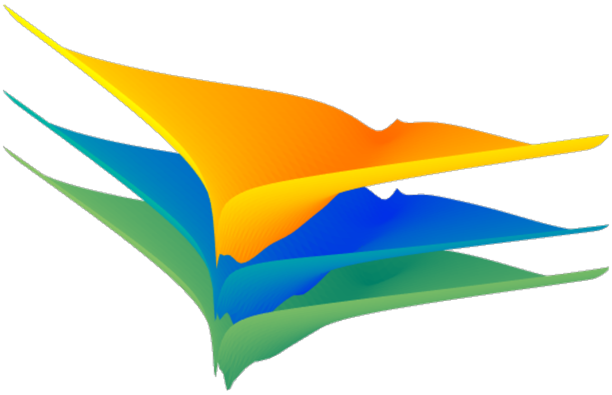
UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

UNIVERSITAT POLITÈCNICA DE CATALUNYA



Error Estimation and Adaptivity for PGD Solutions

Jonatha Oliveira de Matos Reis

Supervisor: Professor Doctor José Paulo Moitinho de Almeida
Co-Supervisor: Professor Doctor Pedro Díez
Co-Supervisor: Professor Doctor Sergio Zlotnik

Thesis specifically prepared to obtain the PhD Degree in Computational Engineering

DRAFT

Lisbon, July 2020

“ There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable. There is another theory mentioned, which states that this has already happened.”

The Hitchhiker’s Guide to the Galaxy

Acknowledgements

It is my greatest pleasure to begin my acknowledgements by thanking prof. Moitinho, from Instituto Superior Técnico. It was truly a privilege and a pleasure working by your side (and by Zoom) throughout these years. It was countless hours of supervision and valuable lessons, in which probably half these hours were spent fixing plots positions, and I learned a lot from you. It goes without saying that I will be forever thankful for your support. May all our problems be solved by a factor of two, a Jacobian, or both.

My sincere thanks for my co-supervisors Pedro Díez and Sergio Zlotnik, from Universitat Politècnica de Catalunya. It was an honor working alongside you and having such an incredible team of researches around. Thank you for receiving me in Barcelona and for your insights on the research, which were vital to bring order for the chaos of equations and indices.

During these PhD years, I was lucky enough to be in the company of amazing colleagues that made my time in Portugal and Spain a lot easier. I want to thank my office partners in Portugal Arash, Pablo, Bruno, Mahendra, my fellow PhD struggles in Spain Christina, Fabiola, Beatriz, Giacomo, Ygee, Davide and all other amazing people I've met during this journey. I reserve a special thanks for my forever lunch partner in Portugal prof. Carlos Tiago, who always pushed in the right direction, sometimes literally.

I also want to thank my family for their support and never-ending patience and for paving the path that lead me to where I am right now. If I got this far, I owe this to your effort and trust on me. It is never easy being apart but know that you were always with me in my heart.

A special thanks for her that has been with me along all this way. My friend, partner and wife, who were there for all the crazy. It is truly amazing the places we end up only by making plans for something else completely different. It has been an incredible journey this far and I can only imagine where it will lead us next.

So long and thanks for all the fish...

Abstract

A significant part of the current research efforts in computational mechanics is focused on analyzing and handling large amounts of data, exploring high dimensional parametric spaces and providing answers to increasingly complex problems in short or real-time. This alludes to concepts (like digital twins) and technologies (like machine learning), methodologies to be considered in combination with classical computational models. Reduced Order Models (ROM) contribute to address these challenges by reducing the number of degrees of freedom of the models, suppressing redundancies in the description of the system to be modeled and simplifying the representation of the mathematical objects quantifying the physical magnitudes.

Among these reduced order models, the Proper Generalized Decomposition (PGD) can be a powerful tool, as it provides solutions to parametric problems, without being affected by the ”curse of dimensionality”, providing explicit expressions, computed a priori, of the parametric solution, making it well suited to provide real-time responses. The PGD is a well-established reduced order method, but assessing the accuracy of its solutions is still a relevant challenge, particularly when seeking guaranteed bounds of the error of the solutions it provides.

There are several approaches to analyze the errors of approximate solutions, but the only way that provides computable and guaranteed error bounds is by applying dual analysis. The idea behind dual analysis is to use a pair of complementary solutions (one that is compatible, and the other equilibrated) for a specified problem and to use the difference between these solutions, which bounds their errors. Dual analysis is also effective to drive mesh adaptivity refinement processes, as it provides information of the contribution of the elements to the error, either in a global or in a local framework.

In this work we deal with finite element solutions for solid mechanics problems, computing compatible and equilibrated PGD solutions, using them in the context of dual analysis. The PGD approximations are obtained with an algebraic approach, leading to separable solutions that can be manipulated for an efficient computational implementation. We use these solutions to obtain global and local error bounds and use these bounds to drive an adaptivity process. The meshes obtained through these refinements provide solutions with errors significantly lower than those obtained using a uniform refinement.

Resumo

O cenário atual da pesquisa em mecânica computacional está focado em analisar grandes quantidades de dados, explorar grandes espaços paramétricos e fornecer respostas a problemas cada vez mais complexos em tempo (quase) real. Podemos relacionar esses desafios a conceitos como gêmeos digitais, e tecnologias como aprendizado de máquinas, que se combinam com modelos computacionais clássicos. Os Modelos de Ordem Reduzida (MOR) contribuem para solucionar esses problemas através de uma redução dos graus de liberdade destes modelos, suprimindo redundâncias na descrição dos sistemas e simplificando a sua representação matemática.

Dentre estes modelos de ordem reduzida a PGD (de *Proper Generalized Decomposition*, em português Decomposição Generalizada Característica) pode ser uma poderosa ferramenta, fornecendo soluções paramétricas que não são afetadas pelo “maldição da dimensionalidade”. A PGD provê expressões explícitas das soluções paramétricas, sendo estas calculadas a priori, o que torna este método vantajoso quando se buscam soluções em tempo real. A PGD é uma metodologia bem estabelecida, mas avaliar a precisão das aproximações obtidas ainda é uma questão relevante, particularmente quando se procuram majorantes do erro das suas soluções, que sejam garantidos.

Existem diversas maneiras de analisar os erros das aproximações que obtemos, mas a única forma de obter soluções com majorantes dos erros garantidos é aplicando uma análise dual. A ideia por trás da análise dual é utilizar um par de soluções complementares (uma solução compatível e uma solução equilibrada) para um problema específico e usar a diferença entre essas soluções. Ela é também efetiva para direcionar processos de refinamentos de malhas, já que é possível obter a contribuição dos elementos para o erro tanto num contexto local quando global.

Neste trabalho lidamos com soluções de elementos finitos para problemas de mecânica dos sólidos, calculando soluções compatíveis e equilibradas para soluções PGD e usando estas na análise dual do erro. As aproximações por PGD são obtidas utilizando um método algébrico, o que resulta em soluções separáveis que podem ser manipuladas para uma eficiente implementação computacional. Utilizamos estas soluções para obter majorantes locais e globais para o erro e usamos esses majorantes para direcionar processos adaptativos ótimos. As malhas obtidas através desse processo de refinamento geram soluções com erros significativamente menores que aquelas obtidas através de um refinamento uniforme.

Resumen

Una parte importante de los esfuerzos de investigación actuales en mecánica computacional se centra en analizar y manejar grandes cantidades de datos, explorar espacios paramétricos de alta dimensión, y proporcionar respuestas a problemas cada vez más complejos en un tiempo breve o en tiempo real. Esto alude a conceptos (como gemelos digitales) y tecnología (como aprendizaje automático), metodologías a ser consideradas en combinación con modelos computacionales clásicos. Los modelos de orden reducido (ROM) contribuyen a abordar estos desafíos al reducir el número de grados de libertad de los modelos, suprimir redundancias en la descripción del sistema a modelar, y simplificar la representación de los objetos matemáticos.

Entre estos modelos de orden reducido, la PGD (del inglés Porper Generalized Decomposition) puede ser una herramienta poderosa, ya que proporciona soluciones a problemas paramétricos, sin verse afectada por "el problema de la dimensionalidad". La PGD proporciona expresiones explícitas, calculadas a priori, de la solución paramédica, por lo que es adecuado para proporcionar. La PGD es un método de reducción de orden bien establecido, pero evaluar la precisión de sus soluciones sigue siendo un desafío relevante, en particular cuando se buscan límites garantizando el error de las soluciones que proporciona.

Existen varios enfoques para analizar los errores de las soluciones aproximadas, pero la única forma de proporcionar límites de error computables y garantizados es mediante la aplicación de un análisis dual. La idea del análisis dual es utilizar un par de soluciones complementarias(una que sea compatible y la otra equilibrada) para un problema específico. La diferencia entre las dos soluciones es lo que limitará el error. El análisis dual también es efectivo para impulsar procesos de refinamientos de adaptabilidad de malla, ya que proporciona información sobre la contribución de los elementos al error, sea en un marco global o local.

En este trabajo tratamos con soluciones de elementos finitos para problemas de mecánica de sólidos , calculando soluciones PGD compatibles y equilibradas, utilizándolas en contexto análisis dual. Las aproximaciones PGD se obtienen con un enfoque algebraico, lo que lleva a soluciones separables para obtener límites en el error globales como locales, y los empleamos para impulsar un proceso de adaptabilidad. Las mallas producidas proporcionan soluciones con errores significativamente más bajos que las obtenidos usando un refinamiento uniforme.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The current challenges in computational mechanics pertain to analyzing and handling large amounts of data, exploring high-dimensional parametric spaces and providing answers in short or real-time. This alludes to concepts like *digital twins* and technologies like *machine learning*, which are not only buzzwords, but methodologies to be considered in combination with the classical computational models [1, 2]. When modeling demanding systems like these, a high computational effort is needed, leading to a large response time between the identification of the details of the model to be considered and the availability of the response. Furthermore, the modification of details in the model leads to re-analyses of the process, further delaying the determination of the desired response. This is a major issue in our current quick paced reality, which relies in the premise that the faster you can come up with a reliable solution, the better.

One way to achieve accurate solutions while maintaining feasible computational times is through Model Order Reduction (MOR) techniques, which aim to reduce the degrees of freedom of a problem by considering a simplified representation of the physical system [3]. A huge advantage of some the MOR methods is the possibility of performing the heavier computation, such as extracting the important modes of a solution, in an ”offline” phase. This allows the choice of allocating computational power and time to compute and store the set of all parametric solutions of a problem and use less powerful equipment to output particular solutions in real-time.

Among these reduced order methods, we can refer to the Proper Orthogonal Decomposition (POD), a technique often used for complex non-liner problems, that relies on *a posteriori* data to compute reduced basis, or the Proper Generalized Decomposition (PGD), a powerful way to deal with the *curse of dimensionality* in parametric problems, that builds approximated solutions using modes that are computed *a priori* [4, 5]. This thesis will focus on the PGD, analyzing the solutions it provides.

The PGD is used in several types of problems, ranging from multidimensional and parametric models all the way through the creation of computational vademecum to be used for real time simulations [6, 7]. The possibility of trading a few hours of computational time for real-time responses, makes the PGD specially attractive for applications such as *digital twins*. One issue with PGD is its complexity when dealing with geometry as one of the parameters, although there are researches that try to tackle this problem with different approaches, such as integrating PGD solutions and NURBS [8–11].

Another challenge that the application of PGD faces is that the method lacks a posteriori estimations tools and adaptivity strategies[5, 12–14]. The solutions obtained through reduced order methods are approximations of the exact solution of the mathematical equations used to model the problem being considered, requiring verification and validation tools to attest their accuracy, and PGD is no exception. There are several ways to analysis the errors in the approximations obtained, but the only way of obtaining solutions with guaranteed error bounds is by applying dual error analysis [15].

The key idea of dual analysis is to combine the information from complementary solutions of a problem, one compatible and the other equilibrated, to compute a bound on the error of both solutions. The finite element method (FEM) is usually applied to solve the PGD equations and in a finite element approximation we can have either a displacement model, where compatibility is imposed exactly, or an equilibrium model, where we can impose equilibrium exactly. It is known that when taking these models separately, for force-driven problems, we have compatible models that are too stiff, and equilibrium models that are too flexible. An opposite effect is observed in a displacement driven problem. For the general case the energy of the difference of the two complementary solutions is a bound on their errors [16].

One of the earlier works using the combination of equilibrium and compatibility to obtain error bounds is the work of Prager and Synge [17], where the steps to obtain bounds of the error of dual solutions are presented, with calculations that rely on geometric concepts to show the relation between the dual solutions. Later works in the field are assigned to Freijs de Veubeke in the finite element method, with formulations of equilibrium elements [18]. More recent researches in the field were performed by Chamoin and Ladevèze, focusing in the constitute relation error and equilibrated fields obtained from recovered solutions[19–21], and by Moitinho, using the extended formulation from Prager to general problems proposed by Debongnie [16] and using a newly form of computing the actual equilibrated fields based on hybrid finite elements [22, 23].

When analyzing the errors of a solutions, it is often desirable to have information of specific quantities of interest (QoI). There are several works that takes the concept of goal-oriented formulations and apply to reduced methods, many specific for the PGD method, but few that explore the use of the dual analysis in this context [24–26]. The solutions obtained from dual analysis are ideal when working with QoI, as they can provide high quality bounds of error on the QoI of the worst solution [27]. Dual analysis is also effective to drive adaptivity mesh refinement processes, as they provide information of the errors in the elements either in a global or in a local framework [28].

The thesis main goal is assessing the error of PGD solutions and use theses errors to drive adaptivity process both the in the physical an parametric domains. This major goal can be divided into smaller specific goals and, throughout this text, we explain how we can achieve them.

The specific objectives can be summarized as:

- Define a linear elastic problem and obtain approximated solutions, one compatible and one equilibrated, writing them using the PGD methodology: We need to define a working problem in order to perform the error analysis in the PGD solutions. The linear elastic problem is selected because of the well established theory behind it and the simplicity in obtain solutions with a physical meaning that is easy to interpret.

- Use these dual solutions to compute upper bounds for their error, measured in the energy norm, as proposed in [29]: Naturally, there are several ways to assess the errors of a solution, but we choose the dual analysis as its the only one with guaranteed lower and upper bounds. This way we can certify the results obtained from the PGD model.

- Extend the application of dual analysis to obtain bounds for Quantities of Interest: It is important to obtain find a way to assess the solutions obtained from the PGD, but it is also necessary that the format of the outputs provided by this assessments have a physical meaning. Therefore, computing quantities of interest and their respective bounds using

PGD is a necessary step to extract a meaning from the outputs provided by the dual analysis.

- Use the complementary solutions to drive mesh adaptivity processes driven by global and local output errors: Once we are capable of accurately assessing the errors of the PGD solutions, we desire to use this information to define a mesh that can provide better solutions at a reduced cost.

1.2 Thesis Outline

The text presented here is a combination of the papers written during the course of the PhD program. The content of the papers was adapted to fit one single work, including, but not limited, to the addition of variables definitions and the presentation of the results.

Chapter 2 presents the linear elastic problem, defining the continuous equations that described the problem, along with techniques to find approximated solutions. This chapter introduces the problem equations, with a focus on the approximations obtained for the equilibrated finite element formulation.

The following chapter presents the main idea behind dual analysis, showing how to use the complementary solutions to compute bounds for global and local output errors. This chapter has a complete description of the process to obtain displacements as local outputs, along with their respective errors.

