used.

Future research can be addresses in the field of the prescribed error evolution and is influence in the final shape in order to decrease the computational cost of the optimization process which so far, continues being excessive, especially if evolutionary algorithms are used.

Chapter 8

Conclusions

During this thesis we have introduced the cgFEM technology for solving the linear elasticity problem. This method has proved its competitiveness against the traditional FEM with a variety of numerical examples showed in this work. The key points of the increased performance of the cgFEM technology are:

- The use of Cartesian grids that allows making the mesh independent of the geometry embedding the problem domain into a bigger domain for which the process of creating a FE mesh results trivial. The geometry is transferred to the FE model via an intersection process that will define the relative position of the elements with the geometry. After the intersection process we obtain internal elements, which do not require any special treatment, external elements that will not be considered in the analysis and intersected elements, whose element matrices will be obtained using a special integration procedure. This process avoids the tedious meshing process of the traditional FEM codes, see Chapter 2.
- The hierarchical data structure, easily built due to the Cartesian grid mesh structure, allows for an easy to implement (and computationally

efficient) data sharing process. This process permits to share integration information between elements, such as the stiffness and, in general, all basic element data. The sharing data process can be applied into the h -adaptive refinement process (vertical data sharing) or also in the structural shape optimization process (horizontal data sharing), see Chapter 2.

- The robust SPR-CD technique. This versatile technique is the kernel of the error estimation process of the cgFEM code. This technique in combination with the ZZ error estimation is used to drive the h -adaptive refinement process based on the error in energy norm. Additionally the SPR-CD technique provides numerical upper error bounds and guaranteed lower error bounds in energy norm, see Chapter 3. Furthermore, the error estimation in QoI and also the numerical bounding techniques in QoI are based on the used of the SPR-CD technique, see Chapter 4. The FUB upper bound technique of the error in energy norm is also based on the SPR-CD technique results. And finally, the error estimation of the recovered solution provided by the SPR-CD technique is also based on a fast post-processing the SPR-CD output, see Chapter 6. Then the SPR-CD technique is mainly present in all the developments of this thesis.
- The introduction of a heuristic error estimator for the recovered field provided by the SPR-CD technique. In Chapter 6 we have introduced three practical error estimators for the recovered solution provided by the SPR-CD technique. This fact allows to use the recovered solution, of a better quality than the raw FE solution, as output of the cgFEM code. This permits to evaluate the solution for a prescribed error level with a considerably reduction in the computational cost as the results have shown.

All these features of the cgFEM code presented along this thesis make the code competitive for practical applications. We consider remarkable the substitution of the FE solution by the recovered solution provided by the SPR-CD

technique. This is possible because of the introduction of the corresponding error estimator for the recovered stress field. This introduces an important increase in the performance of the cgFEM code. In order to check the competitiveness of the code, we have used the cgFEM code as the solver of structural shape optimization processes (see Chapter 7). This has allowed to apply all the improvements described in the previous enumeration and control its performance and robustness in a vast number of different geometries. The results obtained for the structural shape optimization processes presented show the robustness of the cgFEM code. A basic comparison with a commercial code has shown that the FE code developed in this thesis is at least as fast as the commercial code. This can be considered as an outstanding result taking into account that our code has been fully implemented in Matlab[®] 2010b.

Along this thesis we have also explained the possibility to interpret the dual problem, used to evaluate the error in QoIs, as an abstract elasticity problem. In Chapter 4 we present some relations for practical QoIs that permit to interpret the abstract dual load as a set of traction, body forces, initial stresses, initial strains,... This allowed us to use the SPR-CD techniques to evaluate a recovered field of the dual problem locally equilibrated. Combining the recovered solutions of the primal problem and also of the dual one in a modified version of the ZZ error estimator we have obtained very accurate error estimations of the error in the QoI as well as error bounds, see Chapter 4.

Finally, we would like to remark that we have introduced error bounding techniques based on recovery procedures, which are able to provide guaranteed upper error bounds in energy norm. Among these three techniques we would like to highlight the behavior of the FUB technique which provides accurate results at a reasonable computational cost and that could be easily implemented into existing commercial codes. The FUB technique consists in a post-process of the SPR-CD recovered stress field. This post-process corrects the ZZ error estimation accounting for the lacks of internal and boundary equilibrium of the recovered solution, see Chapter 5.

8.1 Future works

This work has opened several lines of research for the cgFEM technology. Now we will summarize the most important:

- To extend the cgFEM technology to 3D. The crucial aspect of this point is to accurately and also efficiently intersect the 3D geometry with the mesh structure. This process is even more complex when considering that the 3D body is defined by Non-Uniform Rational B-Splines (NURBS). A great effort in this sense is being done in the Department of Mechanical Engineering and Materials at UPV.
- To enrich the FE basis functions. It would be very interesting to represent weak discontinuities, such as internal material interfaces or even strong discontinuities such as cracks. This task should be addressed by implementing XFEM in the cgFEM code. This would allow to easily deal with crack propagation and would provide interesting capabilities to the cgFEM code.
- To use the upper bounding procedures introduced in Chapter 5 to the error estimation in QoI. The FUB technique is easy to implement and computationally efficient. Thus, it would be interesting to apply this technique to obtain guaranteed, and accurate, bounds in the QoI.
- Another interesting point of this thesis is that we have introduced a method to evaluate the error of the recovered solution provided by the SPR-CD technique. This allows to use the recovered fields as the standard output of the cgFEM code. However, this error estimation requires consistent mathematical support.

Bibliography

- [1] Bugeda G, Oliver J. A general methodology for structural shape optimization problems using automatic adaptive remeshing. *International Journal for Numerical Methods in Engineering* 1993; **36**(18):3161–3185.
- [2] Ródenas JJ, Bugeda G, Albelda J, Oñate E. On the need for the use of error-controlled finite element analyses in structural shape optimization processes. *International Journal for Numerical Methods in Engineering* 2011; **87**:1105–1126.
- [3] Zienkiewicz OC, Zhu JZ. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering* 1987; **24**(2):337–357.
- [4] Ródenas JJ, Bugeda G, Albelda J, Oñate E, Nadal E. Sobre la necesidad de controlar el error de discretización de elementos finitos en optimización de forma estructural con algoritmos evolutivos. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* 2012; **28**(1):1–11.
- [5] Bordas SPA, Rabczuk T, Ródenas JJ, Kerfriden P, Moumnassi M, Belouettar S. Recent Advances Towards Reducing the Meshing and Remeshing Burden in Computational Sciences. *Computational Technology Reviews, vol. 2* 2010; :21–82.

- [6] Nadal E, Ródenas JJ, Sánchez-Orgaz EM, López-Real S, Martí-Pellicer J. Sobre la utilización de códigos de elementos finitos basados en mallas cartesianas en optimización estructural. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* 2013; **Accepted**.
- [7] Zienkiewicz OC, Zhu JZ. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *International Journal for Numerical Methods in Engineering* 1992; **33**(7):1331–1364.
- [8] Zienkiewicz OC, Zhu JZ. The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity. *International Journal for Numerical Methods in Engineering* 1992; **33**(7):1365–1382.
- [9] Daneshmand F, Kazemzadeh-Parsi MJ. Static and dynamic analysis of 2D and 3D elastic solids using the modified FGFEM. *Elements in Analysis and Design* 2009; **228**:3891–3910.
- [10] Stein E, Olgierd C, Zienkiewicz, a pioneer in the development of the finite element method in engineering science. *Steel Construction* 2009; **2**(4):264–272.
- [11] Bishop J. Rapid stress analysis of geometrically complex domains using implicit meshing. *Computational Mechanics* 2003; **30**:460–478.
- [12] Cottrell JA, Hughes TJR, Bazilevs Y. *Isogeometric Analysis: Toward Integration of CAD and FEA*. 1st edn., Wiley, 2009.
- [13] Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(41-44):2680–2686.
- [14] Heikkola E, Kunzetsov YA, Lipnikov KN. Fictitious domain methods for the numerical solution of three-dimensional acoustic scattering problems. *Journal of Computational Acoustics* 1999; **7**(3):161–193.