Chapter 4 introduces the parametric problem and the concepts of the PGD approximations, applying it to obtain PGD compatible and equilibrated solutions. The PGD approximations are used in the equations for the error measures, leading to the respective PGD approximated error measures. A few implementations details are described, as different approaches can be taken to reduce the computational cost of the solutions.

Chapter 5 summarizes the practical aspects for the adaptivity process, with a brief explanation of the methodology used to obtain the adapted solutions and meshes. The stopping criteria used for the adaptivity and PGD models are also explained here.

The chapter 6 gathers the results obtained throughout the thesis. This chapter is divided in three parts: the first one focuses on the results obtained for a one dimensional problem, that serve as validation for the results. This problem also serve to introduce the adaptive process using the global errors measures, for both the physical and the parametric spaces. The second part of this chapter described the work with a 2D problem, focusing on the use of the global error for mesh adaptivity. The last part deals with a quantity of interest problem, presenting results for meshes adapted based on the local output error measures.

The final chapter is a conclusion of the work, summarizing the concepts described and proposing new researching topics.

We appended the papers this thesis is based on in the end. I participate in three of them as the first author and in one as the co-author.

Chapter 2

Governing Equations

A large portion of engineering problems can be categorized, at least in an initial phase, as elasticity problems, from the analysis of the stresses on small mechanical systems to the design of buildings. This chapter describes all the necessary equations to define and solve a two dimensional linear elastic problem. It describes the continuous problem as well as the steps to obtain approximated solutions that are either compatible or equilibrated. We give emphasis on how to obtain the matrices for the equilibrium formulation. These approximations will be used afterwards to obtain PGD parametric approximations.

2.1 Continuous Problem

We consider a linear elastic body occupying a domain $\Omega \subset \mathbb{R}^2$ and subject to displacements $\mathbf{u}$. The loadings consist of body forces $\bar{\mathbf{b}}$, imposed tractions $\bar{\mathbf{t}}$, and prescribed boundary displacements $\bar{\mathbf{u}}$, where a bar is used to indicate that these are the input quantities. The boundary of Ω is decomposed as $\Gamma = \Gamma_N \cup \Gamma_D$, with Γ_N and Γ_D being respectively the Neumann and Dirichlet, so that $\Gamma_N \cap \Gamma_D = \emptyset$.

Considering that the problem is defined in a Cartesian reference system (x, y), we can write the compatibility equation for this problem, as:

$$\varepsilon = \mathbf{D}\mathbf{u}, \tag{2.1}$$

where ε is the strain vector represented in Voigt's notation and $\mathbf{D}$ the differential operator defined as:

$$\mathbf{D} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}. \tag{2.2}$$

The equilibrium equation can be written as:

$$\mathbf{D}^T \sigma + \bar{\mathbf{b}} = \mathbf{0}, \tag{2.3}$$

where σ is the Cauchy stress tensor represented in Voigt's notation and $\bar{\mathbf{b}}$ the body forces. Finally, we can write the constitutive relation that links the former two equations as:

$$\varepsilon = \boldsymbol{\mathcal{C}}\sigma, \tag{2.4}$$

where $\boldsymbol{\mathcal{C}}$ is the elasticity operator, defined for plane stresses as:

$$\boldsymbol{\mathcal{C}} = \frac{1}{E} \begin{bmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 2(1+\nu) \end{bmatrix}. \tag{2.5}$$

The material variables E and ν are the Young's modulus and the Poisson's ratio, respectively. The boundary conditions for this problem are:

$$\mathbf{u} - \bar{\mathbf{u}} = \mathbf{0} \qquad \text{on } \Gamma_D \tag{2.6}$$

for the Dirichhlet boundary and

$$\mathbf{N}\sigma - \bar{\mathbf{t}} = \mathbf{0} \qquad \text{on } \Gamma_N \tag{2.7}$$

for the Neumann boundary, with $\mathbf{N}$ being the boundary operator defined as:

$$\mathbf{N} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix}. \tag{2.8}$$

2.2 Compatible Finite Element Formulations and Approximations

When considering a compatible displacement field, we can define a strain energy $\mathcal{U}$:

$$\mathcal{U}(\varepsilon(\mathbf{u})) = \frac{1}{2}\int_{\Omega} \left(\varepsilon(\mathbf{u})^T\, \boldsymbol{\mathcal{C}}^{-1}\, \varepsilon(\mathbf{u})\right) \mathrm{d}\Omega \tag{2.9}$$

and the work of the external forces $\mathcal{V}$

$$\mathcal{V}(\mathbf{u}) = \int_{\Omega} \mathbf{u}^T\bar{\mathbf{b}}\ \mathrm{d}\Omega\ + \int_{\Gamma_N} \mathbf{u}^T\bar{\mathbf{t}}\ \mathrm{d}\Gamma. \tag{2.10}$$

The solution of the continuous problem described in Section 2.1 can be obtained from the set of all compatible displacement fields by

$$\min\left(\mathcal{U}(\varepsilon(\mathbf{u})) - \mathcal{V}(\mathbf{u})\right). \tag{2.11}$$

Based on the approximation of the displacement field $\mathbf{u}$, we can write an approximate displacement field $\mathbf{u}_h$ using the standard finite element approach, such that:

$$\mathbf{u}_h = \mathbf{U}\,\hat{\mathbf{u}}, \tag{2.12}$$

where $\mathbf{U}$ is the matrix with the displacement approximation functions and $\hat{\mathbf{u}}$ is the vector of nodal (or modal) unknowns. These displacements are continuous and the associated strains are piecewise continuous, given by

$$\varepsilon_h = (\mathbf{D}\mathbf{U})\,\hat{\mathbf{u}} = \mathbf{B}\,\hat{\mathbf{u}}. \tag{2.13}$$

The solution of (2.11) can be obtained from

$$\boldsymbol{\mathcal{K}}\,\hat{\mathbf{u}} = \hat{\mathbf{f}}, \tag{2.14}$$

which depends on the stiffness matrix, so that $\boldsymbol{\mathcal{K}}\hat{\mathbf{u}} = \partial\mathcal{U}(\mathbf{B}\hat{\mathbf{u}})/\partial\hat{\mathbf{u}}$, and on the vector of equivalent generalised nodal forces, $\hat{\mathbf{f}} = \partial\mathcal{V}(\mathbf{U}\hat{\mathbf{u}})/\partial\hat{\mathbf{u}}$.

The corresponding approximated solution, defined by the generalised displacement parameters, $\hat{\mathbf{u}}$, strongly satisfies the compatibility conditions and weakly satisfies equilibrium. Displacement based Finite Element formulations are widely known and the details of this approach will be skipped.

2.3 Equilibrium Finite Element Formulations and Approximations

For an equilibrated Cauchy stress field σ, the complementary strain energy $\mathcal{U}_c$ is defined as

$$\mathcal{U}_c(\sigma) = \frac{1}{2}\int_{\Omega} \left(\sigma^T \boldsymbol{\mathcal{C}}\, \sigma\right) \mathrm{d}\Omega, \tag{2.15}$$

together with the work of the imposed displacements $\mathcal{V}_c$

$$\mathcal{V}_c(\sigma) = \int_{\Gamma_d} (\mathbf{N}\sigma)^T \bar{\mathbf{u}}\ \mathrm{D}\Gamma. \tag{2.16}$$

The solution of the problem described in Section 2.1 can be obtained from the set of all equilibrated stress fields by

$$\min\left(\mathcal{U}_c(\sigma) - \mathcal{V}_c(\sigma)\right). \tag{2.17}$$

Due to the tensorial nature of the stress field, the problem of obtaining an equilibrated approximation of the stresses is not as straightforward as it is for the displacements, which are vectors. When such an approximation is determined, the corresponding approximated Cauchy stresses σ_h can be written, assuming for simplicity that the body forces are equal to zero, as

$$\sigma_h = \mathbf{S}_g\, \hat{\mathbf{s}}_g. \tag{2.18}$$

where each column of $\mathbf{S}_g$ is a globally equilibrated stress distribution, and $\hat{\mathbf{s}}_g$ collects the corresponding parameters. In general each of these distributions involves more than one element because they must satisfy not only the equilibrium equation (2.3) and the boundary condition (2.7), which can be imposed for individual elements, but also the generalisation (2.7) for the inter-element boundaries, which has a global effect.

The solution of (2.17) can be obtained from

$$-\boldsymbol{\mathcal{F}}_g\, \hat{\mathbf{s}}_g = \hat{\mathbf{e}}_g, \tag{2.19}$$

which depends on the flexibility matrix, so that $\boldsymbol{\mathcal{F}}_g \hat{\mathbf{s}}_g = \partial \mathcal{U}_c(\mathbf{S}_g \hat{\mathbf{s}}_g)/\partial \hat{\mathbf{s}}_g$, and on the vector of equivalent strains, $\hat{\mathbf{e}}_g = \partial \mathcal{V}_c(\mathbf{S}_g \hat{\mathbf{s}}_g)/\partial \hat{\mathbf{s}}_g$.

The tensorial nature of the stress field implies that inter-element equilibrium is expressed in terms of the projection of the stress components on the interface, instead of imposing continuity of the components of the stress tensor. We do not use system (2.19) as such because the aforementioned conditions imply that it is not possible to obtain, by direct assembly, a globally equilibrated stress field. Instead, a hybrid approach is used [22], wherein the equilibrium (2.3) is imposed *a priori* inside each element, while the boundary condition (2.7) and its generalisation are imposed *a posteriori* in a weak form. The weighting functions are selected so that, for polynomial approximations, the weak form implies strong equilibrium.

2.3.1 Hybrid Equilibrium Finite Elements

The Lagrange multipliers corresponding to these constraints, $\mathbf{v}$, are identified as the displacements at the edges of the elements, which are approximated as

$$\mathbf{v}_h = \mathbf{V}\,\hat{\mathbf{v}}, \tag{2.20}$$

where the columns of $\mathbf{V}$ assemble the bases for the displacements on all the edges of the mesh and $\hat{\mathbf{v}}$ collects the weights defining their combination.

In order to have an energetically consistent approach the functions used to approximate the displacements correspond to the weighting functions used to weakly impose equilibrium.

The starting point of the formulation is an element-wise approximation of self-equilibrated stresses, without consideration of what happens on their boundaries. We expresses these approximated Cauchy stresses as

$$\sigma_h = \mathbf{S}\,\hat{\mathbf{s}}, \tag{2.21}$$

where the columns of $\mathbf{S}$ are such that:

$$\mathbf{D}^T \mathbf{S} = \mathbf{0}, \tag{2.22}$$

thus enforcing equilibrium inside each element.

On a given side i of an element we define a hybrid energy $\mathcal{V}_h^i$, which projects the element tractions onto the displacements of that side, meaning that:

$$\mathcal{V}_h^i(\mathbf{v}, \sigma) = \int_{\Gamma_i} \mathbf{v}^T (\mathbf{N}\sigma) \, \mathrm{d}\Gamma, \tag{2.23}$$

so that weak equilibrium for a side Γ_i of an element belonging to Γ_N is imposed by

$$\mathcal{V}_h^i(\mathbf{v}, \sigma) = \mathcal{V}_t^i(\mathbf{v}) \quad \forall \mathbf{v}, \qquad \text{with } \mathcal{V}_t^i(\mathbf{v}) = \int_{\Gamma_i} \mathbf{v}^T \bar{\mathbf{t}} \, \mathrm{d}\Gamma. \tag{2.24}$$

Similarly, when side i is adjacent to elements A and B, weak equilibrium is imposed by

$$\mathcal{V}_h^i(\mathbf{v}, \sigma_A) + \mathcal{V}_h^i(\mathbf{v}, \sigma_B) = \mathbf{0} \quad \forall \mathbf{v}. \tag{2.25}$$

Then equilibrium between elements and on Γ_N is imposed by elemental equilibrium matrix given by

$$\boldsymbol{\mathcal{D}} \, \hat{\mathbf{s}} = \partial \mathcal{V}_h^i(\mathbf{V}\hat{\mathbf{v}}, \mathbf{S}\hat{\mathbf{s}}) / \partial \hat{\mathbf{v}}. \tag{2.26}$$

The transpose of this matrix expresses the effect of the boundary displacements, which induce equivalent strains that depend on the parameters of the boundary displacements.

$$\boldsymbol{\mathcal{D}}^T \hat{\mathbf{v}} = \partial \mathcal{V}_h^i(\mathbf{V}\hat{\mathbf{v}}, \mathbf{S}\hat{\mathbf{s}}) / \partial \hat{\mathbf{s}}. \tag{2.27}$$

Combining (2.19) with the constraints resulting from the application of the Lagrange, (2.26) and (2.27), the resulting system has the following form:

$$\begin{bmatrix} -\boldsymbol{\mathcal{F}} & \boldsymbol{\mathcal{D}}^T \\ \boldsymbol{\mathcal{D}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{s}} \\ \hat{\mathbf{v}} \end{Bmatrix} = \begin{Bmatrix} \hat{\mathbf{e}} \\ \hat{\mathbf{t}} \end{Bmatrix}, \tag{2.28}$$

where $\boldsymbol{\mathcal{F}}\hat{\mathbf{s}} = \partial \mathcal{U}_c(\mathbf{S}\hat{\mathbf{s}}) / \partial \hat{\mathbf{s}}$, $\hat{\mathbf{e}} = \partial \mathcal{V}_c(\mathbf{S}\hat{\mathbf{s}}) / \partial \hat{\mathbf{s}}$, and $\hat{\mathbf{t}} = \partial \mathcal{V}_t^i(\mathbf{V}\hat{\mathbf{v}}) / \partial \hat{\mathbf{v}}$.

The corresponding approximated solution, defined by the generalised stress parameters, $\hat{\mathbf{s}}$, and by the generalised boundary displacements, $\hat{\mathbf{v}}$, strongly satisfies the equilibrium conditions and weakly satisfies compatibility.

In the following Section we briefly described the main aspects of the stress based approach that is used.

2.3.2 Implementation Details for Hybrid Equilibrium Elements

The approach used for the equilibrium formulation differs from that used in [22] because it works with the stress approximation defined in the frame of the master element, allowing for analytical expressions of the integrals involved, which can potentially account for variations in the geometry of the mesh. The details of this implementation are presented in [23] and briefly described next.

Traditionally, we have $\mathbf{S}(x, y)$, which is equivalent to say that $\mathbf{S}$ is defined in the global reference frame. Here, in order to work in the frame of the master element, we derive $\mathbf{S}(\xi, \eta)$ from an approximation of the first Piola-Kirchhoff stress tensor $\mathbf{P}$, associated with the mapping from the physical undeformed configuration to the master element.

Triangular elements are used, with a mapping between master and physical domain, where $\mathbf{F}$ is the Jacobian matrix of the transformation and J its determinant. The Cauchy stresses are obtained from the first Piola-Kirchhoff stresses as

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix} = \frac{1}{J} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} F_{11} & F_{21} \\ F_{12} & F_{22} \end{bmatrix} = \frac{1}{J} \mathbf{P}\, \mathbf{F}^T. \tag{2.29}$$

Using Voigt's notation, a transformation matrix $\mathbf{T}_P$ is introduced to write (2.29) as:

$$\sigma = \begin{bmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xy} \end{bmatrix} = \frac{1}{J}\mathbf{T}_P\mathbf{P} = \frac{1}{J} \begin{bmatrix} F_{11} & 0 & F_{12} & 0 \\ 0 & F_{22} & 0 & F_{21} \\ F_{21} & 0 & F_{22} & 0 \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{22} \\ P_{12} \\ P_{21} \end{bmatrix}, \tag{2.30}$$

which is not always valid, because, for a generic $\mathbf{P}$, the corresponding σ_{xy} is not necessarily equal to σ_{yx}.