- [15] Hetmaniuk U, Farhat C. 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* 2003; **39**:707–725.
- [16] Farhat C, Hetmaniuk U. 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* 2002; **54**:1309–1332.
- [17] Ye T, Mittal R, Udaykumar HS, Shyy W. Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries. *Journal of Computational Physics* 1999; **156**:209–240.
- [18] Jahangirian A, Shoraka Y. Adaptive unstructured grid generation for engineering computation of aerodynamic flows. *Mathematics and Computers in Simulation* 2008; **78**:627–644.
- [19] Silva Santos CM, Greaves DM. Using hierarchical Cartesian grids with multigrid acceleration. *International Journal for Numerical Methods in Engineering* 2007; **69**:1755–1774.
- [20] Preusser T, Rumpf M, Schwen LO. Finite Element Simulation of Bone Microstructures. *14th FE Workshop*, 2007; 1–15.
- [21] Ramière I, Angot P, Belliard M. Fictitious domain methods to solve convection-diffusion problems with general boundary conditions. *Proc. in the 17th Computational Fluid Dynamics Conference AIAA*, 2005.
- [22] DeRose Jr GCA, Díaz AR. Solving three-dimensional layout optimization problems using fixed scale wavelets. *Computational Mechanics* 2000; **25**:274–285.
- [23] Jang GW, Kim YY, Choy KK. Remesh-free shape optimization using the wavelet-Galerkin method. *International Journal of Solids and Structures* 2004; **41**:6465–6483.

- [24] Mäkinen RAE, Rossi R, Toivanen J. A moving mesh fictitious domain approach for shape optimization problems. *Mathematical Modeling and Numerical Analysis*. 2000; **34**(1):31–45.
- [25] Victoria M, Querin OM, Martí P. Topology design for multiple loading conditions of continuum structures using isolines and isosurfaces. *Finite Elements in Analysis and Design* 2010; **46**(3):229–237.
- [26] Parussini L, Pedirola V. Fictitious Domain approach with hp-finite element approximation for incompressible fluid flow. *Journal of Computational Physics* 2009; **228**:3891–3910.
- [27] Zhang L, Gerstenberger A, Wang X, Liu WK. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:2051–2067.
- [28] Roma AM, Peskin CS, Bergery MJ. An Adaptive Version of the Immersed Boundary Method. *Journal of Computational Physics* 1999; **153**:509–534.
- [29] García-Ruíz MJ, Steven GP. Fixed grid finite elements in elasticity problems. *Engineering Computations* 1999; **16**:145–164.
- [30] Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* 1999; **46**(1):131–150.
- [31] Sukumar N, Prévost JH. Modeling quasi-static crack growth with the extended finite element method. {Part I}: Computer implementation. *International Journal of Solids and Structures* 2003; **40**:7539–7552.
- [32] Strouboulis T, Copps K, Babuška I. The generalized finite element method. *Computer Methods in Applied Mechanics and Engineering* May 2001; **190**(32-33):4081–4193.
- [33] Strouboulis T, Zhang L, Babuscaronka I. Assessment of the cost and accuracy of the generalized FEM. *International Journal for Numerical Methods in Engineering* 2007; **69**(2):250–283.

- [34] Melenk JM, Babuška I. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**(1-4):289–314.
- [35] Stolarska M, Chopp DL, Moës N, Belytschko T. Modelling crack growth by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering* 2001; **51**(8):943–960.
- [36] Mournassi M, Belouettar S, Béchet E, Bordas SPA, Quoirin D, Potier-Ferry M. Finite element analysis on implicitly defined domains: An accurate representation based on arbitrary parametric surfaces. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**(5-8):774–796.
- [37] Lian WD, Legrain G, Cartraud P. Image-based computational homogenization and localization: comparison between X-FEM/levelset and voxel-based approaches. *Computational Mechanics* 2013; **51**(3):279–293.
- [38] Legrain G, Chevaugéon N, Dréau K. High order X-FEM and levelsets for complex microstructures: Uncoupling geometry and approximation. *Computer Methods in Applied Mechanics and Engineering* 2012; **241-244**(0):172–189.
- [39] Simpson RN, Bordas S, Trevelyan J, Rabczuk T. A two-dimensional Isogeometric Boundary Element Method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering* 2012; **209-212**:87–100.
- [40] Scott M, Simpson RN, Evans J, Lipton S, Bordas S, Hughes TJR, Sederberg TW. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering* 2013; **254**:197–221.
- [41] Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**(39-41):4135–4195.

- [42] Verfürth R. A review of a posteriori error estimation techniques for elasticity problems. *Computational Methods in Applied Mechanics and Engineering* 1999; **176**:419–440.
- [43] Zienkiewicz OC, Taylor R. *The Finite Element Method*, vol. 1. 4 edn., Butterworth-Heinemann: Oxford, UK, 1989.
- [44] Meagher D. Geometric modeling using octree encoding. *Computer Graphics and Image Processing* 1982; **19**(2):129–147.
- [45] Yerry MA, Shephard MS. Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering* 1984; **20**(11):1965–1990.
- [46] Baehmann PL, Wittchen SL, Shephard MS, Grice KR, Yerry MA. Robust, geometrically based, automatic two-dimensional mesh generation. *International Journal for Numerical Methods in Engineering* 1987; **24**(6):1043–1078.
- [47] Jackins CL, Tanimoto SL. Oct-trees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing* 1980; **14**(3):249–270.
- [48] Ródenas JJ, Tarancón JE, Albelda J, Roda A, Fuenmayor FJ. Hierarchical properties in elements obtained by subdivision: A hierarchical h-adaptivity program. *Adaptive Modeling and Simulation 2005*, Díez P, Wiberg NE (eds.), 2005.
- [49] Abel JF, Shephard MS. An algorithm for multipoint constraints in finite element analysis. *International Journal for Numerical Methods in Engineering* 1979; **1979**(3):464–467.
- [50] Farhat C, Lacour C, Rixen D. Incorporation of linear multipoint constraints in substructured based iterative solvers. Part 1: a numerically scalable algorithm. *International Journal for Numerical Methods in Engineering* 1998; **43**(6):997–1016.

- [51] Loon R, Anderson PD, Hart J, Baaijens FPT. A combined fictitious domain/adaptive meshing method for fluid-structure interaction in heart valves. *International Journal for Numerical Methods in Fluids* 2004; **46**(5):244–533.
- [52] Guo B, Babuška I. The h-p version of the finite element method. *Computational Mechanics* 1986; **1**:21–41.
- [53] Tarancón JE. Estimación y control del error de discretización en el método hp de elementos finitos. PhD Thesis, Universitat Politècnica de València 2002.
- [54] Löhner R, Cebal J, Castro M, Baum JD, Luo H, Mestreau E, Soto O. Adaptive Embedded Unstructured Grid Methods. *Mecánica Computacional Vol. XXIII*, Buscaglia G, Dari E, Zamonsky O (eds.). Bariloche, 2004.
- [55] Fuenmayor FJ, Oliver JL. Criteria to achieve nearly optimal meshes in the h-adaptive finite element method. *International Journal for Numerical Methods in Engineering* 1996; **39**(23):4039–4061.
- [56] Tur M, Albelda J, Nadal E, Ródenas JJ. Imposing Dirichlet boundary conditions in hierarchical cartesian meshes by means of stabilized Lagrange multipliers. *International Journal for Numerical Methods in Engineering* 2013; **Accepted**.
- [57] Corral C, Giménez I, Tur M, Ródenas JJ. Iterative Preconditioned Methods for the Solution of Contact Problems by the Finite Element Method. *Proceedings of the Fifth International Conference on Engineering Computational Technology*, 2006; 87.
- [58] Kettil P, Ekevid T, Wiberg NE. Towards fully mesh adaptive FE-simulations in 3D using multi-grid solver. *Computers & Structures* May 2003; **81**(8-11):735–746.
- [59] Collins MP, Vecchio FJ. Failure of and offshore platform 2000.

- [60] van der Zee KG, Dede L, Prudhomme S. Convergence of goal-oriented error estimates. *6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012)*, Eberhardsteiner J, Böhn H, Rammerstorfer F (eds.), TU Wien: Vienna, Austria, 2012; 394.
- [61] Babuška I, Rheinboldt WC. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering* 1978; **12**(10):1597–1615.
- [62] Gerasimov T, Rüter M, Stein E. An explicit residual-type error estimator for Q 1 -quadrilateral extended finite element method in two-dimensional linear elastic fracture mechanics. *International Journal for Numerical Methods in Engineering* 2012; **90**(April):1118–1155.
- [63] Stein E, Gerasimov T, Rüter M. Error-controlled adaptive multiscale analysis for crack initiation and propagation in brittle materials. *Adaptive Modeling and Simulation 2013*, Moitinho de Almeida JP, Díez P, Tiago C, Parés N (eds.), Lisbon, 2013; 26.
- [64] Ainsworth M, Oden JT. *A posteriori Error Estimation in Finite Element Analysis*, vol. 142. John Wiley & Sons: Chichester, 2000.
- [65] Díez P, Parés N, Huerta A. Recovering lower bounds of the error by post-processing implicit residual a posteriori error estimates. *International Journal for Numerical Methods in Engineering* 2003; **56**(10):1465–1488.
- [66] Díez P, Parés N, Huerta A. Accurate upper and lower error bounds by solving flux-free local problems in stars. *Revue européenne des éléments finis* 2004; **13**(5-6-7):497.
- [67] Zhang Z, Zhu JZ. Analysis of the superconvergent patch recovery technique and a posteriori error estimator in the finite element method (II). *Computer Methods in Applied Mechanics and Engineering* 1998; **163**(1-4):159–170.
- [68] Wiberg NE, Abdulwahab F, Ziukas S. Enhanced superconvergent patch recovery incorporating equilibrium and boundary conditions.

- International Journal for Numerical Methods in Engineering* 1994; **37**(20):3417–3440.
- [69] Blacker T, Belytschko T. Superconvergent patch recovery with equilibrium and conjoint interpolant enhancements. *International Journal for Numerical Methods in Engineering* 1994; **37**(3):517–536.
- [70] Ramsay ACA, Maunder EAW. Effective error estimation from continuous, boundary admissible estimated stress fields. *Computers & Structures* 1996; **61**(2):331–343.
- [71] Ródenas JJ, Tur M, Fuenmayor FJ, Vercher A. Improvement of the superconvergent patch recovery technique by the use of constraint equations: the SPR-C technique. *International Journal for Numerical Methods in Engineering* 2007; **70**(6):705–727.
- [72] Ródenas JJ, Giner E, Fuenmayor FJ, González-Estrada OA. Accurate recovery-type error estimation for linear elastic fracture mechanics in FEM and X-FEM based on a singular+smooth field splitting. *International Conference on Adaptive Modeling and Simulation. ADMOS 2007*, International Center for Numerical Methods in Engineering (CIMNE), 2007; 202–205.
- [73] Díez P, Ródenas JJ, Zienkiewicz OC. Equilibrated patch recovery error estimates : simple and accurate upper bounds of the error. *International Journal for Numerical Methods in Engineering* 2007; **69**(10):2075–2098.
- [74] Ródenas JJ, González-Estrada OA, Díez P, Fuenmayor FJ. Accurate recovery-based upper error bounds for the extended finite element framework. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(37-40):2607–2621.
- [75] Xiao QZ, Karihaloo BL. Improving the accuracy of XFEM crack tip fields using higher order quadrature and statically admissible stress recovery. *International Journal for Numerical Methods in Engineering* 2006; **66**(9):1378–1410.

- [76] Bordas SPA, Duflot M. Derivative recovery and a posteriori error estimate for extended finite elements. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**(35-36):3381–3399.
- [77] Bordas SPA, Duflot M, Le P. A simple error estimator for extended finite elements. *Communications in Numerical Methods in Engineering* 2008; **24**(11):961–971.
- [78] Duflot M, Bordas SPA. A posteriori error estimation for extended finite elements by an extended global recovery. *International Journal for Numerical Methods in Engineering* 2008; **76**:1123–1138.
- [79] González-Estrada OA, Ródenas JJ, Fuenmayor FJ, Chinesta F. Exact satisfaction of boundary equilibrium to obtain upper error bounds in energy norm using MLS recovery techniques. *Adaptive Modeling and Simulation. Proceedings of V ADMOS 2011*, Aubry D, Díez P, Tie B, Parés N (eds.), CINME: Paris, 2011.
- [80] Ródenas JJ, González-Estrada OA, Tarancón JE, Fuenmayor FJ. 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* 2008; **76**(4):545–571.
- [81] Ladevèze P, Leguillon D. Error estimate procedure in the finite element method and applications. *SIAM Journal on Numerical Analysis* 1983; **20**(3):485–509.
- [82] Ladevèze P, Chamoin L, Florentin E. A new non-intrusive technique for the construction of admissible stress fields in model verification. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(9-12):766–777.
- [83] Ladevèze P, Maunder EAW. A general method for recovering tractions equilibrating element. *Computer Methods in Applied Mechanics and Engineering* 1996; **137**(2):111–151.

- [84] Ladevèze P, Rougeot P. New advances on a posteriori error on constitutive relation in fe analysis. *Computer Methods in Applied Mechanics and Engineering* 1997; **150**(1-4):239–249.
- [85] Ladevèze P, Rougeot P, Blanchard P, Moreau JP. Local error estimators for finite element linear analysis. *Computer Methods in Applied Mechanics and Engineering* 1999; **176**(1-4):231–246.
- [86] Almeida Pereira OJB, Moitinho de Almeida JP, Maunder EAW. Adaptive methods for hybrid equilibrium finite element models. *Computational Methods in Applied Mechanics and Engineering* 1999; **176**:19–39.
- [87] Almeida Pereira OJB, Moitinho de Almeida JP. A posteriori error estimation for equilibrium finite elements in elastostatic problems. *Computer Assisted Mechanics and Engineering Sciences* 2001; **8**(2-3):439–453.
- [88] Moitinho de Almeida JP, Maunder EAW. Recovery of equilibrium on star patches using a partition of unity technique. *International Journal for Numerical Methods in Engineering* 2009; **79**:1493–1516.
- [89] Almeida Pereira OJB, Moitinho de Almeida JP. Dual adaptive finite element refinement for multiple local quantities in linear elastostatics. *International Journal for Numerical Methods in Engineering* 2010; .
- [90] Babuška I, Miller A. A feedback finite element method with a posteriori error estimation: Part I. The finite element method and some basic properties of the a posteriori error estimator. *Computer Methods in Applied Mechanics and Engineering* 1987; **61**(1):1–40.
- [91] Chua Sk, Wheeden RL. Estimates of the best constants for weighted Poincaré inequalities on convex domains. *London Mathematical Society* 2006; **93**(3):197–226.
- [92] Babuška I, Strouboulis T. *The finite element method and its reliability*. Oxford University Press, 2001.

- [93] Bank RE, Weiser A. Some A Posteriori Error Estimators For Elliptic Partial Differential Equations. *Mathematics of Computation* 1985; **44**:283–301.
- [94] Stein E. Finite Element Methods for Elasticity with Error-Controlled Discretization and Model Adaptivity. *Encyclopedia of Computational Mechanics*. 2007.
- [95] Pled F, Chamoin L, Ladevèze P. An enhanced method with local energy minimization for the robust a posteriori construction of equilibrated stress fields in finite element analyses. *Computational Mechanics* 2011; .
- [96] Parés N, Díez P, Huerta A. Subdomain-based flux-free a posteriori error estimators. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**(4-6):297–323.
- [97] Fraeijs de Veubeke B. Displacement and equilibrium models in the finite element method. *Stress Analysis*, Zienkiewicz OC, Holister GS (eds.). chap. 9, John Wiley & Sons, 1965.
- [98] Babuška I, Strouboulis T, Upadhyay CS. A model study of the quality of a posteriori error estimators for linear elliptic problems. Error estimation in the interior of patchwise uniform grids of triangles. *Computer Methods in Applied Mechanics and Engineering* 1994; **114**(3-4):307–378.
- [99] Babuška I, Strouboulis T, Upadhyay CS, Gangaraj SK, Copps K. Validation of a posteriori error estimators by numerical approach. *International Journal for Numerical Methods in Engineering* 1994; **37**(7):1073–1123.
- [100] Babuška I, Strouboulis T, Upadhyay CS. A model study of the quality of a posteriori error estimators for finite element solutions of linear elliptic problems, with particular reference to the behaviour near the boundary. *International Journal for Numerical Methods in Engineering* 1997; **40**(14):2521–2577.
- [101] Wiberg NE, Abdulwahab F. Patch recovery based on superconvergent derivatives and equilibrium. *International Journal for Numerical Methods in Engineering* 1993; **36**(16):2703–2724.