In general, these stresses do not verify equilibrium in the absence of body forces, neither in terms of forces (because they may have non-zero divergence) nor in terms of moments (because the corresponding Cauchy stresses may be non-symmetric).

The approximation of the unconstrained (without equilibrium conditions) first Piola-Kirchhoff stress tensor is, in Voigt's notation:

$$\mathbf{P} = \mathbf{\Phi}_d^P \hat{\mathbf{P}}, \tag{2.31}$$

where $\mathbf{\Phi}_d^P(\xi, \eta)$ is the matrix with the unconstrained approximation of the Piola stress components of degree d.

The stress approximation that we are seeking is obtained by imposing conditions on the resulting stresses, $\mathbf{P}$, so that they:

- correspond to a symmetric Cauchy stress tensor;
- have zero divergence for either the Piola or for the Cauchy stress tensor.

These are linear constraints, which impose relations between the components of $\hat{\mathbf{P}}$, enforced by considering a smaller set of parameters, $\hat{\mathbf{P}}_E$, sequentially left multiplied by two transformation matrices, $\mathbf{T}_S$ (imposing symmetry) and $\mathbf{T}_D$ (imposing zero divergence).

The resulting equation is [23]:

$$\sigma = \underbrace{\frac{1}{J}\mathbf{T}_P \mathbf{\Phi}_d^P \mathbf{T}_S \mathbf{T}_D}_{\mathbf{S}} \, \underbrace{\hat{\mathbf{P}}_E}_{\hat{\mathbf{s}}} \tag{2.32}$$

The matrices in (2.28) are not computed globally. Instead they are independently determined for each element and assembled. Note that $\boldsymbol{\mathcal{F}}$ is block diagonal, while matrix $\boldsymbol{\mathcal{D}}$ is connecting adjacent elements, imposing inter-element equilibrium and sharing the boundary displacements of the common side.

The flexibility matrix $\boldsymbol{\mathcal{F}}_e$ of element e is:

$$\boldsymbol{\mathcal{F}}_e = \int_{\Omega_e} \mathbf{S}_e^T \boldsymbol{\mathcal{C}} \, \mathbf{S}_e \, \mathrm{d}\Omega \, . \tag{2.33}$$

Substituting (2.32) in (2.33), and dropping the reference to the element, we get:

$$\boldsymbol{\mathcal{F}} = \frac{1}{J} \mathbf{T}_D^T \mathbf{T}_S^T \int_{\Omega_0} \mathbf{\Phi}_d^{P^T} (\mathbf{T}_P^T \boldsymbol{\mathcal{C}} \mathbf{T}_P) \, \mathbf{\Phi}_d^P \, \mathrm{d}\Omega_0 \, \mathbf{T}_S \mathbf{T}_D, \tag{2.34}$$

where Ω_0 is the domain of the master element.

For boundary i of that element, the equilibrium matrix, when defined as a function of the Cauchy stresses, is:

$$\boldsymbol{\mathcal{D}}_i = \int_{\Gamma_i} \mathbf{V}_i^T \mathbf{N}_i^T \mathbf{S} \; \mathrm{d}\Gamma. \tag{2.35}$$

Using the first Piola-Kirchhoff stress tensor we can write:

$$\mathbf{N}^T \sigma = \mathbf{N}_0^T \mathbf{P}, \qquad \text{with} \qquad \mathbf{N}_0^T = \begin{bmatrix} n_\xi & 0 & n_\eta & 0 \\ 0 & n_\eta & 0 & n_\xi \end{bmatrix}. \tag{2.36}$$

Substituting Equations (2.32) and (2.36) into Equation (2.35) we have:

$$\boldsymbol{\mathcal{D}}_i = \int_{\Gamma_i} \mathbf{V}_i^T \mathbf{N}_0^T \boldsymbol{\Phi}_d^P \; \mathrm{d}\Gamma_i \, \mathbf{T}_S \mathbf{T}_D \; . \tag{2.37}$$

One major issue with equilibrium formulations is that the governing system (2.28) may have dependent equations [22], associated with the row rank deficiency of the assembled $\boldsymbol{\mathcal{D}}$ matrix. The consequences of this situation are twofold: It is possible to have boundary displacements, known as spurious kinematic modes (SKMs), that do not introduce generalized strains in elements, although they do not correspond to rigid body modes; There are inadmissible loads, *i.e.* loads that cannot be equilibrated by the stresses in the elements.

In this application, we avoid these problems by either selecting meshes known *a priori* to be free from SKMs, or by having loads that are known to be admissible and blocking the SKMs using the technique described in Section 6.2 of [22]. Because this approach requires knowing the SKMs, we use the sparse SVD routine in matlab, applied to compute the lowest singular values of the assembled $\boldsymbol{\mathcal{D}}$. This procedure has to determine some non null singular values in order to guarantee that all the spurious modes, which are associated with (numerically) null singular values, are determined.

Chapter 3

Error Measures

Solution obtained through approximations need some sort of verification to assess its accuracy. Knowing the errors of a solution and how they range can lead to cheaper and more efficient designs. In this chapter we use complementary solutions to compute bounds for the global errors, local outputs and bounds of their errors. We present a detailed derivation of the equations used to obtain selected displacement averages, which is analogous to the equations presented in [27] to obtain stresses averages. We briefly explain how to obtain bounds both in a local and global framework, but a more complete explanation can be found in [22] and references therein.

3.1 Bound of the Global Errors

Given a pair of dual solutions, the bound, ϵ^2, of the square energy of their error is [17]:

$$\epsilon^2 = \int_\Omega (\sigma_k - \sigma_s)(\varepsilon_k - \varepsilon_s)\ \mathrm{d}\Omega \ \geq \ \begin{cases} \displaystyle\int_\Omega (\sigma - \sigma_s)(\varepsilon - \varepsilon_s)\ \mathrm{d}\Omega \\ \displaystyle\int_\Omega (\sigma_k - \sigma)(\varepsilon_k - \varepsilon)\ \mathrm{d}\Omega \end{cases} \tag{3.1}$$

where no subscript indicates the exact solution, the subscript k indicates solutions that are $\underline{\text{k}}$inematically admissible and s solutions that are $\underline{\text{s}}$tatically admissible. We call the integrand in (3.1) an *error density*, defined as:

$$\rho = (\sigma_k - \sigma_s)(\varepsilon_k - \varepsilon_s) \tag{3.2}$$

Substituting the elasticity equations and assuming no initial strains, Equation (3.1) becomes:

$$\epsilon^2 = \int_\Omega \left(\varepsilon_k^T \boldsymbol{\mathcal{C}}^{-1} \varepsilon_k + \sigma_s^T \boldsymbol{\mathcal{C}} \sigma_s - 2\sigma_s^T \varepsilon_k \right) \, \mathrm{d}\Omega \, . \tag{3.3}$$

We will often refer to the quantities obtained from this equation as the global error bounds, as they account for the behaviour of the whole problem being studied.

3.2 Local Outputs and their Bounds

When solving an elasticity problem, it is common to have values that are more relevant for the design, such as stresses in zones with known concentrations or displacements that should not exceed a certain threshold. In order to look at these outputs we define local quantities, which represent weighted integrals of stresses or displacements in the zones of particular interest.

For instance, when these weighting functions are constant and equal to unity, the quantities will be average values of the fields multiplied by its dimension measure (volume, area or length). For example, when seeking the output along a line, the local quantity is the average value of the field at that line multiplied by its length. Different forms can be used for the weighting functions, with each form having a different physical meaning for the outputs obtained. For the examples described in this paper, we will always compute the local quantities as the average along lines.

We can compute the local quantities by applying the principle of virtual work, where the weighting functions act as virtual loads. In this text we will refer to these loads as *virtual*, in the sense that they do not need to exist. These loads are also usually referred to in the literature as dual, adjoint or extractors. Likewise, the load that is physically applied to the structure is denoted as *real*.

From here on, a bar on top of a variable indicates that it is associated with the virtual problem, except for the prescribed actions.

Using stresses from the virtual solution $\bar{\sigma}_s$ that are equilibrated, we can obtain the exact local displacement output $\mathcal{L}_k(\mathbf{u})$ of the exact displacement field $\mathbf{u}$ as:

$$\mathcal{L}_k(\mathbf{u}) = \int_\Omega \bar{\sigma}_s^T \varepsilon \, \mathrm{d}\Omega - \int_{\Gamma_u} (\mathbf{n} \; \bar{\sigma}_s) \, \bar{\mathbf{u}} \, \mathrm{d}\Gamma \tag{3.4}$$

and a similar expression is used to obtain an exact local stress output $\mathcal{L}_s(\sigma)$ [22].

Usually, it is not possible to obtain the exact solutions, hence we apply (3.4) to approximated solutions, in order to obtain estimates of the local output and bounds of their error.

To achieve this, we force the output derived from a compatible solution $\mathcal{L}_k(\mathbf{u}_k)$ to appear by adding and subtracting $\int_\Omega \bar{\sigma}_s^T \varepsilon_k \, \mathrm{d}\Omega$ in Equation (3.4), so that:

$$\mathcal{L}_k(\mathbf{u}) = \int_\Omega \bar{\sigma}_s^T (\varepsilon - \varepsilon_k) \, \mathrm{d}\Omega + \underbrace{\int_\Omega \bar{\sigma}_s^T \varepsilon_k \, \mathrm{d}\Omega - \int_{\Gamma_D} (\mathbf{n} \, \bar{\sigma}_s) \, \bar{\mathbf{u}} \, \mathrm{d}\Gamma}_{\mathcal{L}_k(\mathbf{u}_k)} \tag{3.5}$$

where we introduce the local displacement output obtained from the compatible solution $\mathcal{L}_k(\mathbf{u}_k)$, noting that a similar expression can be obtained from the equilibrated solution which is denoted as $\mathcal{L}_k(\sigma_s)$.

Now we manipulate Equation (3.5) to force a term that can be bounded to appear. We do so by adding and subtracting $\int_\Omega \bar{\sigma}_k^T (\varepsilon - \varepsilon_k) \, \mathrm{d}\Omega$:

$$\mathcal{L}_k(\mathbf{u}) = \int_\Omega (\bar{\sigma}_s^T - \bar{\sigma}_k^T)(\varepsilon - \varepsilon_k) \, \mathrm{d}\Omega + \int_\Omega \bar{\sigma}_k^T (\varepsilon - \varepsilon_k) \, \mathrm{d}\Omega + \mathcal{L}_k(\mathbf{u}_k) \tag{3.6}$$

The second integral in this expression is not computable, since it requires the exact value of ε. By adding and subtracting $\int_\Omega \bar{\sigma}_k^T (\varepsilon - \varepsilon_s) \, \mathrm{d}\Omega$ in Equation (3.6) we define a corrected output $\tilde{\mathcal{L}}_k(\mathbf{u}_k)$, which has an error that is bounded, resulting in:

$$\mathcal{L}_k(\mathbf{u}) = \int_\Omega (\bar{\sigma}_s^T - \bar{\sigma}_k^T)(\varepsilon - \varepsilon_k) \, \mathrm{d}\Omega + \underbrace{\mathcal{L}_k(\mathbf{u}_k) + \int_\Omega \bar{\sigma}_k^T (\varepsilon_s - \varepsilon_k) \, \mathrm{d}\Omega}_{\tilde{\mathcal{L}}_k(\mathbf{u}_k)} + \int_\Omega \bar{\sigma}_k^T (\varepsilon - \varepsilon_s) \, \mathrm{d}\Omega \tag{3.7}$$

Notice that the term $\int_\Omega \bar{\sigma}_k^T (\varepsilon - \varepsilon_s) \, \mathrm{d}\Omega$ is equivalent to $\int_\Omega \bar{\varepsilon}_k^T (\sigma - \sigma_s) \, \mathrm{d}\Omega$, which is zero because $(\sigma - \sigma_s)$ is a self balanced stress field and $\bar{\varepsilon}_k$ corresponds to a compatible displacement field with homogeneous boundary conditions. Rewriting Equation (3.7) to isolate the bounding term:

$$\mathcal{L}_k(\mathbf{u}) - \tilde{\mathcal{L}}_k(\mathbf{u}_k) = \int_\Omega (\bar{\sigma}_s^T - \bar{\sigma}_k^T)(\varepsilon - \varepsilon_k) \, \mathrm{d}\Omega \tag{3.8}$$

In an analogous way, by using σ_s instead of $\mathbf{u}_k$ in Equation (3.6) we obtain an error associated with an equilibrated solution as:

$$\mathcal{L}_k(\mathbf{u}) - \mathcal{L}_k(\sigma_s) = \int_\Omega (\bar{\sigma}_s^T - \bar{\sigma}_k^T)(\varepsilon - \varepsilon_s) \, \mathrm{d}\Omega \tag{3.9}$$

We can then average Equations (3.8) and (3.9) to write the error of the output associated with an averaged solution:

$$\mathcal{L}_k(\mathbf{u}) - \tilde{\mathcal{L}}_k(\sigma_a) = \int_\Omega (\bar{\sigma}_s^T - \bar{\sigma}_k^T)(\varepsilon - \varepsilon_a)\,\mathrm{d}\Omega \tag{3.10}$$

Equation (3.10) is not computable because of its dependency on the exact value of ε, however we apply the Cauchy–Schwarz inequality to it, which will lead to

$$|\mathcal{L}_k(\mathbf{u}) - \tilde{\mathcal{L}}_k(\sigma_a)| \leq \underbrace{\sqrt{\int_\Omega (\bar{\sigma}_s^T - \bar{\sigma}_k^T)(\bar{\varepsilon}_s - \bar{\varepsilon}_k)\,\mathrm{d}\Omega}}_{\leq\ \bar{\epsilon}} \underbrace{\sqrt{\int_\Omega (\sigma^T - \sigma_a^T)(\varepsilon - \varepsilon_a)\,\mathrm{d}\Omega}}_{=\frac{1}{2}\epsilon} \ \leq\ \frac{1}{2}\bar{\epsilon}\epsilon = \epsilon_{a_{local}} \tag{3.11}$$

This will result in an explicit definition of upper and lower bounds of the output:

$$\tilde{\mathcal{L}}_k(\sigma_a) -\ \epsilon_{a_{local}} \leq \mathcal{L}_k(\mathbf{u}) \leq \tilde{\mathcal{L}}_k(\sigma_a) +\ \epsilon_{a_{local}}. \tag{3.12}$$

Assuming there are no imposed displacements, we can substitute the elasticity equations to rewrite the corrected average local output $\tilde{\mathcal{L}}_k(\sigma_a)$ as:

$$\tilde{\mathcal{L}}_k(\sigma_a) = \frac{1}{2}\int_\Omega \left(\bar{\sigma}_s^T \boldsymbol{\mathcal{C}}\,\sigma_s + \bar{\sigma}_s^T \varepsilon_k - \bar{\varepsilon}_k^T \boldsymbol{\mathcal{C}}^{-1}\varepsilon_k + \bar{\varepsilon}_k^T \sigma_s\right)\,\mathrm{d}\Omega. \tag{3.13}$$

We aim to compute the bounds as sums of elemental contributions, which may be used as mesh adaptivity indicators. We compute these contributions with a modified version of the equations provided in [27]. We assume the more general case that both compatible and equilibrated solutions have the same importance. Hence, the square bound of average local output error is:

$$\bar{\epsilon}\epsilon = \epsilon_{a_{local}}^2 = \frac{1}{8}\sum_e^{n_e}\left(\epsilon^2\,\bar{\epsilon}_{[e]}^2 + \epsilon_{[e]}^2\,\bar{\epsilon}^2\right) \tag{3.14}$$

We can also write a non squared version of the bounds, that comes from taking the bounds from a different set of equations [28]. The bound of the average local output error is then:

$$\epsilon_{a_{local}} = \frac{1}{4}\sum_e^{n_e}\left(\bar{\epsilon}_{[e]}^2\frac{\epsilon}{\bar{\epsilon}} + \epsilon_{[e]}^2\frac{\bar{\epsilon}}{\epsilon}\right) \tag{3.15}$$

Equation (3.15) is more suitable when paired with the local outputs, because of the consistency of the dimension of the bounds and the outputs. For mesh adaptivity purposes, Equation (3.14) can be better, as it gives more importance to the regions with higher values of the error indicator.

Chapter 4

Parametric Problem and Matrix Separation

When working on an engineering project, it is common to have many iterations of the design that have only small differences in the parameters definition, such as different dimensions for a section of the part or a change in the material being used. Every new iteration with new parameters requires the verification of the solutions obtained, which can be costly if the problem is large or if many iterations are needed. The PGD method can be used to circumvent this problem. This chapter focus on the definition of parametric PGD approximations, for the solutions obtained as well as for the their errors and respective bounds.

4.1 Parametric PGD Approximations

We are interested in defining solutions as a function of material parameters for force driven problems without considering body loads. We define $\boldsymbol{\mu}$, a vector of n_p parameters $\mu_1, \mu_2, \ldots, \mu_{n_p}$, related to the material properties of the problem, each defined in its own domain $\Omega_i \subset \mathbb{R}$, with $i = 1, 2, \ldots, n_p$. Then, the vector of parameters $\boldsymbol{\mu}$ is defined so that $\boldsymbol{\mu} \subset \Omega_{\boldsymbol{\mu}} = \Omega_1 \otimes \Omega_2 \otimes \cdots \otimes \Omega_{n_p} \subset \mathbb{R}^{n_p}$.

Ideally, we would want to obtain the solution of the continuous problem defined in Section 2 as a function of these parameters μ_i. However, even for fixed parameters, solving this problem requires techniques that involve some form of approximation. We use the finite element method to transform the continuous problem into a discrete one, wherein the solution depends on a finite set of unknowns.

For the compatible approach, the unknowns of our discrete problem are the displacements $\hat{\mathbf{u}}(\boldsymbol{\mu})$, whereas for the equilibrated approach, the stresses $\hat{\mathbf{s}}(\boldsymbol{\mu})$ and the boundary displacements $\hat{\mathbf{v}}(\boldsymbol{\mu})$ are to be sought.

In the case that we are considering, a force driven problem with boundary tractions and with the parameters related to the material properties, the governing systems of these finite element formulations take the following forms:

$$\boldsymbol{\mathcal{K}}(\boldsymbol{\mu})\,\hat{\mathbf{u}}(\boldsymbol{\mu}) = \hat{\mathbf{f}}; \tag{4.1}$$

$$\begin{bmatrix} -\boldsymbol{\mathcal{F}}(\boldsymbol{\mu}) & \boldsymbol{\mathcal{D}}^T \\ \boldsymbol{\mathcal{D}} & \mathbf{0} \end{bmatrix} \left\{ \begin{array}{c} \hat{\mathbf{s}}(\boldsymbol{\mu}) \\ \hat{\mathbf{v}}(\boldsymbol{\mu}) \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{0} \\ \hat{\mathbf{t}} \end{array} \right\}. \tag{4.2}$$

When the effect of body forces is also considered, the zero on the second term of (4.2) becomes $\mathbf{e}(\boldsymbol{\mu})$. Similarly, for the case of imposed displacements, that term becomes non-zero, but does not depend on the parameters, whereas $\hat{\mathbf{f}}$ becomes $\hat{\mathbf{f}}(\boldsymbol{\mu})$. Finally, geometry related parameters will affect all terms of both systems.

It is possible to solve this system for a set of discrete values of the parameters μ_i, but as the number of parameters increases, the curse of dimensionality takes place and the cost of finding the parametric solution becomes prohibitive.

To avoid this problem, we are seeking a PGD solution [3], using the `algebraicPGDtools`[30]. This approach is ideal for solutions that can be represented in a separated manner, producing results in a separated tensor format, although non-separable inputs can also be used [14].

The PGD approximated solution for our unknowns are:

$$\hat{\mathbf{u}}(\boldsymbol{\mu}) \approx \hat{\mathbf{u}}^{n_k}_{PGD}(\boldsymbol{\mu}) = \sum_{m=1}^{n_k} \bar{\mathbf{u}}^m \prod_{i=1}^{n_p} U_i^m(\mu_i); \tag{4.3}$$

$$\hat{\mathbf{s}}(\boldsymbol{\mu}) \approx \hat{\mathbf{s}}^{n_s}_{PGD}(\boldsymbol{\mu}) = \sum_{m=1}^{n_s} \bar{\mathbf{s}}^m \prod_{i=1}^{n_p} S_i^m(\mu_i), \tag{4.4}$$

where n_k is the total number of modes for the compatible solution and n_s for the equilibrated solution. The variables $\bar{\mathbf{u}}$ and $\bar{\mathbf{s}}$ are the displacement and stress vectors describing the spatial modes. Matrices $U(\boldsymbol{\mu})$ and $S(\boldsymbol{\mu})$ correspond to sets of nodal functions of each parameter, which are obtained using the algebraic PGD approach described in [31]. Notice that $\hat{\mathbf{v}}$ has a similar approximation, which will not be used in this thesis and, therefore, is not presented here.

The `algebraicPGDtools` requires the definition of the input values of the problem in a separated manner. The separation of the contribution of each element to $\boldsymbol{\mathcal{F}}(\boldsymbol{\mu})$ and $\boldsymbol{\mathcal{K}}(\boldsymbol{\mu})$ is based on the separation of either (2.4) or of its inverse.

We write, for a given element, the separated form of the elasticity operator $\boldsymbol{\mathcal{C}}$ as

$$\boldsymbol{\mathcal{C}}(\boldsymbol{\mu}) = \sum_{m=1}^{n_C} \bar{\boldsymbol{\mathcal{C}}}^m \prod_{i=1}^{n_p} C_i^m(\mu_i). \tag{4.5}$$

In these expressions, the value of $C_i^m(\mu_i)$ is one, except for the n_C terms associated with the parameters affecting that element. The determination of the corresponding coefficients and of the modes $\bar{\boldsymbol{\mathcal{C}}}^m$ is exemplified later for some particular cases.

Then:

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\mu}) = \sum_{m=1}^{n_C} \bar{\boldsymbol{\mathcal{F}}}^m \prod_{i=1}^{n_p} C_i^m(\mu_i). \tag{4.6}$$

For each element, $\bar{\boldsymbol{\mathcal{F}}}^m$ corresponds to matrix $\boldsymbol{\mathcal{F}}$ computed using $\bar{\boldsymbol{\mathcal{C}}}^m$ instead of $\boldsymbol{\mathcal{C}}$.

Similarly, to obtain the stiffness matrix, we use

$$\boldsymbol{\mathcal{C}}^{-1}(\boldsymbol{\mu}) = \sum_{m=1}^{n_{\underset{\sim}{C}}} \bar{\underset{\sim}{C}}^m \prod_{i=1}^{n_p} \underset{\sim}{C}_i^m(\mu_i), \tag{4.7}$$

so that

$$\boldsymbol{\mathcal{K}}(\boldsymbol{\mu}) = \sum_{m=1}^{n_{\underset{\sim}{C}}} \bar{\boldsymbol{\mathcal{K}}}^m \prod_{i=1}^{n_p} \underset{\sim}{C}_i^m(\mu_i). \tag{4.8}$$

These decompositions, applied to (4.1) and (4.2), are used to obtain the PGD solutions (4.3) and (4.4).

4.2 Examples of Separation of the Constitutive Relations Matrices

To exemplify how the elasticity operators $\boldsymbol{\mathcal{C}}$ or $\boldsymbol{\mathcal{C}}^{-1}$ for plane stress can be expressed in a separated form, we recall that

$$\boldsymbol{\mathcal{C}} = \frac{1}{E}\begin{bmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 2(1+\nu) \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\mathcal{C}}^{-1} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}. \tag{4.9}$$

4.2.1 One material with two parameters

When E and ν are the parameters of the problem, $\boldsymbol{\mu} = \{E,\, \nu\}$, we can express the dependency on the parameters as:

$$\boldsymbol{\mathcal{C}}(E,\nu) = C_1^1(E)\, C_2^1(\nu)\, \bar{\boldsymbol{\mathcal{C}}}^1 + C_1^2(E)\, C_2^2(\nu)\, \bar{\boldsymbol{\mathcal{C}}}^2 = \frac{1}{E}\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}}_{\bar{\boldsymbol{\mathcal{C}}}^1} + \frac{1}{E}\,\nu\,\underbrace{\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix}}_{\bar{\boldsymbol{\mathcal{C}}}^2}, \tag{4.10}$$

where $C_2^1(\nu) = 1$ is implicit. Then, for a given element

$$\boldsymbol{\mathcal{F}}(E,\nu) = \int \mathbf{S}^T \boldsymbol{\mathcal{C}}(E,\nu)\, \mathbf{S}\, d\Omega = \frac{1}{E}\int \mathbf{S}^T \bar{\boldsymbol{\mathcal{C}}}^1 \mathbf{S}\, d\Omega + \frac{1}{E}\,\nu \int \mathbf{S}^T \bar{\boldsymbol{\mathcal{C}}}^2 \mathbf{S}\, d\Omega. \tag{4.11}$$

Similarly, for $\boldsymbol{\mathcal{C}}^{-1}$,

$$\boldsymbol{\mathcal{C}}^{-1}(E,\nu) = \underset{\sim}{C}{}_1^1(E)\, \underset{\sim}{C}{}_2^1(\nu)\, \bar{\underset{\sim}{\boldsymbol{\mathcal{C}}}}{}^1 + \underset{\sim}{C}{}_1^2(E)\, \underset{\sim}{C}{}_2^2(\nu)\, \bar{\underset{\sim}{\boldsymbol{\mathcal{C}}}}{}^2 = E\,\frac{1}{1-\nu^2}\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}}_{\bar{\underset{\sim}{\boldsymbol{\mathcal{C}}}}{}^1} - E\,\frac{\nu}{1-\nu^2}\underbrace{\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}}_{\bar{\underset{\sim}{\boldsymbol{\mathcal{C}}}}{}^2}, \tag{4.12}$$

where $\bar{\underset{\sim}{\boldsymbol{\mathcal{C}}}}{}^m$ is the inverse of $\bar{\boldsymbol{\mathcal{C}}}^m$, for this case. The expression for the elemental stiffness matrices (4.8), replicates the form of (4.11)

$$\boldsymbol{\mathcal{K}}(E,\nu) = \int \mathbf{B}^T \boldsymbol{\mathcal{C}}^{-1}(E,\nu)\, \mathbf{B}\, d\Omega = E\,\frac{1}{1-\nu^2}\int \mathbf{B}^T \bar{\underset{\sim}{\boldsymbol{\mathcal{C}}}}{}^1 \mathbf{B}\, d\Omega - E\,\frac{\nu}{1-\nu^2}\int \mathbf{B}^T \bar{\underset{\sim}{\boldsymbol{\mathcal{C}}}}{}^2 \mathbf{B}\, d\Omega. \tag{4.13}$$

Similar expressions are obtained when, for example, two materials with different parameters are considered. In this case, $n_p = 4$ and the coefficients C_i^m, with an i that does not correspond to the parameters of the material of the element, are equal to one.

In the examples that follow, the same steps are used to obtain $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{K}}$.

4.2.2 Two materials with different Young's moduli

When two materials with the same Poisson's ratio are considered, and their Young moduli are used as material parameters, $\boldsymbol{\mu} = \{E_1, E_2\}$, and $n_C = 1$. The constitutive relation for an element which is assigned to E_1 becomes

$$\boldsymbol{\mathcal{C}}(E_1, E_2) = C_1^1(E_1)\, C_2^1(E_2)\, \bar{\boldsymbol{\mathcal{C}}}^1 = \frac{1}{E_1} \underbrace{\begin{bmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 2(1+\nu) \end{bmatrix}}_{\bar{\boldsymbol{\mathcal{C}}}^1}, \tag{4.14}$$

where $C_2^1(E_2)$ is one, because the element is not assigned to E_2.

4.2.3 Two materials with different Poisson's ratio

In the case when the same Young modulus is considered, and different values of the Poisson ratios are used as material parameters, $\boldsymbol{\mu} = \{\nu_1, \nu_2\}$, and $n_C = 2$. The constitutive relation for an element which is assigned to ν_1 becomes

$$\boldsymbol{\mathcal{C}}(\nu_1, \nu_2) = C_1^1(\nu_1)\, C_2^1(\nu_2)\, \bar{\boldsymbol{\mathcal{C}}}^1 + C_1^2(\nu_1)\, C_2^2(\nu_2)\, \bar{\boldsymbol{\mathcal{C}}}^2 = \underbrace{\begin{bmatrix} \frac{1}{E} & 0 & 0 \\ 0 & \frac{1}{E} & 0 \\ 0 & 0 & \frac{2}{E} \end{bmatrix}}_{\bar{\boldsymbol{\mathcal{C}}}^1} + \nu_1 \underbrace{\begin{bmatrix} 0 & -\frac{1}{E} & 0 \\ -\frac{1}{E} & 0 & 0 \\ 0 & 0 & \frac{2}{E} \end{bmatrix}}_{\bar{\boldsymbol{\mathcal{C}}}^2}. \tag{4.15}$$

In this expression, $C_1^1(\nu_1)$, $C_2^1(\nu_2)$ and $C_2^2(\nu_2)$ are equal to one. The first because the mode is unaffected by the parameter, the others because we are dealing with an element that is not assigned to ν_2.