- [102] Aalto J. Built-in field equations for recovery procedures. *Computers & Structures* 1997; **64**(1-4):157–176.
- [103] Aalto J, Isoherranen H. An element by element recovery method with built-in field equations. *Computers & Structures* 1997; **64**(1-4):177–196.
- [104] Aalto J, Aman M. Polynomial representations for patch recovery procedures. *Computers & Structures* 1999; **73**(1-5):119–146.
- [105] Kvamsdal T, Okstad KM. Error estimation based on Superconvergent Patch Recovery using statically admissible stress fields. *International Journal for Numerical Methods in Engineering* 1998; **42**(3):443–472.
- [106] Boroomand B, Zienkiewicz OC. Recovery by equilibrium in patches (REP). *International Journal for Numerical Methods in Engineering* 1997; **40**(1):137–164.
- [107] Boroomand B, Zienkiewicz OC. An improved REP recovery and the effectivity robustness test. *International Journal for Numerical Methods in Engineering* 1997; **40**(17):3247–3277.
- [108] Nadal E, Bordas S, Ródenas JJ, Tarancón JE, Tur M. Accurate Stress Recovery for the Two-Dimensional Fixed Grid Finite Element Method. *Proceedings of the Tenth International Conference on Computational Structures Technology*, 2010; 1–20.
- [109] Timoshenko SP, Goodier JN. *Theory of Elasticity*. 2nd edn., McGraw-Hill: New York, 1951.
- [110] Tabbara M, Blacker T, Belytschko T. Finite element derivative recovery by moving least square interpolants. *Computer Methods in Applied Mechanics and Engineering* 1994; **117**(1-2):211–223.
- [111] Wiberg NE, Abdulwahab F. Error estimation with postprocessed finite element solutions. *Computers & Structures* 1997; **64**(1-4):113–137.
- [112] Szabó BA, Babuška I. *Finite Element Analysis*. John Wiley & Sons: New York, 1991.

- [113] Ródenas JJ, Giner E, Tarancón JE, González-Estrada OA. A recovery error estimator for singular problems using singular+smooth field splitting. *Fifth International Conference on Engineering Computational Technology*, Topping BHV, Montero G, Montenegro R (eds.), Civil-Comp Press: Stirling, Scotland, 2006.
- [114] Ladevèze P, Marin P, Pelle JP, Gastine JL. Accuracy and optimal meshes in finite element computation for nearly incompressible materials. *Computer Methods in Applied Mechanics and Engineering* 1992; **94**(3):303–315.
- [115] Coorevits P, Ladevèze P, Pelle JP. An automatic procedure with a control of accuracy for finite element analysis in 2D elasticity. *Computer Methods in Applied Mechanics and Engineering* 1995; **121**:91–120.
- [116] Li LY, Bettess P. Notes on mesh optimal criteria in adaptive finite element computations. *Communications in Numerical Methods in Engineering* 1995; **11**(11):911–915.
- [117] Nadal E, Ródenas JJ, Albelda J, Tur M, Tarancón JE, Fuenmayor FJ. Efficient Finite Element Methodology Based on Cartesian Grids: Application to Structural Shape Optimization. *Abstract and Applied Analysis* 2013; :1–19.
- [118] Paraschivoiu M, Peraire J, Patera AT. A posteriori finite element bounds for linear-functional outputs of elliptic partial differential equations. *Computer Methods in Applied Mechanics and Engineering* 1997; **150**(1-4):289–312.
- [119] Oden JT, Prudhomme S. Goal-oriented error estimation and adaptivity for the finite element method. *Computers & Mathematics with Applications* 2001; **41**(5-6):735–756.
- [120] Ródenas JJ. Goal Oriented Adaptivity: Una introducción a través del problema elástico lineal. *Technical Report*, CIMNE, PI274, Barcelona, Spain 2005.

- [121] González-Estrada OA, Ródenas JJ, Nadal E, Bordas SPA, Kerfriden P. Equilibrated patch recovery for accurate evaluation of upper error bounds in quantities of interest. *Adaptive Modeling and Simulation. Proceedings of V ADMOS 2011*, Aubry D, Díez P, Tie B, Parés N (eds.), CINME: Paris, 2011.
- [122] Verdugo F, Díez P, Casadei F. Natural quantities of interest in linear elastodynamics for goal oriented error estimation and adaptivity. *Adaptive Modeling and Simulation. Proceedings of V ADMOS 2011*, Aubry D, Díez P, Tie B, Parés N (eds.), CINME: Paris, 2011.
- [123] Maunder EAW, Moitinho de Almeida JP. Recovery of equilibrium on star patches from conforming finite elements with a linear basis. *International Journal for Numerical Methods in Engineering* 2012; **89**:1497–1526.
- [124] Ciarlet PG. *The finite element method for elliptic problems*. 1 edn., North-Holland publishing company, 1978.
- [125] Rüter M, Gerasimov T, Stein E. Goal-oriented explicit residual-type error estimates in XFEM. *Computational Mechanics* 2012; **52**(2):361–376.
- [126] Babushka I, Strouboulis T, Gangarajb SK. A posteriori estimation of the error in the recovered derivatives the finite element solution. *Computational method* 1997; **150**:369–396.
- [127] Storn R, Price K. Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces. *International Computer Science Institute, Berkley, CA, Tech. Rep. TR* 1995; :12–95.
- [128] Liu GR. MFree Shape Function Construction. *Mesh Free Methods. Moving beyond the Finite Element Method*. chap. 5, CRC Press: Boca Ratón, Florida, 2003; 693.

Appendix A

Displacement-based Moving Least Squares recovery with constrains. The MLS-CD technique

A.1 Introduction

In this thesis we have described several SPR-based recovery processes, such as the plain SPR, the SPR-C or the SPR-CD technique. However there exist other recovery processes which are more versatile in the sense that they do not require a mesh to be used and can therefore be directly applied to any type of BVP's solver, such as the meshless methods. In this appendix we will explain the details of a moving least squares recovery technique so-called MLS-CD which is a displacement based version of the one introduced by Ródenas *et. al.* [79]. The MLS-CX technique proposed in [79] had similar properties

to the ones obtained with the SPR-C technique. Analogously, the technique here proposed is similar to the SPR-CD technique. The MLS-CD includes the boundary equilibrium, internal equilibrium and Dirichlet constraints, to further increase the accuracy of both, the displacement and stress recovered fields.

A.2 The MLS-CD recovery technique

As introduced in Section 3, one way to estimate the error in energy norm of the FE solution is to use the ZZ error estimator (2.28). The use of the ZZ error estimator requires an improved stress field. During this work we have proposed to use the SPR-CD recovery process for that purpose, however we would like to briefly introduce this MLS-based recovery processes. The output of the MLS-CD recovery process will also be a recovered pair $(\mathbf{u}_{\text{MLS-CD}}^*, \boldsymbol{\sigma}_{\text{MLS-CD}}^*)$ as in the SPR-CD technique. The error in energy norm will be estimated as follows:

$$\mathcal{E}_{\text{MLS-CD}}^2 := \int_{\Omega} (\boldsymbol{\sigma}_{\text{MLS-CD}}^* - \boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} (\boldsymbol{\sigma}_{\text{MLS-CD}}^* - \boldsymbol{\sigma}^h) \, d\Omega \quad (\text{A.1})$$

A.2.1 MLS recovery process

The MLS technique is based on a weighted least squares formulation biased towards the test point where the value of the function has to be obtained. The technique considers a polynomial expansion for each one of the components of the recovered displacement field in the form:

$$u_k^*(\mathbf{x}) = \mathbf{p}(\mathbf{x}) \mathbf{a}_k(\mathbf{x}) \quad k = x, y \quad (\text{A.2})$$

where \mathbf{p} represents a polynomial basis of one degree higher than the FE interpolation degree and \mathbf{a}_k are unknown coefficients:

$$\mathbf{p}(\mathbf{x}) = \{1 \ x \ y \ x^2 \ xy \ y^2 \ \dots\} \quad (\text{A.3})$$

$$\mathbf{a}_k(\mathbf{x}) = \{a_{0_k}(\mathbf{x}) \ a_{1_k}(\mathbf{x}) \ a_{2_k}(\mathbf{x}) \ a_{3_k}(\mathbf{x}) \ a_{4_k}(\mathbf{x}) \ a_{5_k}(\mathbf{x}) \ \dots\}^T \quad (\text{A.4})$$

For 2D, $\mathbf{x} = (x, y)$, the expression to evaluate the recovered displacement field reads:

$$\mathbf{u}_{\text{MLS-CD}}^*(\mathbf{x}) = \begin{Bmatrix} u_x^*(\mathbf{x}) \\ u_y^*(\mathbf{x}) \end{Bmatrix} = \mathbf{P}(\mathbf{x})\mathbf{A}(\mathbf{x}) = \begin{bmatrix} \mathbf{p}(\mathbf{x}) & \mathbf{0} \\ \mathbf{0} & \mathbf{p}(\mathbf{x}) \end{bmatrix} \begin{Bmatrix} \mathbf{a}_x(\mathbf{x}) \\ \mathbf{a}_y(\mathbf{x}) \end{Bmatrix} \quad (\text{A.5})$$

Notice that the main difference between the SPR-CD and the MLS-CD technique is that in this case the coefficients \mathbf{a}_k are not constant. The format of (A.5), considering the two components of the displacement vector in a single equation, will result useful to impose the constraints required to satisfy the equilibrium equations and the Dirichlet boundary conditions.

Suppose that $\boldsymbol{\chi}$ is a point within $\Omega_{\mathbf{x}}$, being $\Omega_{\mathbf{x}}$ the support corresponding to a point \mathbf{x} defined by a distance (radius) $R_{\Omega_{\mathbf{x}}}$. The MLS approximation for each displacement component at $\boldsymbol{\chi}$ is given by:

$$u_k^*(\mathbf{x}, \boldsymbol{\chi}) = \mathbf{p}(\boldsymbol{\chi})\mathbf{a}_k(\mathbf{x}) \quad \forall \boldsymbol{\chi} \in \Omega_{\mathbf{x}}, \quad k = x, y \quad (\text{A.6})$$

To obtain the coefficients \mathbf{A} we have adopted the *Continuous Moving Least Squares Approximation* described in [128]. The following functional will be minimized:

$$\begin{aligned} J'(\mathbf{x}) &= \int_{\Omega_{\mathbf{x}}} W(\mathbf{x} - \boldsymbol{\chi}) \left[\mathbf{u}_{\text{MLS-CD}}^*(\mathbf{x}, \boldsymbol{\chi}) - \mathbf{u}^h(\boldsymbol{\chi}) \right]^2 d\boldsymbol{\chi} \\ &= \int_{\Omega_{\mathbf{x}}} W(\mathbf{x} - \boldsymbol{\chi}) \left[\mathbf{P}(\boldsymbol{\chi})\mathbf{A}(\mathbf{x}) - \mathbf{u}^h(\boldsymbol{\chi}) \right]^2 d\boldsymbol{\chi} \end{aligned} \quad (\text{A.7})$$

where W is the MLS weighting function, which in this work has been taken as the fourth-order spline, commonly used in the MLS related literature:

$$W(\mathbf{x} - \boldsymbol{\chi}) = \begin{cases} 1 - 6s^2 + 8s^3 - 3s^4 & \text{if } |s| \leq 1 \\ 0 & \text{if } |s| > 1 \end{cases} \quad (\text{A.8})$$

where s denotes the normalised distance function given by

$$s = \frac{\|\mathbf{x} - \boldsymbol{\chi}\|}{R_{\Omega_{\mathbf{x}}}} \quad (\text{A.9})$$

A.2.2 Satisfaction of the Dirichlet equation

We would like to enforce the satisfaction of the Dirichlet boundary conditions along the Dirichlet boundaries. In order to avoid the introduction of discontinuities in the recovered field, we have followed a *nearest point* approach. The nearest point approach can be used to impose the satisfaction of both, Dirichlet and Neumann boundary conditions. The approach introduces the exact satisfaction of the boundary conditions in a smooth continuous manner. As the constraint is smoothly introduced there is no jump when the support does not longer intersects Γ . For a point $\mathbf{x} \in \Omega$ whose support $\Omega_{\mathbf{x}}$ intersects the boundary Γ , the constraints are considered only in the closest points $\boldsymbol{\chi}_j \in \Gamma$ on the boundaries within the support of \mathbf{x} , as shown in Figure A.1. Note that we can have more than one *nearest point* for a given support, as is the case of a point \mathbf{x} approaching a corner where we take one point for each side of the corner (see Figure A.1). In this case, two different points have to be considered on the boundary to avoid jumps induced by the different boundary conditions when crossing the diagonal that bisects the corner.

The MLS functional expressed in its continuous version (see [128]) and incorporating the boundary constraints reads:

$$\begin{aligned} J''(\mathbf{x}) &= J'(\mathbf{x}) + \sum_{j=1}^{nDbc} W'(\mathbf{x} - \boldsymbol{\chi}_j) [u_k^*(\mathbf{x}, \boldsymbol{\chi}_j) - u_k^{ex}(\boldsymbol{\chi}_j)]^2 \\ &= J'(\mathbf{x}) + \sum_{j=1}^{nDbc} W'(\mathbf{x} - \boldsymbol{\chi}_j) [\mathbf{c}_k \mathbf{P}(\boldsymbol{\chi}_j) \mathbf{A}(\mathbf{x}) - u_k^{ex}(\boldsymbol{\chi}_j)]^2 \quad k = x, y \end{aligned} \quad (\text{A.10})$$

where $nDbc$ is the number of points $\boldsymbol{\chi}_j$ on the boundary where the known boundary constraints u_k^{ex} is the exact displacement field at Dirichlet bound-

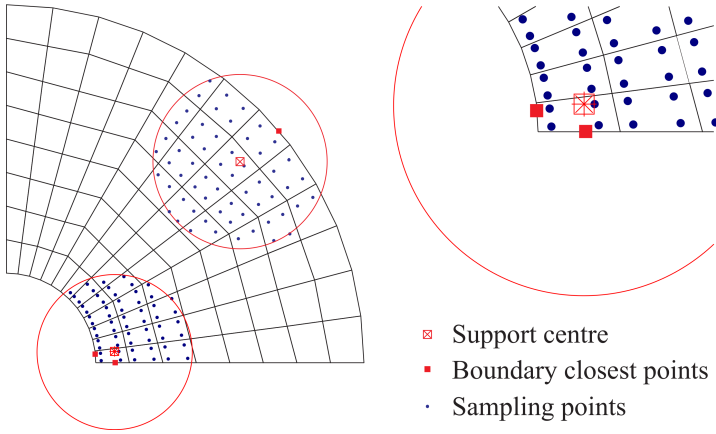


Figure A.1: MLS support with boundary conditions applied on the nearest boundary points.

aries and \mathbf{c}_k is the displacement component selector. In the previous equations W' is a weighting function defined as:

$$W'(\mathbf{x} - \boldsymbol{\chi}_j) = \frac{W(\mathbf{x} - \boldsymbol{\chi}_j)}{s} = \begin{cases} \frac{1}{s} - 6s + 8s^2 - 3s^3 & \text{if } |s| \leq 1 \\ 0 & \text{if } |s| > 1 \end{cases} \quad (\text{A.11})$$

This function has two main characteristics:

1. W' includes the weighting function W such that the term for the boundary constraint is introduced smoothly into the functional $J'(\mathbf{x})$. As a result, the recovered stress field will be continuous in Ω
2. W' also includes s^{-1} such that the weight of the boundary constraint in $J''(\mathbf{x})$ increases as we approach the boundary (when $\mathbf{x} \rightarrow \boldsymbol{\chi}_j$ $s \rightarrow 0$), therefore $\mathbf{u}_{\text{MLS-CD}}^*$ will tend to exactly satisfy boundary condition as $\mathbf{x} \rightarrow \boldsymbol{\chi}_j$ (see Figure A.2). Note that to estimate the error using the numerical integration in (A.1), the value of $\boldsymbol{\sigma}_{\text{MLS-CD}}^*$ is never evaluated on the boundary (where $s = 0$) because the integration points considered

are always inside the elements. If the recovered displacement field has to be evaluated along the boundary, we will replace this formulation by the standard Lagrange Multiplier technique as presented in Section 3.3 to impose the Dirichlet boundary conditions.

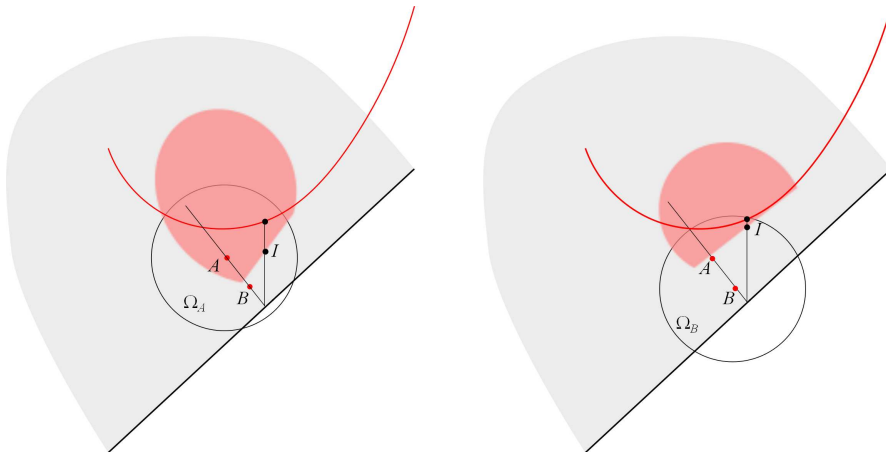


Figure A.2: Satisfaction of boundary conditions. We observe two zones corresponding to the support of points A and B , Ω_A and Ω_B , with B closer to the boundary. The red line indicates the exact value at the boundary imposed (displacement or traction). We observe that when the support point gets closer to the boundary ($A \rightarrow B$), the surface (shaded red area) represents more accurately the exact value at the boundary. When the support point is on the boundary the recovered field (stress or displacement) will represent the boundary conditions exactly.