4.3 PGD Approximated Energy of the Error

Assuming a finite element mesh composed of n_e elements, the parametric form of the energy of the error can be written as:

$$\begin{aligned}\epsilon^2(\boldsymbol{\mu}) = \sum_{e=1}^{n_e} \epsilon^2_{[e]}(\boldsymbol{\mu}) = & \\ \sum_{e=1}^{n_e} \Big([\hat{\mathbf{u}}]_e^T(\boldsymbol{\mu})\, \boldsymbol{\mathcal{K}}_{[e]}(\boldsymbol{\mu})\, [\hat{\mathbf{u}}]_e(\boldsymbol{\mu}) + [\hat{\mathbf{s}}]_e^T(\boldsymbol{\mu})\, \boldsymbol{\mathcal{F}}_{[e]}(\boldsymbol{\mu})\, [\hat{\mathbf{s}}]_e(\boldsymbol{\mu}) - 2\, [\hat{\mathbf{s}}]_e^T(\boldsymbol{\mu})\, \boldsymbol{\mathcal{S}}_{[e]}\, [\hat{\mathbf{u}}]_e(\boldsymbol{\mu}) \Big). & \quad (4.16)\end{aligned}$$

Substituting (4.3), (4.4), (4.8) and (4.6) into (4.16), we have:

$$\begin{aligned}\epsilon^2(\boldsymbol{\mu}) = \sum_{e=1}^{n_e} \epsilon^2_{[e]}(\boldsymbol{\mu}) = & \\ \sum_{e=1}^{n_e} \Bigg(\left(\sum_{m=1}^{n_k} [\bar{\mathbf{u}}^m]_e \prod_{i=1}^{n_p} U_i^m(\mu_i) \right)^T \left(\sum_{c=1}^{n_{\underset{\sim}{C}}} \bar{\boldsymbol{\mathcal{K}}}^c_{[e]} \prod_{i=1}^{n_p} \underset{\sim}{C}{}_i^c(\mu_i) \right) \left(\sum_{k=1}^{n_k} [\bar{\mathbf{u}}^k]_e \prod_{i=1}^{n_p} U_i^k(\mu_i) \right) + & \\ \left(\sum_{m=1}^{n_s} [\bar{\mathbf{s}}^m]_e \prod_{i=1}^{n_p} S_i^m(\mu_i) \right)^T \left(\sum_{c=1}^{n_C} \bar{\boldsymbol{\mathcal{F}}}^c_{[e]} \prod_{i=1}^{n_p} C_i^c(\mu_i) \right) \left(\sum_{k=1}^{n_s} [\bar{\mathbf{s}}^k]_e \prod_{i=1}^{n_p} S_i^k(\mu_i) \right) - & \\ 2 \left(\sum_{m=1}^{n_s} [\bar{\mathbf{s}}^m]_e \prod_{i=1}^{n_p} S_i^m(\mu_i) \right)^T \bar{\boldsymbol{\mathcal{S}}}_{[e]} \left(\sum_{k=1}^{n_k} [\bar{\mathbf{u}}^k]_e \prod_{i=1}^{n_p} U_i^k(\mu_i) \right) \Bigg); & \quad (4.17)\end{aligned}$$

The products in the PGD solutions can be analytically expanded, so that from two PGD approximations, with n_1 and n_2 modes we obtain an equivalent expression for the product expressed as a PGD approximation with $n_1 n_2$ modes.

Given

$$g_1 = \sum_{m=1}^{n_1} \bar{g}_1^m \prod_{i=1}^{n_p} G_{1_i}^m(\mu_i) \qquad \text{and} \qquad g_2 = \sum_{k=1}^{n_2} \bar{g}_2^k \prod_{j=1}^{n_p} G_{2_j}^k(\mu_j), \tag{4.18}$$

we have

$$g_1\, g_2 = \left(\sum_{m=1}^{n_1} \bar{g}_1^m \prod_{i=1}^{n_p} G_{1_i}^m(\mu_i) \right) \left(\sum_{k=1}^{n_2} \bar{g}_2^k \prod_{j=1}^{n_p} G_{2_j}^k(\mu_j) \right) \tag{4.19}$$

$$= \sum_{m=1}^{n_1} \sum_{k=1}^{n_2} \bar{g}_1^m \bar{g}_2^k \prod_{i=1}^{n_p} \left(G_{1_i}^m(\mu_i)\, G_{2_i}^k(\mu_i) \right). \tag{4.20}$$

The application of this formula implies an increase of the number of modes that have to be considered, which can be substantial when both n_1 and n_2 are large. Generally, only some of these modes will affect significantly the solution. In this situation, it is possible to apply a PGD compression procedure [30] which reduce the number of modes, with a minimum effect in the quality of the compressed solution. Using this property, we have:

$$\epsilon^2(\boldsymbol{\mu}) = \sum_{e=1}^{n_e} \epsilon^2_{[e]}(\boldsymbol{\mu}) =$$

$$\sum_{e=1}^{n_e} \Bigg(\sum_{m=1}^{n_k} \sum_{k=1}^{n_k} \sum_{c=1}^{n_{\underset{\sim}{C}}} [\bar{\mathbf{u}}^{m^T}]_e \, \bar{\boldsymbol{\mathcal{K}}}^c_{[e]} \, [\bar{\mathbf{u}}^k]_e \prod_{i=1}^{n_p} U_i^m(\mu_i) \underset{\sim}{C}{}_i^c(\mu_i) U_i^k(\mu_i) + \sum_{m=1}^{n_s} \sum_{k=1}^{n_s} \sum_{c=1}^{n_C} [\bar{\mathbf{s}}^{m^T}]_e \, \bar{\boldsymbol{\mathcal{F}}}^c_{[e]} \, [\bar{\mathbf{s}}^k]_e \prod_{i=1}^{n_p} S_i^m(\mu_i) C_i^c(\mu_i) S_i^k(\mu_i) - 2 \sum_{m=1}^{n_s} \sum_{k=1}^{n_k} [\bar{\mathbf{s}}^{m^T}]_e \, \boldsymbol{\mathcal{S}}_{[e]} \, [\bar{\mathbf{u}}^k]_e \prod_{i=1}^{n_p} S_i^m(\mu_i) \, U_i^k(\mu_i) \Bigg). \quad (4.21)$$

If we define:

$$m_K = n_k^2 \, n_{\underset{\sim}{C}} \qquad m_F = n_s^2 \, n_C \qquad m_S = n_k \, n_s$$

$$\sum_{e=1}^{n_e} \underbrace{\sum_{m=1}^{n_k} \sum_{k=1}^{n_k} \sum_{c=1}^{n_{\underset{\sim}{C}}}}_{m_K} \left(\underbrace{[\bar{\mathbf{u}}^{m^T}]_e \, \bar{\boldsymbol{\mathcal{K}}}^c_{[e]} \, [\bar{\mathbf{u}}^k]_e}_{[\alpha_K^k]_e} \underbrace{\prod_{i=1}^{n_p} U_i^m(\mu_i) \underset{\sim}{C}{}_i^c(\mu_i) U_i^k(\mu_i)}_{\beta_K^k(\boldsymbol{\mu})} \right) = \sum_{e=1}^{n_e} \sum_{k=1}^{m_K} [\alpha_K^k]_e \, \beta_K^k(\boldsymbol{\mu}),$$

$$\sum_{e=1}^{n_e} \underbrace{\sum_{m=1}^{n_s} \sum_{k=1}^{n_s} \sum_{c=1}^{n_C}}_{m_F} \left(\underbrace{[\bar{\mathbf{s}}^{m^T}]_e \, \bar{\boldsymbol{\mathcal{F}}}^c_{[e]} \, [\bar{\mathbf{s}}^k]_e}_{[\alpha_F^f]_e} \underbrace{\prod_{i=1}^{n_p} S_i^m(\mu_i) C_i^c(\mu_i) S_i^k(\mu_i)}_{\beta_F^f(\boldsymbol{\mu})} \right) = \sum_{e=1}^{n_e} \sum_{f=1}^{m_F} [\alpha_F^f]_e \, \beta_F^f(\boldsymbol{\mu}),$$

$$\sum_{e=1}^{n_e} \underbrace{\sum_{m=1}^{n_s} \sum_{k=1}^{n_k}}_{m_S} \left(\underbrace{[\bar{\mathbf{s}}^{m^T}]_e \, \bar{\boldsymbol{\mathcal{S}}}^c_{[e]} \, [\bar{\mathbf{u}}^k]_e}_{[\alpha_S^s]_e} \underbrace{\prod_{i=1}^{n_p} S_i^m(\mu_i) \, U_i^k(\mu_i)}_{\beta_S^s(\boldsymbol{\mu})} \right) = \sum_{e=1}^{n_e} \sum_{s=1}^{m_S} [\alpha_S^s]_e \, \beta_S^s(\boldsymbol{\mu}),$$

we can simplify Equation (4.21) to:

$$\epsilon^2(\boldsymbol{\mu}) = \sum_{e=1}^{n_e} \epsilon^2_{[e]}(\boldsymbol{\mu}) = \sum_{e=1}^{n_e} \left(\sum_{k=1}^{m_K} [\alpha_K^k]_e \, \beta_K^k(\boldsymbol{\mu}) + \sum_{f=1}^{m_F} [\alpha_F^f]_e \, \beta_F^f(\boldsymbol{\mu}) - 2 \sum_{s=1}^{m_S} [\alpha_S^s]_e \, \beta_S^s(\boldsymbol{\mu}) \right). \quad (4.22)$$

Notice that the expression in (4.22) rearranges the equivalent expression in [32], merging sums and indices in order to produce a more compact form.

We also want to use the energy of the error to drive an adaptivity process, but (4.22) is not a good adaptivity indicator, as each set of parameters will lead to different values of the energy of the error in the element. An alternative to this is to perform an integral of the energy of the error in the parametric space as well. This will lead to:

$$\int_{\Omega_{\boldsymbol{\mu}}} \epsilon^2(\boldsymbol{\mu})\,\mathrm{d}\Omega_{\boldsymbol{\mu}} = \int_{\Omega_{\boldsymbol{\mu}}} \sum_{e=1}^{n_e} \epsilon^2_{[e]}(\boldsymbol{\mu})\,\mathrm{d}\Omega_{\boldsymbol{\mu}} =$$
$$\sum_{e=1}^{n_e}\left(\sum_{k=1}^{m_K}[\alpha_K^k]_e\int_{\Omega_{\boldsymbol{\mu}}}\left(\beta_K^k(\boldsymbol{\mu})\right)\mathrm{d}\Omega_{\boldsymbol{\mu}} + \sum_{f=1}^{m_F}[\alpha_F^f]_e\int_{\Omega_{\boldsymbol{\mu}}}\left(\beta_F^f(\boldsymbol{\mu})\right)\mathrm{d}\Omega_{\boldsymbol{\mu}} - 2\sum_{s=1}^{m_S}[\alpha_S^s]_e\int_{\Omega_{\boldsymbol{\mu}}}\left(\beta_S^s(\boldsymbol{\mu})\right)\mathrm{d}\Omega_{\boldsymbol{\mu}}\right). \tag{4.23}$$

For simplicity, we define (4.23) as:

$$\Psi_{\mathrm{global}^2} = \int_{\Omega_{\boldsymbol{\mu}}} \epsilon^2(\boldsymbol{\mu})\,\mathrm{d}\Omega_{\boldsymbol{\mu}}. \tag{4.24}$$

4.4 PGD Approximated Local Outputs

After solving the integrals in Equation (3.13), we can write the parametric form of the approximate local outputs as:

$$\begin{aligned}\tilde{\mathcal{L}}_k(\sigma_a(\boldsymbol{\mu})) = \frac{1}{2}\sum_{e=1}^{n_e}([\bar{\hat{\mathrm{s}}}]_e^T(\boldsymbol{\mu})\,\boldsymbol{\mathcal{F}}_{[e]}(\boldsymbol{\mu})\,[\hat{\mathbf{s}}]_e(\boldsymbol{\mu}) + [\bar{\hat{\mathrm{s}}}]_e^T(\boldsymbol{\mu})\,\boldsymbol{\mathcal{S}}_{[e]}\,[\hat{\mathbf{u}}]_e(\boldsymbol{\mu}) \\ - [\bar{\hat{u}}]_e^T(\boldsymbol{\mu})\,\boldsymbol{\mathcal{K}}_{[e]}(\boldsymbol{\mu})\,[\hat{\mathbf{u}}]_e(\boldsymbol{\mu}) + [\bar{\hat{u}}]_e^T(\boldsymbol{\mu})\,\boldsymbol{\mathcal{S}}_{[e]}^T\,[\hat{\mathbf{s}}]_e(\boldsymbol{\mu})).\end{aligned} \tag{4.25}$$

For simplicity, we suppress the dependency of $\tilde{\mathcal{L}}_k$ on σ_a and use $\tilde{\mathcal{L}}_k(\boldsymbol{\mu})$ instead. We can substitute Equations (4.3), (4.4), (4.8) and (4.6) into Equation (4.25) and use an equivalent definition for the α and β values, adding a tilde to indicate that these variables result from a combination of the real and virtual problems. The resulting equation is:

$$\tilde{\mathcal{L}}_k(\boldsymbol{\mu}) = \frac{1}{2}\sum_{e=1}^{n_e}\left(\sum_{\tilde{f}=1}^{\tilde{m}_F}[\tilde{\alpha}_F^{\tilde{f}}]_e\,\beta_F^{\tilde{f}}(\boldsymbol{\mu}) + \sum_{\tilde{s}=1}^{\tilde{m}_S}[\tilde{\alpha}_S^s]_e\,\tilde{\beta}_S^s(\boldsymbol{\mu}) - \sum_{\tilde{k}=1}^{\tilde{m}_K}[\tilde{\alpha}_K^k]_e\,\tilde{\beta}_K^{\tilde{k}}(\boldsymbol{\mu}) + \sum_{\tilde{s}=1}^{\tilde{m}_S}[\tilde{\alpha}_{ST}^s]_e\,\tilde{\beta}_{ST}^s(\boldsymbol{\mu})\right). \tag{4.26}$$

Notice that we can write the elemental contributions of the approximated local outputs in a form that is similar to (4.22), however, since the value of the local output in a single element does not have a physical meaning and will not be used in our computations, the corresponding equation is omitted.

4.5 PGD Approximated Bounds of the Local Output Error

If we consider that ϵ^2 comes from the real problem, we can define a similar $\bar{\epsilon}^2$ for the virtual problem. That is:

$$\bar{\epsilon}^2(\boldsymbol{\mu}) = \sum_{e=1}^{n_e} \bar{\epsilon}^2_{[e]}(\boldsymbol{\mu}) = \sum_{e=1}^{n_e} \left(\sum_{\bar{k}=1}^{\bar{m}_K} [\bar{\alpha}^{\bar{k}}_K]_e \; \bar{\beta}^{\bar{k}}_K(\boldsymbol{\mu}) + \sum_{\bar{f}=1}^{\bar{m}_F} [\bar{\alpha}^{\bar{f}}_F]_e \; \bar{\beta}^{\bar{f}}_F(\boldsymbol{\mu}) - 2 \sum_{s=1}^{\bar{m}_S} [\bar{\alpha}^{\bar{s}}_S]_e \; \bar{\beta}^{\bar{s}}_S(\boldsymbol{\mu}) \right) \tag{4.27}$$

We can write the parametric version of the square of the bound of the local output error of the average solution, from Equation (3.14), as:

$$\epsilon^2_{a_{local}}(\boldsymbol{\mu}) = \frac{1}{8} \sum_{e}^{n_e} \left(\underbrace{\epsilon^2(\boldsymbol{\mu})\,\bar{\epsilon}^2_{[e]}(\boldsymbol{\mu})}_{\bar{\eta}_{[e]}(\boldsymbol{\mu})} + \underbrace{\epsilon^2_{[e]}(\boldsymbol{\mu})\,\bar{\epsilon}^2(\boldsymbol{\mu})}_{\eta_{[e]}(\boldsymbol{\mu})} \right) \tag{4.28}$$

For simplicity, we will detail only with the first term $\bar{\eta}$ of Equation (4.28), but the computations for η are analogous. Substituting (4.22) and (4.27) into (4.28) we have:

$$\begin{aligned} \bar{\eta}_{[e]}(\boldsymbol{\mu}) = \sum_{e_i=1}^{n_e} & \left(\sum_{k=1}^{m_K} [\alpha^k_K]_{e_i} \; \beta^k_K(\boldsymbol{\mu}) + \sum_{f=1}^{m_F} [\alpha^f_F]_{e_i} \; \beta^f_F(\boldsymbol{\mu}) - 2 \sum_{s=1}^{m_S} [\alpha^s_S]_{e_i} \; \beta^s_S(\boldsymbol{\mu}) \right) \\ & \left(\sum_{\bar{k}=1}^{\bar{m}_K} [\bar{\alpha}^{\bar{k}}_K]_e \; \bar{\beta}^{\bar{k}}_K(\boldsymbol{\mu}) + \sum_{\bar{f}=1}^{\bar{m}_F} [\bar{\alpha}^{\bar{f}}_F]_e \; \bar{\beta}^{\bar{f}}_F(\boldsymbol{\mu}) - 2 \sum_{\bar{s}=1}^{\bar{m}_S} [\bar{\alpha}^{\bar{s}}_S]_e \; \bar{\beta}^{\bar{s}}_S(\boldsymbol{\mu}) \right) \end{aligned} \tag{4.29}$$

From $\bar{\eta}_{[e]}(\boldsymbol{\mu})$ and $\eta_{[e]}(\boldsymbol{\mu})$, we can substitute them in Equation (4.28). This will provide, for each set of parameters, the elemental contribution to the square of the bound of the local output error of the average solution.

We can also write the parametric version of the bound, from Equation (3.15), as:

$$\epsilon_{a_{local}}(\boldsymbol{\mu}) = \frac{1}{4} \sum_{e}^{n_e} \left(\underbrace{\bar{\epsilon}^2_{[e]}(\boldsymbol{\mu}) \frac{\epsilon(\boldsymbol{\mu})}{\bar{\epsilon}(\boldsymbol{\mu})}}_{\bar{\omega}_{[e]}(\boldsymbol{\mu})} + \underbrace{\epsilon^2_{[e]}(\boldsymbol{\mu}) \frac{\bar{\epsilon}(\boldsymbol{\mu})}{\epsilon(\boldsymbol{\mu})}}_{\omega_{[e]}(\boldsymbol{\mu})} \right) \tag{4.30}$$

We can substitute (4.22) and (4.27) in the first term of Equation (4.30), resulting in:

$$\bar{\omega}_{[e]}(\boldsymbol{\mu}) = \left(\sum_{\bar{k}=1}^{\bar{m}_K} [\bar{\alpha}_K^{\bar{k}}]_e \; \bar{\beta}_K^{\bar{k}}(\boldsymbol{\mu}) + \sum_{\bar{f}=1}^{\bar{m}_F} [\bar{\alpha}_F^{\bar{f}}]_e \; \bar{\beta}_F^{\bar{f}}(\boldsymbol{\mu}) - 2 \sum_{\bar{s}=1}^{\bar{m}_S} [\bar{\alpha}_S^{\bar{s}}]_e \; \bar{\beta}_S^{\bar{s}}(\boldsymbol{\mu}) \right)$$
$$\left(\frac{\sqrt{\sum_{e_i=1}^{n_e} \left(\sum_{k=1}^{m_K} [\alpha_K^k]_{e_i} \; \beta_K^k(\boldsymbol{\mu}) + \sum_{f=1}^{m_F} [\alpha_F^f]_{e_i} \; \beta_F^f(\boldsymbol{\mu}) - 2 \sum_{s=1}^{m_S} [\alpha_S^s]_{e_i} \; \beta_S^s(\boldsymbol{\mu}) \right)}}{\sqrt{\sum_{\bar{e}_i=1}^{n_e} \left(\sum_{\bar{k}=1}^{\bar{m}_K} [\bar{\alpha}_K^{\bar{k}}]_{\bar{e}_i} \; \bar{\beta}_K^{\bar{k}}(\boldsymbol{\mu}) + \sum_{\bar{f}=1}^{\bar{m}_F} [\bar{\alpha}_F^{\bar{f}}]_{\bar{e}_i} \; \bar{\beta}_F^{\bar{f}}(\boldsymbol{\mu}) - 2 \sum_{\bar{s}=1}^{\bar{m}_S} [\bar{\alpha}_S^{\bar{s}}]_{\bar{e}_i} \; \bar{\beta}_S^{\bar{s}}(\boldsymbol{\mu}) \right)}} \right) \tag{4.31}$$

The computation of the second term ω is analogous. After we compute both $\bar{\omega}_{[e]}(\boldsymbol{\mu})$ and $\omega_{[e]}(\boldsymbol{\mu})$ we can substitute them into Equation (4.31). This will lead to the average local output error for each set of parameters.

In terms of computing bounds, it is more interesting to use (4.30) than (4.28). This is because the squared bound is not a good bounding measure, as it does not have a physical meaning.

Neither (4.28) nor (4.30) are good adaptivity indicators, as each set of parameters will lead to different values of the local output error in each element. Again, a solution to this problem is to perform an integral in the parametric space of the local output error or of its squared form.

Notice that, in terms of adaptivity, it is more convenient to use the integral of (4.28) than the integral of (4.30). The first reason is because a mesh refined using the results coming from the squared error will give more importance to the regions with a higher error indicator. The second reason regards the computational implementation of these expressions. Performing the integral in the parametric domain can be computationally expensive, and the alternative to deal with this issue that we will present in the next section does not work for a squared root equation. The integral for the squared bound is:

$$\int_{\Omega_{\boldsymbol{\mu}}} \epsilon^2_{a_{local}}(\boldsymbol{\mu}) \, \mathrm{d}\Omega_{\boldsymbol{\mu}} = \frac{1}{8} \sum_{e=1}^{n_e} \left(\int_{\Omega_{\boldsymbol{\mu}}} \left(\bar{\eta}_{[e]}(\boldsymbol{\mu}) + \eta_{[e]}(\boldsymbol{\mu}) \right) \mathrm{d}\Omega_{\boldsymbol{\mu}} \right) \tag{4.32}$$

For simplicity, we define (4.32) as:

$$\Psi_{\mathrm{local}^2} = \int_{\Omega_{\boldsymbol{\mu}}} \epsilon^2_{a_{local}}(\boldsymbol{\mu}) \, \mathrm{d}\Omega_{\boldsymbol{\mu}}, \tag{4.33}$$

We will also define the total number of PGD modes n_m in a solution as:

$$n_m = (m_K + m_F + m_S), \tag{4.34}$$

and n_M the number of modes resulting for the product between a primal and virtual problem, such that:

$$n_M = n_m \, n_{\bar{m}}. \tag{4.35}$$

4.6 Implementation Details

There are several ways to numerically compute Equations (4.28), (4.30) and (4.32), which involve products of PGD solutions. The optimal approach to compute these products depends on the characteristics of the problem being studied. During the course of our implementation, we noticed a few aspects that are worth mentioning, as they can largely influence the computational complexity. Each approach has its advantages and disadvantages and we will summarize them into two categories: when the simulations are performed in a large parametric domain and when the number of modes for the PGD approximations is large. Of course, the concept of ”large” is entirely subjective, and we will try to give some examples for comparison. The results presented here are merely qualitative, as the code is not optimized for speed. The characteristics of the problem that is used are described in Section 6.

4.6.1 Large parametric domain

When performing the integral Ψ_{local^2}, the size of the parametric domain affects the computational cost. If the integrals are performed numerically as in (4.32), it will be necessary to compute all the possible combinations of parameters in order to perform the integral, since it is the last operation performed.

This means that, for instance, a problem with three parameters and a domain composed of $n_h = 50$ points, will have 1.25×10^5 combinations of parameters, which can lead to a very high computational cost. Naturally, increasing the number of integration points is desirable when seeking an accurate solution, as it will lead to a decrease in the errors coming from the numerical integration. These benefits occurs up to a certain threshold, after which other sources of error will have a greater influence in the solutions.

An alternative to (4.32) is to take advantage of the algebraic format of our solutions. Firstly, if we go back to Equation (4.29), we can manipulate it so that we are able to isolate the summations in e_i for their corresponding $[\alpha]_{e_i}$ and expand the product as:

$$
\begin{aligned}
&\bar{\eta}_{[e]}(\boldsymbol{\mu}) = \\
&\sum_{k=1}^{m_K}\left(\sum_{e_i=1}^{n_e}[\alpha_K^k]_{e_i}\right)\beta_K^k(\boldsymbol{\mu})\sum_{\bar{k}=1}^{\bar{m}_K}[\bar{\alpha}_K^{\bar{k}}]_e\,\bar{\beta}_K^{\bar{k}}(\boldsymbol{\mu}) + \sum_{k=1}^{m_K}\left(\sum_{e_i=1}^{n_e}[\alpha_K^k]_{e_i}\right)\beta_K^k(\boldsymbol{\mu})\sum_{\bar{f}=1}^{\bar{m}_F}[\bar{\alpha}_F^{\bar{f}}]_e\,\bar{\beta}_F^{\bar{f}}(\boldsymbol{\mu}) - \\
&2\sum_{k=1}^{m_K}\left(\sum_{e_i=1}^{n_e}[\alpha_K^k]_{e_i}\right)\beta_K^k(\boldsymbol{\mu})\sum_{\bar{s}=1}^{\bar{m}_S}[\bar{\alpha}_S^{\bar{s}}]_e\,\bar{\beta}_S^{\bar{s}}(\boldsymbol{\mu}) + \sum_{f=1}^{m_F}\left(\sum_{e_i=1}^{n_e}[\alpha_F^f]_{e_i}\right)\beta_F^f(\boldsymbol{\mu})\sum_{\bar{k}=1}^{\bar{m}_K}[\bar{\alpha}_K^{\bar{k}}]_e\,\bar{\beta}_K^{\bar{k}}(\boldsymbol{\mu}) + \\
&\sum_{f=1}^{m_F}\left(\sum_{e_i=1}^{n_e}[\alpha_F^f]_{e_i}\right)\beta_F^f(\boldsymbol{\mu})\sum_{\bar{f}=1}^{\bar{m}_F}[\bar{\alpha}_F^{\bar{f}}]_e\,\bar{\beta}_F^{\bar{f}}(\boldsymbol{\mu}) - 2\sum_{f=1}^{m_F}\left(\sum_{e_i=1}^{n_e}[\alpha_F^f]_{e_i}\right)\beta_F^f(\boldsymbol{\mu})\sum_{\bar{s}=1}^{\bar{m}_S}[\bar{\alpha}_S^{\bar{s}}]_e\,\bar{\beta}_S^{\bar{s}}(\boldsymbol{\mu}) - \\
&2\sum_{s=1}^{m_S}\left(\sum_{e_i=1}^{n_e}[\alpha_S^s]_{e_i}\right)\beta_S^s(\boldsymbol{\mu})\sum_{\bar{k}=1}^{\bar{m}_K}[\bar{\alpha}_K^{\bar{k}}]_e\,\bar{\beta}_K^{\bar{k}}(\boldsymbol{\mu}) - 2\sum_{s=1}^{m_S}\left(\sum_{e_i=1}^{n_e}[\alpha_S^s]_{e_i}\right)\beta_S^s(\boldsymbol{\mu})\sum_{\bar{f}=1}^{\bar{m}_F}[\bar{\alpha}_F^{\bar{f}}]_e\,\bar{\beta}_F^{\bar{f}}(\boldsymbol{\mu}) + \\
&\qquad 4\sum_{s=1}^{m_S}\left(\sum_{e_i=1}^{n_e}[\alpha_S^s]_{e_i}\right)\beta_S^s(\boldsymbol{\mu})\sum_{\bar{s}=1}^{\bar{m}_S}[\bar{\alpha}_S^{\bar{s}}]_e\,\bar{\beta}_S^{\bar{s}}(\boldsymbol{\mu})
\end{aligned}
\tag{4.36}
$$

Secondly, we can rearrange the summations and move the parametric terms $\beta(\boldsymbol{\mu})$ together:

$$
\begin{aligned}
&\bar{\eta}_{[e]}(\boldsymbol{\mu}) = \\
&\sum_{k=1}^{m_K}\sum_{\bar{k}=1}^{\bar{m}_K}\left(\sum_{e_i=1}^{n_e}[\alpha_K^k]_{e_i}\right)[\bar{\alpha}_K^{\bar{k}}]_e\,\beta_K^k(\boldsymbol{\mu})\,\bar{\beta}_K^{\bar{k}}(\boldsymbol{\mu}) + \sum_{k=1}^{m_K}\sum_{\bar{f}=1}^{\bar{m}_F}\left(\sum_{e_i=1}^{n_e}[\alpha_K^k]_{e_i}\right)[\bar{\alpha}_F^{\bar{f}}]_e\,\beta_K^k(\boldsymbol{\mu})\,\bar{\beta}_F^{\bar{f}}(\boldsymbol{\mu}) - \\
&2\sum_{k=1}^{m_K}\sum_{\bar{s}=1}^{\bar{m}_S}\left(\sum_{e_i=1}^{n_e}[\alpha_K^k]_{e_i}\right)[\bar{\alpha}_S^{\bar{s}}]_e\,\beta_K^k(\boldsymbol{\mu})\,\bar{\beta}_S^{\bar{s}}(\boldsymbol{\mu}) + \sum_{f=1}^{m_F}\sum_{\bar{k}=1}^{\bar{m}_K}\left(\sum_{e_i=1}^{n_e}[\alpha_F^f]_{e_i}\right)[\bar{\alpha}_K^{\bar{k}}]_e\,\beta_F^f(\boldsymbol{\mu})\,\bar{\beta}_K^{\bar{k}}(\boldsymbol{\mu}) + \\
&\sum_{f=1}^{m_F}\sum_{\bar{f}=1}^{\bar{m}_F}\left(\sum_{e_i=1}^{n_e}[\alpha_F^f]_{e_i}\right)[\bar{\alpha}_F^{\bar{f}}]_e\,\beta_F^f(\boldsymbol{\mu})\,\bar{\beta}_F^{\bar{f}}(\boldsymbol{\mu}) - 2\sum_{f=1}^{m_F}\sum_{\bar{s}=1}^{\bar{m}_S}\left(\sum_{e_i=1}^{n_e}[\alpha_F^f]_{e_i}\right)[\bar{\alpha}_S^{\bar{s}}]_e\,\beta_F^f(\boldsymbol{\mu})\,\bar{\beta}_S^{\bar{s}}(\boldsymbol{\mu}) - \\
&2\sum_{s=1}^{m_S}\sum_{\bar{k}=1}^{\bar{m}_K}\left(\sum_{e_i=1}^{n_e}[\alpha_S^s]_{e_i}\right)[\bar{\alpha}_K^{\bar{k}}]_e\,\beta_S^s(\boldsymbol{\mu})\,\bar{\beta}_K^{\bar{k}}(\boldsymbol{\mu}) - 2\sum_{s=1}^{m_S}\sum_{\bar{f}=1}^{\bar{m}_F}\left(\sum_{e_i=1}^{n_e}[\alpha_S^s]_{e_i}\right)[\bar{\alpha}_F^{\bar{f}}]_e\,\beta_S^s(\boldsymbol{\mu})\,\bar{\beta}_F^{\bar{f}}(\boldsymbol{\mu}) + \\
&\qquad 4\sum_{s=1}^{m_S}\sum_{\bar{s}=1}^{\bar{m}_S}\left(\sum_{e_i=1}^{n_e}[\alpha_S^s]_{e_i}\right)[\bar{\alpha}_S^{\bar{s}}]_e\,\beta_S^s(\boldsymbol{\mu})\,\bar{\beta}_S^{\bar{s}}(\boldsymbol{\mu})
\end{aligned}
\tag{4.37}
$$