A.2.3 Satisfaction of the equilibrium equation

The boundary equilibrium equation must be satisfied at each point along the contour. In [73, 71, 74], where an SPR-based technique was used, the authors enforced the satisfaction of the boundary conditions in patches along the boundary using Lagrange Multipliers to impose the appropriate constraints between the unknown coefficients to be evaluated. In the case of the Dirichlet boundary conditions, this approach produces discontinuities in a MLS formulation as we move from a support fully in the interior of the domain to a

support intersecting the boundary. As previously mentioned we have followed a *nearest point* approach to introduces the exact satisfaction of the boundary equilibrium equation in a smooth continuous manner.

Let us express the stress vector $\boldsymbol{\sigma}_{\text{MLS-CD}}^*(\mathbf{x}, \boldsymbol{\chi})$ in a coordinate system (x', y') aligned with the contour at $\boldsymbol{\chi}_j$ such that x' is the outward normal vector, rotated an angle α with respect to x :

$$\boldsymbol{\sigma}_{\text{MLS-CD}}^{*'}(\mathbf{x}, \boldsymbol{\chi}) = \mathbf{R}(\alpha)\boldsymbol{\sigma}_{\text{MLS-CD}}^*(\mathbf{x}) \quad (\text{A.12})$$

where \mathbf{R} is the stress rotation matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_{x'x'} \\ \mathbf{r}_{y'y'} \\ \mathbf{r}_{x'y'} \end{bmatrix} = \begin{bmatrix} \cos^2 \alpha & \sin^2 \alpha & \sin(2\alpha) \\ \sin^2 \alpha & \cos^2 \alpha & -\sin(2\alpha) \\ -\sin(2\alpha)/2 & \sin(2\alpha)/2 & \cos(2\alpha) \end{bmatrix} \quad (\text{A.13})$$

The MLS functional expressed in its continuous version and incorporating the boundary constraints reads:

$$J(\mathbf{x}) = J''(\mathbf{x}) + \sum_{j=1}^{nNbc} W'(\mathbf{x} - \boldsymbol{\chi}_j) [\sigma_k^{*'}(\mathbf{x}, \boldsymbol{\chi}_j) - \sigma_k^{ex}(\boldsymbol{\chi}_j)]^2 \quad (\text{A.14})$$

where $nNbc$ is the number of nearest points along the boundaries with the support of x .

In order to add this constrain we need an expression for the stress field $\sigma_k^{*'}(\mathbf{x}, \boldsymbol{\chi})$ in the support. First we define the strain field components as follows:

$$\begin{aligned} \varepsilon_{km}^*(\mathbf{x}, \boldsymbol{\chi}) &= \frac{1}{2} \frac{\partial u_k^*(\mathbf{x}, \boldsymbol{\chi})}{\partial m} + \frac{1}{2} \frac{\partial u_m^*(\mathbf{x}, \boldsymbol{\chi})}{\partial k} \\ &= \frac{1}{2} \left[\frac{\partial \mathbf{p}(\boldsymbol{\chi})}{\partial m} \mathbf{a}_k(\mathbf{x}) + \mathbf{p}(\boldsymbol{\chi}) \frac{\partial \mathbf{a}_k(\mathbf{x})}{\partial m} \right] \\ &\quad + \frac{1}{2} \left[\frac{\partial \mathbf{p}(\boldsymbol{\chi})}{\partial k} \mathbf{a}_m(\mathbf{x}) + \mathbf{p}(\boldsymbol{\chi}) \frac{\partial \mathbf{a}_m(\mathbf{x})}{\partial k} \right] \end{aligned} \quad (\text{A.15})$$

where $k = x, y$ and $m = x, y$ indicate the components of the strain field, the values of $\mathbf{a}_k(\mathbf{x})$ and $\mathbf{a}_m(\mathbf{x})$ and their derivatives are unknown. The stress field

is evaluated as usual, using the Hook's law matrix \mathbf{D} :

$$\boldsymbol{\sigma}_{\text{MLS-CD}}^*(\mathbf{x}, \boldsymbol{\chi}) = \mathbf{D} \begin{Bmatrix} \varepsilon_{xx}^*(\mathbf{x}, \boldsymbol{\chi}) \\ \varepsilon_{yy}^*(\mathbf{x}, \boldsymbol{\chi}) \\ 2\varepsilon_{xy}^*(\mathbf{x}, \boldsymbol{\chi}) \end{Bmatrix} \quad (\text{A.16})$$

For the evaluation of the derivatives of the displacement field needed to evaluate the strain field we use the following methodology. Evaluating $\partial J'' / \partial \mathbf{A} = 0$ results in the following linear system of equations that could be used to evaluate \mathbf{A}

$$\mathbf{M}(\mathbf{x})\mathbf{A}(\mathbf{x}) = \mathbf{G}(\mathbf{x}) \quad (\text{A.17})$$

$$\begin{aligned} \mathbf{M}(\mathbf{x}) &= \int_{\Omega_{\mathbf{x}}} W(\mathbf{x} - \boldsymbol{\chi}) \mathbf{P}^T(\boldsymbol{\chi}) \mathbf{P}(\boldsymbol{\chi}) d\boldsymbol{\chi} + \sum_{j=1}^{nDbc} W'(\mathbf{x} - \boldsymbol{\chi}_j) \mathbf{P}^T(\boldsymbol{\chi}_j) \mathbf{c}_k^T \mathbf{c}_k \mathbf{P}(\boldsymbol{\chi}_j) \\ \mathbf{G}(\mathbf{x}) &= \int_{\Omega_{\mathbf{x}}} W(\mathbf{x} - \boldsymbol{\chi}) \mathbf{P}^T(\boldsymbol{\chi}) \boldsymbol{\sigma}^h(\boldsymbol{\chi}) d\boldsymbol{\chi} + \sum_{j=1}^{nDbc} W'(\mathbf{x} - \boldsymbol{\chi}_j) \mathbf{P}^T(\boldsymbol{\chi}_j) \mathbf{c}_k^T u_k^{ex}(\boldsymbol{\chi}_j) \end{aligned} \quad (\text{A.18})$$

As an example, we now show how to evaluate the derivatives in the x direction. Differentiating (A.17) with respect to x we obtain:

$$\frac{\partial \mathbf{M}(\mathbf{x})}{\partial x} \mathbf{A}(\mathbf{x}) + \mathbf{M}(\mathbf{x}) \frac{\partial \mathbf{A}(\mathbf{x})}{\partial x} = \frac{\partial \mathbf{G}(\mathbf{x})}{\partial x} \quad (\text{A.19})$$

then,

$$\frac{\partial \mathbf{A}(\mathbf{x})}{\partial x} = \begin{Bmatrix} \frac{\partial a_x(\mathbf{x})}{\partial x} \\ \frac{\partial a_y(\mathbf{x})}{\partial x} \end{Bmatrix} = \mathbf{M}^{-1}(\mathbf{x}) \left[\frac{\mathbf{G}(\mathbf{x})}{\partial x} - \frac{\partial \mathbf{M}(\mathbf{x})}{\partial x} \mathbf{A}(\mathbf{x}) \right] \quad (\text{A.20})$$

Using this definition for the derivatives of the coefficient vector $\mathbf{A}(\mathbf{x})$ we evaluate the strain field defined in (A.15) and therefore the stress field can be

evaluated as follows:

$$\begin{aligned}
\boldsymbol{\sigma}^*_{\text{MLS-CD}}(\mathbf{x}, \boldsymbol{\chi}) &= \\
\mathbf{D} &\left\{ \begin{aligned} &\left[\begin{array}{cc} \frac{\partial \mathbf{P}(\boldsymbol{\chi})}{\partial x} & 0 \\ 0 & \frac{\partial \mathbf{P}(\boldsymbol{\chi})}{\partial y} \end{array} \right] \mathbf{A}(\mathbf{x}) - \left[\begin{array}{cc} \mathbf{P}(\boldsymbol{\chi}) & 0 \\ 0 & 0 \\ 0 & \mathbf{P}(\boldsymbol{\chi}) \end{array} \right] \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{M}(\mathbf{x})}{\partial x} \mathbf{A}(\mathbf{x}) \\ &- \left[\begin{array}{cc} 0 & 0 \\ 0 & \mathbf{P}(\boldsymbol{\chi}) \\ \mathbf{P}(\boldsymbol{\chi}) & 0 \end{array} \right] \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{M}(\mathbf{x})}{\partial y} \mathbf{A}(\mathbf{x}) \\ &+ \left. \left[\begin{array}{cc} \mathbf{P}(\boldsymbol{\chi}) & 0 \\ 0 & 0 \\ 0 & \mathbf{P}(\boldsymbol{\chi}) \end{array} \right] \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{G}(\mathbf{x})}{\partial x} + \left[\begin{array}{cc} 0 & 0 \\ 0 & \mathbf{P}(\boldsymbol{\chi}) \\ \mathbf{P}(\boldsymbol{\chi}) & 0 \end{array} \right] \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{G}(\mathbf{x})}{\partial y} \right\} \\
&= \mathbf{D} \left\{ \begin{aligned} &\left[\mathbf{L}\mathbf{P}(\boldsymbol{\chi}) - \mathbf{P}_x^*(\boldsymbol{\chi})\mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{M}(\mathbf{x})}{\partial x} - \mathbf{P}_y^*(\boldsymbol{\chi})\mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{M}(\mathbf{x})}{\partial y} \right] \mathbf{A}(\mathbf{x}) \\ &+ \mathbf{P}_x^*(\boldsymbol{\chi})\mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{G}(\mathbf{x})}{\partial x} + \mathbf{P}_y^*(\boldsymbol{\chi})\mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{G}(\mathbf{x})}{\partial y} \end{aligned} \right\} \\
&= \mathbf{H}(\mathbf{x}, \boldsymbol{\chi})\mathbf{A}(\mathbf{x}) + \mathbf{F}(\mathbf{x}, \boldsymbol{\chi})
\end{aligned} \tag{A.21}$$

where the partial derivatives of \mathbf{M} and \mathbf{G} with respect, for example, to x are

$$\begin{aligned}
\frac{\partial \mathbf{M}}{\partial x} &= \int_{\Omega_x} \frac{\partial W(\mathbf{x} - \boldsymbol{\chi})}{\partial x} \mathbf{P}^T(\boldsymbol{\chi}) \mathbf{P}(\boldsymbol{\chi}) \, d\boldsymbol{\chi} + \\
&\quad \sum_{j=1}^{nDbc} \frac{\partial W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial x} \mathbf{P}^T(\boldsymbol{\chi}_j) \mathbf{c}_k^T \mathbf{c}_k \mathbf{P}(\boldsymbol{\chi}_j)
\end{aligned} \tag{A.22}$$

$$\begin{aligned}
\frac{\partial \mathbf{G}}{\partial x} &= \int_{\Omega_x} \frac{\partial W(\mathbf{x} - \boldsymbol{\chi})}{\partial x} \mathbf{P}^T(\boldsymbol{\chi}) \boldsymbol{\sigma}^h(\boldsymbol{\chi}) \, d\boldsymbol{\chi} + \\
&\quad \sum_{j=1}^{nDbc} \frac{\partial W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial x} \mathbf{P}^T(\boldsymbol{\chi}_j) \mathbf{c}_k^T u_k^{ex}(\boldsymbol{\chi}_j)
\end{aligned} \tag{A.23}$$

where, differentiating (A.8) and (A.11):

$$\frac{\partial W(\mathbf{x} - \boldsymbol{\chi})}{\partial x} = \frac{\partial W(\mathbf{x} - \boldsymbol{\chi})}{\partial s} \frac{\partial s}{\partial x} \tag{A.24}$$

$$\frac{\partial W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial x} = \frac{\partial W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial s} \frac{\partial s}{\partial x} \quad (\text{A.25})$$

In these equations $\partial s / \partial x$ can be obtained from (A.9) or, alternatively, from (A.36) for the case shown in the next Section.

Finally the functional (A.14) can be rewritten as follows:

$$J(\mathbf{x}) = J''(\mathbf{x}) + \sum_{j=1}^{nNbc} W'(\mathbf{x} - \boldsymbol{\chi}_j) [\mathbf{r}_{k'} [\mathbf{H}(\mathbf{x}, \boldsymbol{\chi}_j) \mathbf{A}(\mathbf{x}) + \mathbf{F}(\mathbf{x}, \boldsymbol{\chi}_j)] - \sigma_{k'}^{ex}(\boldsymbol{\chi}_j)]^2 \quad (\text{A.26})$$

Notice that when evaluating the derivatives at this step the full functional $J(\mathbf{x})$ is not considered (as it is being built). Then the boundary equilibrium is not strictly fulfilled, although it has a high level of accuracy. In practice can be considered that the boundary equilibrium is satisfied.

A.2.4 Satisfaction of the internal equilibrium equation

In addition to the enforcement of the Dirichlet boundary conditions and the boundary equilibrium, we will also consider the satisfaction of the internal equilibrium equation using the Lagrange Multipliers technique. Thus, we will try to enforce the recovered stress field $\boldsymbol{\sigma}_{\text{MLS-CD}}^*$ to satisfy the internal equilibrium equation:

$$\mathbf{L}^T \boldsymbol{\sigma}_{\text{MLS-CD}}^* + \mathbf{b} = \mathbf{0} \quad (\text{A.27})$$

In order to build the constrain equations we need to evaluate the first derivatives of expression (A.21). The following equation represents the derivate in

x direction for the stress field.

$$\begin{aligned}
& \frac{\partial \sigma_{\text{MLS-CD}}^*}{\partial x} \\
&= \mathbf{D} \left\{ \left[\frac{\partial \mathbf{LP}(\boldsymbol{\chi})}{\partial x} \right. \right. \\
&\quad - \frac{\partial \mathbf{P}_x^*(\boldsymbol{\chi})}{\partial x} \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{M}(\mathbf{x})}{\partial x} - \mathbf{P}_x^*(\boldsymbol{\chi}) \frac{\partial \mathbf{M}^{-1}(\mathbf{x})}{\partial x} \frac{\partial \mathbf{M}(\mathbf{x})}{\partial x} - \mathbf{P}_x^*(\boldsymbol{\chi}) \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial^2 \mathbf{M}(\mathbf{x})}{\partial x^2} \\
&\quad \left. \left. - \frac{\partial \mathbf{P}_y^*(\boldsymbol{\chi})}{\partial x} \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{M}(\mathbf{x})}{\partial x} - \mathbf{P}_x^*(\boldsymbol{\chi}) \frac{\partial \mathbf{M}^{-1}(\mathbf{x})}{\partial x} \frac{\partial \mathbf{M}(\mathbf{x})}{\partial x} - \mathbf{P}_x^*(\boldsymbol{\chi}) \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial^2 \mathbf{M}(\mathbf{x})}{\partial yx} \right] \right. \\
&\mathbf{A}(\mathbf{x}) \\
&\quad + \left[\mathbf{LP}(\boldsymbol{\chi}) - \mathbf{P}_x^*(\boldsymbol{\chi}) \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{M}(\mathbf{x})}{\partial x} - \mathbf{P}_y^*(\boldsymbol{\chi}) \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{M}(\mathbf{x})}{\partial y} \right] \frac{\partial \mathbf{A}(\mathbf{x})}{\partial x} \\
&\quad + \frac{\partial \mathbf{P}_x^*(\boldsymbol{\chi})}{\partial x} \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{G}(\mathbf{x})}{\partial x} + \mathbf{P}_x^*(\boldsymbol{\chi}) \frac{\partial \mathbf{M}^{-1}(\mathbf{x})}{\partial x} \frac{\partial \mathbf{G}(\mathbf{x})}{\partial x} - \mathbf{P}_x^*(\boldsymbol{\chi}) \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial^2 \mathbf{M}(\mathbf{x})}{\partial x^2} \\
&\quad \left. + \frac{\partial \mathbf{P}_y^*(\boldsymbol{\chi})}{\partial x} \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{G}(\mathbf{x})}{\partial x} - \mathbf{P}_x^*(\boldsymbol{\chi}) \frac{\partial \mathbf{M}^{-1}(\mathbf{x})}{\partial x} \frac{\partial \mathbf{G}(\mathbf{x})}{\partial x} - \mathbf{P}_x^*(\boldsymbol{\chi}) \mathbf{M}^{-1}(\mathbf{x}) \frac{\partial^2 \mathbf{G}(\mathbf{x})}{\partial yx} \right\} \\
&\hspace{15em} (\text{A.28})
\end{aligned}$$

Equation (A.28) has new terms that have to be evaluated. To obtain this we can consider now functional $J(\mathbf{x})$ in (A.26) because we have already evaluated the terms corresponding to boundary equilibrium.