Finally, as the parametric terms β do not depend on the elements, we can perform the integral directly in the parametric domain, so that:

$$
\begin{aligned}
&\int_{\Omega_{\boldsymbol{\mu}}} \bar{\eta}_{[e]}(\boldsymbol{\mu})\,\mathrm{d}\Omega_{\boldsymbol{\mu}} = \\
&\sum_{k=1}^{m_K}\sum_{\bar{k}=1}^{\bar{m}_K}\left(\sum_{e_i=1}^{n_e}[\alpha_K^k]_{e_i}\right)[\bar{\alpha}_K^{\bar{k}}]_e \int_{\Omega_{\boldsymbol{\mu}}}\Big(\beta_K^k(\boldsymbol{\mu})\,\bar{\beta}_K^{\bar{k}}(\boldsymbol{\mu})\Big)\,\mathrm{d}\Omega_{\boldsymbol{\mu}}+ \\
&\sum_{k=1}^{m_K}\sum_{\bar{f}=1}^{\bar{m}_F}\left(\sum_{e_i=1}^{n_e}[\alpha_K^k]_{e_i}\right)[\bar{\alpha}_F^{\bar{f}}]_e \int_{\Omega_{\boldsymbol{\mu}}}\Big(\beta_K^k(\boldsymbol{\mu})\,\bar{\beta}_F^{\bar{f}}(\boldsymbol{\mu})\Big)\,\mathrm{d}\Omega_{\boldsymbol{\mu}}- \\
2&\sum_{k=1}^{m_K}\sum_{\bar{s}=1}^{\bar{m}_S}\left(\sum_{e_i=1}^{n_e}[\alpha_K^k]_{e_i}\right)[\bar{\alpha}_S^{\bar{s}}]_e \int_{\Omega_{\boldsymbol{\mu}}}\Big(\beta_K^k(\boldsymbol{\mu})\,\bar{\beta}_S^{\bar{s}}(\boldsymbol{\mu})\Big)\,\mathrm{d}\Omega_{\boldsymbol{\mu}}+ \\
&\sum_{f=1}^{m_F}\sum_{\bar{k}=1}^{\bar{m}_K}\left(\sum_{e_i=1}^{n_e}[\alpha_F^f]_{e_i}\right)[\bar{\alpha}_K^{\bar{k}}]_e \int_{\Omega_{\boldsymbol{\mu}}}\Big(\beta_F^f(\boldsymbol{\mu})\,\bar{\beta}_K^{\bar{k}}(\boldsymbol{\mu})\Big)\,\mathrm{d}\Omega_{\boldsymbol{\mu}}+ \\
&\sum_{f=1}^{m_F}\sum_{\bar{f}=1}^{\bar{m}_F}\left(\sum_{e_i=1}^{n_e}[\alpha_F^f]_{e_i}\right)[\bar{\alpha}_F^{\bar{f}}]_e \int_{\Omega_{\boldsymbol{\mu}}}\Big(\beta_F^f(\boldsymbol{\mu})\,\bar{\beta}_F^{\bar{f}}(\boldsymbol{\mu})\Big)\,\mathrm{d}\Omega_{\boldsymbol{\mu}}- \\
2&\sum_{f=1}^{m_F}\sum_{\bar{s}=1}^{\bar{m}_S}\left(\sum_{e_i=1}^{n_e}[\alpha_F^f]_{e_i}\right)[\bar{\alpha}_S^{\bar{s}}]_e \int_{\Omega_{\boldsymbol{\mu}}}\Big(\beta_F^f(\boldsymbol{\mu})\,\bar{\beta}_S^{\bar{s}}(\boldsymbol{\mu})\Big)\,\mathrm{d}\Omega_{\boldsymbol{\mu}}- \\
2&\sum_{s=1}^{m_S}\sum_{\bar{k}=1}^{\bar{m}_K}\left(\sum_{e_i=1}^{n_e}[\alpha_S^s]_{e_i}\right)[\bar{\alpha}_K^{\bar{k}}]_e \int_{\Omega_{\boldsymbol{\mu}}}\Big(\beta_S^s(\boldsymbol{\mu})\,\bar{\beta}_K^{\bar{k}}(\boldsymbol{\mu})\Big)\,\mathrm{d}\Omega_{\boldsymbol{\mu}}- \\
2&\sum_{s=1}^{m_S}\sum_{\bar{f}=1}^{\bar{m}_F}\left(\sum_{e_i=1}^{n_e}[\alpha_S^s]_{e_i}\right)[\bar{\alpha}_F^{\bar{f}}]_e \int_{\Omega_{\boldsymbol{\mu}}}\Big(\beta_S^s(\boldsymbol{\mu})\,\bar{\beta}_F^{\bar{f}}(\boldsymbol{\mu})\Big)\,\mathrm{d}\Omega_{\boldsymbol{\mu}}+ \\
&\qquad 4\sum_{s=1}^{m_S}\sum_{\bar{s}=1}^{\bar{m}_S}\left(\sum_{e_i=1}^{n_e}[\alpha_S^s]_{e_i}\right)[\bar{\alpha}_S^{\bar{s}}]_e \int_{\Omega_{\boldsymbol{\mu}}}\Big(\beta_S^s(\boldsymbol{\mu})\,\bar{\beta}_S^{\bar{s}}(\boldsymbol{\mu})\Big)\,\mathrm{d}\Omega_{\boldsymbol{\mu}}
\end{aligned}
\tag{4.38}
$$

Using Equation (4.38) simplifies the integrals, as in this format they can be computed as one dimensional products of integrals. The drawback of this approach is that it is now required to perform products between modes of the real and virtual solutions, which can lead to a very high total number of PGD modes n_M. This product is not mandatory if we work with Equation (4.32). Another drawback is that we cannot extend this approach to the integral of (4.30), as the integral of the square root is not the square root of the integral.

Table 4.1 presents the speed up obtained by using Equation (4.38) instead of (4.29), with different number of points n_h to describe the parametric domain and different number of PGD modes, with n_e equal to 20 elements. It is observed that the speed up increases quickly as we increase the number of points. One important aspect is that the performance of the simulations using the

TABLE 4.1: Speed up obtained for the computations performed using the rearranged form of Ψ_{local^2}. Results for different number of integration points n_h and number of PGD modes.

	speed up	
n_h	$n_m \times n_{\bar{m}} = 300 \times 300$	$n_m \times n_{\bar{m}} = 7500 \times 7500$
10	1.611	0.018
25	4.333	0.110
50	14.04	0.432
100	22.72	1.372
200	65.97	4.139

integrals as in (4.38) are largely influenced by the number of modes for the PGD approximations. This method is ideal when the ratio between the number of points and the expected number of modes in the simulation is large enough. The number of elements in the mesh does not seem to have a relevant influence in the variation of the speed up.

4.6.2 Large number of modes

The computation of the bound of the error of the local output $\epsilon^2_{a_{local}}$ requires a product between two quantities that are approximated using a PGD, resulting in n_m or $n_{\bar{m}}$ modes. When the form of the integral (4.38) is used, these quantities are further multiplied. This means that the total number of modes n_M to compute the integral of the local output error can increase rapidly. For instance, a problem with a single parameter, in which the displacements $\hat{\mathbf{u}}$ and stresses $\hat{\mathbf{s}}$ for the real and virtual problem are described using 20 PGD modes will result in:

$$\begin{aligned} n_M &= n_m \times n_{\bar{m}} \\ &= (m_K + m_F + m_S)(\bar{m}_K + \bar{m}_F + \bar{m}_S) \\ &= (n_k^2 \, n_{\underset{\sim}{C}} + n_s^2 \, n_C + n_k \, n_s)(n_{\bar{k}}^2 \, n_{\underset{\sim}{C}} + n_{\bar{s}}^2 \, n_C + n_{\bar{k}} \, n_{\bar{s}}) \\ &= (20^2 \times 1 + 20^2 \times 1 + 20 \times 20)(20^2 \times 1 + 20^2 \times 1 + 20 \times 20) \\ &= 1.44 \times 10^6 \end{aligned}$$

Much of the information contained in these modes is redundant for the description of the error. This additional information is usually a result of the lack of orthogonality between the PGD modes [33]. A consequence of having a high number of modes is that the system memory needed to store such solutions must be large. Also, if one seeks to check an specific solution, the time necessary to perform the simple PGD summation of modes is no longer marginal.

A way to effectively work around this problem is to apply a compression to the solutions. The `pgdCompression` tool is part of the `algebraicPGDtools` and uses a least-square projection of the original solution into the same approximation space to try to obtain a sufficiently accurate approximation with fewer modes.

A drawback of using the compression in the context of the dual error analysis is that the solutions bounding properties are lost. This can be a problem when looking for specific solutions, although a sufficient small tolerance for the compression process will still result in accurate results.

For our problem, we can apply the compression at different steps and in different ways. We could, for example, compress the inputs (4.3) and (4.4) even before starting the error computations. We chose to apply the compression on the output from (4.29) which is a scalar and therefore will require less information to be described. We can also decide if we want to compress all elements of (4.29) at once or perform a different compression for each element.

Applying the compression to all elements at once is beneficial when working with fine meshes, as the number of operations is lower, but the results obtained have a much lower accuracy than those obtained using the compression in each element. The drawback of using the compression in each element is that, as each element will have its own PGD definition, the parametric domain β will also depend on the elements, what makes it impossible to use the modified integral (4.38).

Furthermore, the benefits of applying the compression simultaneously to all elements, when we have a large number of elements, quickly diminishes as we increase the number of PGD modes in the original solution. Table 4.2 presents the new number of modes n_{m^c} and $n_{\bar{m}^c}$ obtained for the real and virtual solutions after using the compression for all elements at once. We consider a parametric domain composed of $n_h = 50$ points and a number of elements equal to $n_e = 20$.

Table 4.2: Number of modes obtained for the computations performed using the pgdCompression routine. Results for different number of modes n_m and $n_{\bar{m}}$, and different number of mesh elements n_e.

$n_m \times n_{\bar{m}}$	$n_{m^c} \times n_{\bar{m}^c}$
300×300	20×14
1875×1875	19×14
7500×7500	21×14
30000×30000	21×14

The number of modes obtained using the compression is smaller than the number of PGD modes in the original solution. It is also noticeable that increasing the number of modes in the original solution has little influence in the resulting number of compressed modes, since the additional modes do not significantly affect the final compressed solution. The relative difference of the results with and without compression is also dependent on the tolerance of the compression process, but is well within an acceptable range, with the default tolerance of 1×10^{-3} leading to results with less than 0.0002% difference. This relative difference depends on the complexity of the problem being studied, but by decreasing the compression tolerance we can decrease the relative difference values, with the drawback of requiring more modes in the compressed solutions.

Chapter 5

Mesh Adaptivity

The simulations based on the finite element method rely on the partition of a continuous problem domain into smaller elements, that when combined result in a mesh in which the solutions are computable. If these elements are distributed in a smarter way in the mesh, for instance with more elements in the regions with a sharper variation of the solution,, it is possible to obtain better solutions at a lower computational cost. We call the process of using information from previous solution to obtain new meshes a *mesh refinement proccess* or *mesh adaptivity.* This chapter explains the bases for a simple adaptive process and how to implement it in our simulations. We also explain the criteria used to stop the PGD computations and the adaptive process.

Adaptivity mesh techniques are designed to improve approximated solutions by modifying either the element mesh configuration or their degrees of freedom [34]. One of the key points in the mesh adaptivity process is the proper definition of the relation between the solution convergence and the element size and/or degree of the approximation functions. This convergence dependency is expected to be precise when a smooth solution is being studied and the mesh that represents the solution domain has a large enough number of elements.

We can usually divide the adaptivity process in three categories: the modification of the size of the elements in a mesh (h- adaptivity); the modification of the approximation degree for the functions (p- adaptivity); and a combination of both these process (hp- adaptivity). This thesis explores the aspects of the h- and p- adaptivities, but not the combination of both.

We know that the convergence of the approximation of the solution of a smooth problem is expressed, under optimal conditions, as [35]:

$$\epsilon^2 = Ch^\lambda, \tag{5.1}$$

where h corresponds to a dimension of the mesh, for example the average size of the elements of the mesh or the number of elements, λ is the convergence rate, related to the degree of the approximations, and C is a problem dependent constant. Notice that this expression only holds in the asymptotic range.

The energy of the error is used directly to drive an adaptivity process aimed at finding a new finite element mesh, with its new error ϵ^2_{new} and the new dimension of the mesh h_{new}. This is a straightforward process because ϵ^2 is a sum of elemental contributions. We know, from equation (5.1) that:

$$C = \frac{\epsilon^2}{h^\lambda}. \tag{5.2}$$

The constant C will not change for the refined mesh, so that:

$$C = \frac{\epsilon^2_{\text{new}}}{h^\lambda_{\text{new}}}. \tag{5.3}$$

Combining equations (5.2) and (5.3), we have the new dimension of the mesh:

$$h_{\text{new}} = \left(\frac{\epsilon^2_{\text{new}}}{\epsilon^2}\right)^{\frac{1}{\lambda}} h. \tag{5.4}$$

Among the different approaches that can be used to guide the refinement process [36], we use the one that tries to impose an even distribution of the density of the error among the elements. We achieve that by requiring that the density of the energy of the error for the new mesh is equal to a fraction of the density of the old one. Considering that we want a reduction of factor α in the error ϵ^2, for $\Omega_{[e]}$, the domain of element e of the original mesh, we have:

$$\epsilon^2_{\text{new}_{[e]}} = \alpha \, |\Omega|_{[e]} \, \frac{\epsilon^2}{|\Omega|}, \tag{5.5}$$

where $|\Omega|_{[e]}$ is the area of element e and $|\Omega|$ is the total area of the domain. Then, the new mesh size in the domain of element e is

$$h_{\text{new}_{[e]}} = \left(\frac{\epsilon^2_{\text{new}_{[e]}}}{\epsilon^2_{[e]}}\right)^{\frac{1}{\lambda}} h_{[e]} = \left(\alpha \frac{|\Omega|_{[e]}}{|\Omega|} \frac{\epsilon^2}{\epsilon^2_{[e]}}\right)^{\frac{1}{\lambda}} h_{[e]}. \tag{5.6}$$

We implement this refinement procedure in an application for 2D linear elastic problems, with parametric values for Young's modulus and/or the Poisson ratio. In this context, instead of using the energy of the error, ϵ^2, we use the integral in the parametric domain of the corresponding PGD approximations, Ψ, whose bounding properties are the same (for simplicity, we will omit the *global*^{2} or *local*^{2} subscripts in this section, as for the sake of mesh adaptivity Ψ can be either the global or local integrals of the error). In this parametric problem we seek approximations for the stresses and displacements and bounds of their global energy error.

We start by defining the geometry and material properties for the problem we want to work with. Once we have the geometry defined, we generate a coarse mesh for that geometry. The solutions obtained for this mesh start an adaptive process, oriented according to the rules given in the previous section. We work with triangular elements and in this study use the `gmsh` software to mesh the geometry automatically [37].

In the adaptive process, for each mesh we start by computing the stiffness matrix and the vector of applied forces for the compatible solution and the flexibility matrix and the vector of boundary forces for the equilibrated solution. We compute these in a separated form, as presented in Section 4, using `Mathematica` to generate the transformation matrices defined in Section 2 and `matlab` to compute the resulting outputs. The matrices and vectors for the compatible and equilibrated solutions, already in a separated form, are used as input for the `algebraicPGDtools` routines.