- the first derivative of the \mathbf{M} matrix:

$$\begin{aligned}
\frac{\partial \mathbf{M}}{\partial k} &= \int_{\Omega_x} \frac{\partial W(\mathbf{x} - \boldsymbol{\chi})}{\partial k} \mathbf{P}^T(\boldsymbol{\chi}) \mathbf{P}(\boldsymbol{\chi}) \, d\boldsymbol{\chi} \\
&\quad + \sum_{j=1}^{nDbc} \frac{\partial W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial k} \mathbf{P}^T(\boldsymbol{\chi}_j) \mathbf{c}_k^T \mathbf{c}_k \mathbf{P}(\boldsymbol{\chi}_j) \\
&\quad + \sum_{j=1}^{nNbc} \frac{\partial^2 W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial k} \mathbf{H}^T(\mathbf{x}, \boldsymbol{\chi}_j) \mathbf{r}_{k'}^T \mathbf{r}_{k'} \mathbf{H}(\mathbf{x}, \boldsymbol{\chi}_j)
\end{aligned} \tag{A.29}$$

- the first derivative of the \mathbf{G} matrix:

$$\begin{aligned} \frac{\partial \mathbf{G}}{\partial x} &= \int_{\Omega_x} \frac{\partial W(\mathbf{x} - \boldsymbol{\chi}_l)}{\partial x} \mathbf{P}^T(\boldsymbol{\chi}_l) \boldsymbol{\sigma}^h(\boldsymbol{\chi}_l) \, d\boldsymbol{\chi} \\ &+ \sum_{j=1}^{nDbc} \frac{\partial W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial x} \mathbf{P}^T(\boldsymbol{\chi}_j) \mathbf{c}_k^T u_k^{ex}(\boldsymbol{\chi}_j) \\ &+ \sum_{j=1}^{nNbc} \frac{\partial^2 W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial k} \mathbf{H}^T(\mathbf{x}, \boldsymbol{\chi}_j) \mathbf{r}_{k'}^T \sigma_k^{'ex}(\boldsymbol{\chi}_j) \end{aligned} \quad (\text{A.30})$$

- the derivative of the inverse of the \mathbf{M} matrix:

$$\frac{\partial \mathbf{M}^{-1}(\mathbf{x})}{\partial k} = -\mathbf{M}^{-1}(\mathbf{x}) \frac{\partial \mathbf{M}(\mathbf{x})}{\partial k} \mathbf{M}^{-1}(\mathbf{x}) \quad (\text{A.31})$$

- the second derivative of the \mathbf{M} matrix:

$$\begin{aligned} \frac{\partial^2 \mathbf{M}(\mathbf{x})}{\partial km} &= \int_{\Omega_x} \frac{\partial^2 W(\mathbf{x} - \boldsymbol{\chi})}{\partial km} \mathbf{P}^T(\boldsymbol{\chi}) \mathbf{P}(\boldsymbol{\chi}) \, d\boldsymbol{\chi} \\ &+ \sum_{j=1}^{nbc} \frac{\partial^2 W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial km} \mathbf{P}^T(\boldsymbol{\chi}_j) \mathbf{c}_k^T \mathbf{c}_k \mathbf{P}(\boldsymbol{\chi}_j) \\ &+ \sum_{j=1}^{nbc} \frac{\partial^2 W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial km} \mathbf{H}^T(\mathbf{x}, \boldsymbol{\chi}_j) \mathbf{r}_{k'}^T \mathbf{r}_{k'} \mathbf{H}(\mathbf{x}, \boldsymbol{\chi}_j) \end{aligned} \quad (\text{A.32})$$

- the second derivative of the \mathbf{G} vector:

$$\begin{aligned} \frac{\partial^2 \mathbf{G}(\mathbf{x})}{\partial km} &= \int_{\Omega_x} \frac{\partial W(\mathbf{x} - \boldsymbol{\chi})}{\partial x} \mathbf{P}^T(\boldsymbol{\chi}) \boldsymbol{\sigma}^h(\boldsymbol{\chi}) \, d\boldsymbol{\chi} \\ &+ \sum_{j=1}^{nbc} \frac{\partial W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial x} \mathbf{P}^T(\boldsymbol{\chi}_j) \mathbf{c}_k^T u_k^{ex}(\boldsymbol{\chi}_j) \\ &+ \sum_{j=1}^{nbc} \frac{\partial^2 W'(\mathbf{x} - \boldsymbol{\chi}_j)}{\partial km} \mathbf{H}^T(\mathbf{x}, \boldsymbol{\chi}_j) \mathbf{r}_{k'}^T \sigma_k^{'ex}(\boldsymbol{\chi}_j) \end{aligned} \quad (\text{A.33})$$

The use of the Lagrange Multipliers technique to impose the equilibrium constraint (A.27) in (A.26) leads to the following system of equations:

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ \mathbf{B} \end{bmatrix} \quad (\text{A.34})$$

where \mathbf{C} and \mathbf{B} are the terms used to impose the constraint equations and $\boldsymbol{\lambda}$ is the vector of Lagrange Multipliers.

However, in (A.28), it was assumed that \mathbf{A} is evaluated solving $\mathbf{MA} = \mathbf{G}$, although, operating by blocks in (A.34) the following system of equations is obtained:

$$\mathbf{MA} + \mathbf{C}^T \boldsymbol{\lambda} = \mathbf{G} \quad (\text{A.35})$$

Hence, in the formulation proposed in this paper we have neglected the term $\mathbf{C}^T \boldsymbol{\lambda}$ when evaluating the partial derivatives of \mathbf{A} . Evidently, this implies that the internal equilibrium equation is not fully satisfied, leading to a nearly exact satisfaction of the internal equilibrium equation as in the SPR-CD technique.

A.2.5 Visibility

For problems with re-entrant corners a visibility criterion is used to modify the normalised distance s in (A.9). The standard weight function depends on the distance between the central point of the support and the sampling points, decreasing as the sampling points are located farther from the center [76].

Consider a domain with a re-entrant corner as shown in Figure A.3. The value of the weight function for a sampling point $\boldsymbol{\chi}_l$, considering a centre point \mathbf{x} whose support contains the singularity at $\boldsymbol{\chi}_\lambda$, diminishes with the visibility of $\boldsymbol{\chi}_l$ from \mathbf{x} such that, for points that cannot be directly viewed from \mathbf{x} , instead of (A.9), the following equation is used

$$s = \frac{\|\mathbf{x} - \boldsymbol{\chi}_\lambda\| + \|\boldsymbol{\chi}_l - \boldsymbol{\chi}_\lambda\|}{R_{\Omega_{\mathbf{x}}}} \quad (\text{A.36})$$

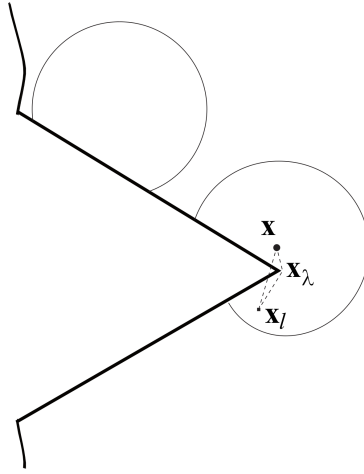


Figure A.3: Domain with re-entrant corner.

A.2.6 Stress splitting for singular problems

It is well known that smoothing techniques perform badly when the solution contains a singularity. In Section 3.3 a technique that decomposes the stress field in singular and smooth parts in order to improve the accuracy of SPR-based error estimators was proposed. The exact solution $(\mathbf{u}, \boldsymbol{\sigma})$ corresponding to a singular problem can be expressed as the contribution of a smooth pair, $(\mathbf{u}_{smo}, \boldsymbol{\sigma}_{smo})$, and a singular pair, $(\mathbf{u}_{sing}, \boldsymbol{\sigma}_{sing})$, then we will apply the same ideas than the ones presented in Section 3.3 for the SPR-CD.

A.3 Concluding remarks

In this Appendix we have introduced a displacement recovery procedure based on an equilibrated moving least squares approach applied to the cgFEM code, the MLS-CD technique. The MLS-CD also includes the splitting procedure presented in Section 3.3 for singular fields and also a visibility criterion for

re-entrant corners. The MLS-CD technique provides an improved continuous recovered pair $(\mathbf{u}_{\text{MLS-CD}}^*, \boldsymbol{\sigma}_{\text{MLS-CD}}^*)$ over all the domain which can be used in combination with the ZZ error estimator to obtain extremely accurate error estimators in energy norm. This technique has the advantage with respect to the SPR-CD that can be directly applied to meshless methods due to the MLS-CD only uses a cloud of sampling points (integration points) and does not attend to the mesh structure. However the computational cost is considerably higher than the one required by the SPR-CD. Therefore the improvement in accuracy does not justify its use when the mesh structure is already present as in the cgFEM technique.