Stopping criteria are needed for the PGD approximations and the adaptivity process. We observed that the stopping criteria provided by the `algebraicPGDtools` code is a reliable way to determine when the PGD solution has stagnated, but because we are independently computing pairs of solution, it may lead to more modes than necessary for our application [7, 21]. We take advantage of also computing the energy of the error to use it as our PGD stopping criterion, assessing the stagnation of the PGD solutions with the relative difference Ψ^R, which measures the relevance for the error of adding additional modes to the PGD approximations. Assuming a tolerance τ_{PGD} and that the integral of the energy of the error of a PGD approximation, Ψ, with up to m modes is defined as Ψ_m, we want:

$$\Psi^R = \frac{\Psi_{m+1} - \Psi_m}{\Psi_m} \geq \tau_{PGD}. \tag{5.7}$$

We recompute Ψ^R for every additional PGD mode. Notice that computing the energy of the error Ψ becomes more costly as the number of modes in the PGD increases, so for problems that we know beforehand that will lead to a large number of PGD modes, it is best to decrease the frequency with which we compute the relative error. For the applications presented, we compute the relative error Ψ^R after every two PGD modes and stop the process when it reaches the defined tolerance τ_{PGD}.

The second stopping criteria deals with the adaptivity process and is defined as the ratio between the error Ψ and the energy of the problem E. We define it as:

$$\Psi^E = \frac{\Psi}{E} \leq \tau_{ERR}. \tag{5.8}$$

This relative integral of the energy error Ψ^E deals with the error in a global manner, giving results that are comparable among different problems. The energy E is found by averaging the energies of the compatible and equilibrated solution. We compute it for every new mode and update it with the smallest value found. This value is chosen because it is one of the best approximations for the real energy of the problem, considering no additional information is known beforehand.

Just like we did for the error Ψ^R, we will want Ψ^E to be lower than a defined tolerance τ_{ERR}. When this value is obtained we can stop the adaptivity process. We also use the maximum number of elements in the mesh as an additional stopping criteria for the adaptivity process, since too many elements can lead to lack of system memory for the simulations. Once one these two criteria is reached, we end our adaptivity process and can start analysing the results. If none of the criteria is met, we generate a new mesh using the concepts explained in Section 6 and restart the process.

Chapter 6

Numerical Applications

This chapter aims to apply the concepts described to a linear elastic problems.

We begin with a simplified, one dimensional example, for which we have a known exact solution. In this problem, we highlight the advantages of using the energy of the error as an error indicator and explore the possibilities of using it as a parametric refinement indicator.

We then move to a two dimensional problem, in which we further explore the use of the energy of the error as a mesh adaptivity indicator. We present the stresses and displacements obtained in a problem with a geometric singularity and how the error behaves in a case like this.

Finally, we look at the error in terms of local outputs and their bounds, with an example that illustrates the use of the energy of the error for a local mesh refinement. We demonstrate the advantages of using a goal-oriented refinement instead of an uniform one and we show the reliability of the bounds obtained when compared with a reference solution.

6.1 1D Problem and Model Validation

6.1.1 Practical Aspects of the One Dimensional Discretization of the Physical Domain

Before we present the results for the one dimensional problem proposed, which will be presented in Figure 6.4, we must note that in this example we use a description of the PGD approximations that is different from the one used in in the `algebraicPGDtools`Ṫhe approach used allows us

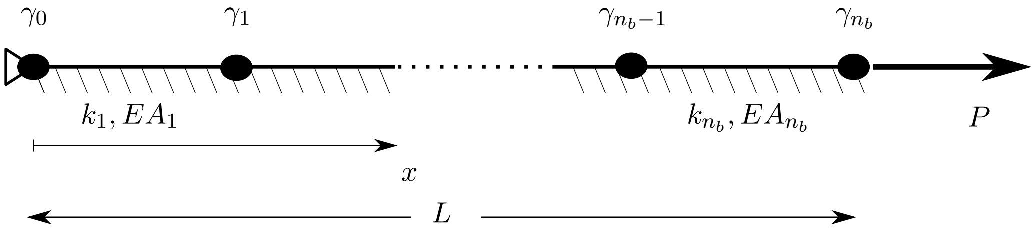


Figure 6.1: Linear elastic straight bar on an elastic support, with n_b sections. The bar is fixed at the start and has a concentrated load at the end.

more flexibility to work and modify the PGD equations, with the drawback of using a code that is much less efficient.

We consider a simple problem for which an analytic solution can be obtained: a linear elastic straight bar divided in n_b sections, each section with a length (γ), an uniform axial stiffness (EA) and an uniform elastic support (k), connected in sequence, which form a one dimensional lattice structure.

Although arbitrary imposed displacements or applied forces may be considered at the nodes that connect the different sections, we restrict our study to the problem where the displacement (Δ) is fixed at the start of the first bar and a non-zero force (P) is considered at the free end of the bar, as presented in Figure 6.1.

We want to obtain a solution for this problem, expressed in terms of either the displacements (u) or of the axial forces (N), that is characterized by the values of γ, EA and k for each section, plus the value of the imposed displacement and of the applied force.

A global coordinate system ($x \in [0, L]$) is used, where the coordinates of the initial and final nodes of section b are γ_{b-1} and γ_b, with $\gamma_0 = 0$ and $\gamma_{n_b} = L$. Therefore, the geometric parameters correspond to the γ_i, with $i = 1, \ldots, (n_b - 1)$.

Each section, which is divided in $n_{e[b]}$ elements, uses an intermediate coordinate system ($\bar{x}_{[b]} \in [0, 1]$). The coordinates of the initial and final nodes of element e are $\bar{\gamma}_{[b]e-1}$ and $\bar{\gamma}_{[b]e}$, with $\bar{\gamma}_{[b]0} = 0$ and $\bar{\gamma}_{[b]n_e} = 1$. In the following, we replace $\bar{\gamma}_{[b]e}$ with $\bar{\gamma}_e$, $\bar{x}_{[b]e}$ with $\bar{x}_e$ and $\bar{x}_{[b]}$ with $\bar{x}$, unless the reference to the section of the element is necessary.

In the elements domain, a local coordinate system is used ($\xi \in [-1, 1]$), which is linearly mapped to $\bar{x}$, which is then linearly mapped to x.

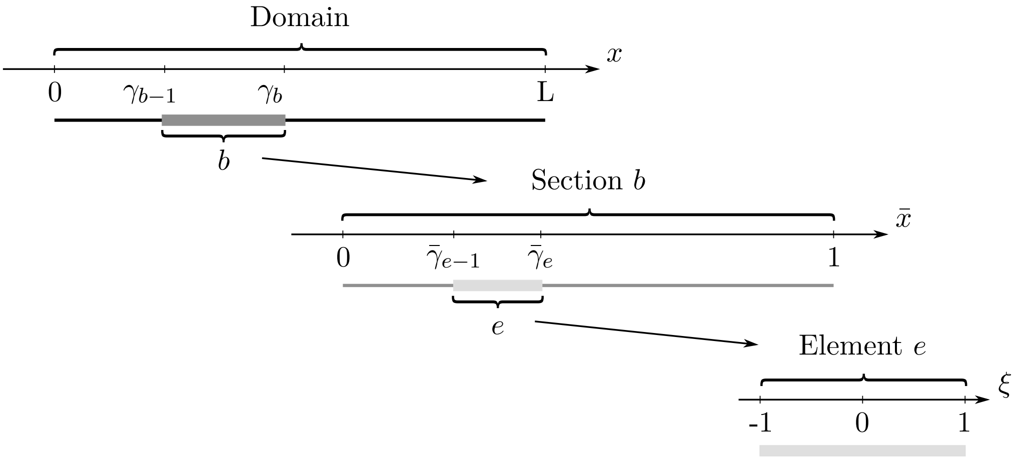


FIGURE 6.2: Geometric mappings: from bar domain to section and from section domain to element.

Using ϕ_0 and ϕ_1 to represent the linear interpolation functions associated with the end nodes of an interval we have:

$$\bar{x}(\xi) = \phi_0(\xi)\,\bar{\gamma}_{e-1} + \phi_1(\xi)\,\bar{\gamma}_e \quad \text{and} \quad x(\bar{x}) = \phi_0(\bar{x})\,\gamma_{b-1} + \phi_1(\bar{x})\,\gamma_b. \tag{6.1}$$

These transformations are illustrated in Figure 6.2.

Each section may be divided in several finite elements. Normally, for problems with fixed geometry, only one mapping is used, from the frame of element (ξ) to the global coordinates (x).

Since both mappings are linear the Jacobian of each transformation is constant, and the Jacobian of the double mapping, J, is equal to their product. The derivative of an arbitrary function, f, is obtained as

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\mathrm{d}f}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}\bar{x}}\frac{\mathrm{d}\bar{x}}{\mathrm{d}x} = \frac{2}{(\bar{\gamma}_e - \bar{\gamma}_{e-1})}\frac{1}{(\gamma_b - \gamma_{b-1})}\frac{\mathrm{d}f}{\mathrm{d}\xi} = \frac{1}{J_e}\frac{\mathrm{d}f}{\mathrm{d}\xi}, \tag{6.2}$$

and its integral is

$$\int f\ \mathrm{d}x = \int f J_e\ \mathrm{d}\xi. \tag{6.3}$$

Our basis, in local coordinates, are obtained by combining the linear interpolation functions with the Legendre polynomials $\mathcal{L}_i(\xi)$:

$$\begin{cases} \phi_0(\xi) = \frac{1-\xi}{2}, \\ \phi_1(\xi) = \frac{1+\xi}{2}, \\ \phi_i(\xi) = \phi_0(\xi)\ \phi_1(\xi)\ \mathcal{L}_{i-2}(\xi), \quad \text{for } i > 1 \end{cases} \tag{6.4}$$

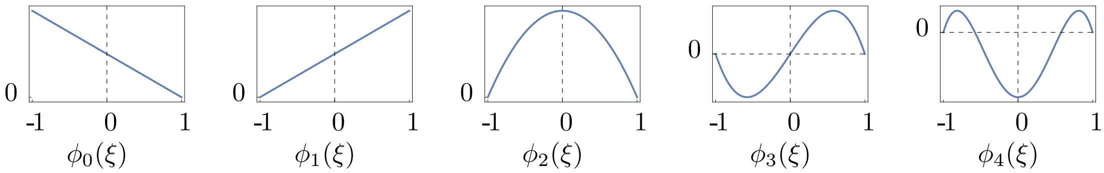


FIGURE 6.3: Basis for one dimensional approximations.

The complete system of equations for each problem (the assembled stiffness or flexibility matrices) requires additional constraints that impose displacement continuity or equilibrium of normal forces between sections. This corresponds to impose:

- continuity of displacements and (weak) equilibrium of nodal forces, or
- equilibrium of normal forces and (weak) continuity of the displacements.

For the compatible formulation, to have a continuous displacement at the node between sections b and $b+1$, we must impose that:

$$u_{[b]n_e}(1) = u_{[b+1]1}(-1) \Longrightarrow \hat{u}_{[b]n_e,1} = \hat{u}_{[b+1]1,0}. \tag{6.5}$$

For the equilibrated formulation, the process is complementary: we impose continuity of the nodal forces and the displacement of the interface is accounted for in the weak compatibility conditions of each adjacent bar. The matrices involved are the same.

Note that, in case we wanted to consider a concentrated force applied at a node, or a relative imposed displacement, the equations need to be modified accordingly.

In the practical implementation of the problem, we apply these mapping considerations for all the equations described so far, but we will omit them from now on to reduce repetition of the text.

6.1.2 Characterization of the Test Case

To apply the concepts presented here, we consider the problem shown in Figure 6.4, with the following characteristics:

- The bar is composed of two sections, with a total summed length L of one;

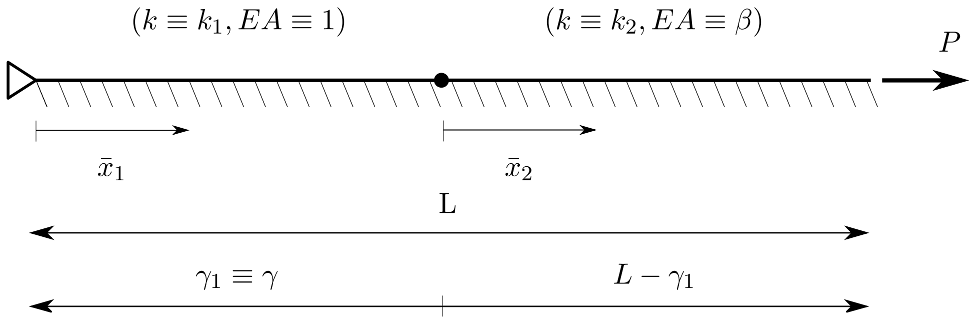


FIGURE 6.4: A bar with two sections, connected at an intermediate point, defined by γ.

- Each section has its own support stiffness k_b;
- The axial stiffness EA for the first section has a unit value;
- The axial stiffness of the second section is equal to β;
- The length of the first section is equal to γ.

The parameters for the PGD with their respective limits are:

$$\begin{cases} \mu_1 \equiv \mu_{k_1} \equiv k_1, & \text{with } k_1 \in [0.1, 10]; \\ \mu_2 \equiv \mu_{k_2} \equiv k_2, & \text{with } k_2 \in [0.1, 10]; \\ \mu_3 \equiv \mu_{\beta} \equiv \beta, & \text{with } \beta \in [0.1, 10]; \\ \mu_4 \equiv \mu_{\gamma} \equiv \gamma, & \text{with } \gamma \in [0.4, 0.6]. \end{cases} \tag{6.6}$$

The calculations are performed considering four actions:

- An applied force at the free end ($P_{x=L} = 1$, $\Delta_{x=0} = 0$);
- An imposed displacement at the support ($P_{x=L} = 0$, $\Delta_{x=0} = 1$);
- An applied load at the connection between the two bars ($P_{x=\gamma} = 1$);
- A relative displacement imposed between the two bars ($\Delta_{x=\gamma} = 1$).

The calculations are performed considering an applied force at the free end ($P_{x=L} = 1$, $\Delta_{x=0} = 0$).

Unless otherwise stated, the polynomial approximations are linear for all parameters. Notice that these degrees can lead to solutions which are not accurate, but they serve the purpose of this thesis, which is to identify regions where the simulations can be improved.

The tolerances τ adopted for the fixed point iteration scheme and the PGD enrichment process are $\tau_{fix} = 1 \times 10^{-3}$ and $\tau_{pgd} = 1 \times 10^{-6}$, respectively, with the maximum amount of iterations being 3 for the fixed point and 3001 for the PGD. Setting such low number of iterations for the fixed point scheme may lead to more terms for the convergence of the PGD, but drastically decreases the computational time, which is a phenomenon also observed in [7, 21], but should not be generalized for all problems, as this behavior is case dependent.

To further improve the convergence of the solutions, the limits for the parameters k_1, k_2 and β were mapped to a logarithmic scale $k_1 = 10^{\hat{k}_1}$, $k_2 = 10^{\hat{k}_2}$ and $\beta = 10^{\hat{\beta}}$, with ranges $\hat{k}_1 \in [-1, 1]$, $\hat{k}_2 \in [-1, 1]$ and $\hat{\beta} \in [-1, 1]$. This mapping does not affect the PGD approximations, serving only to diminish the influence of the edges in the solutions. Additional simulations must be performed with different limits to assess the effect of this modification.

The integral in the parametric domain for the potential energies will be defined as:

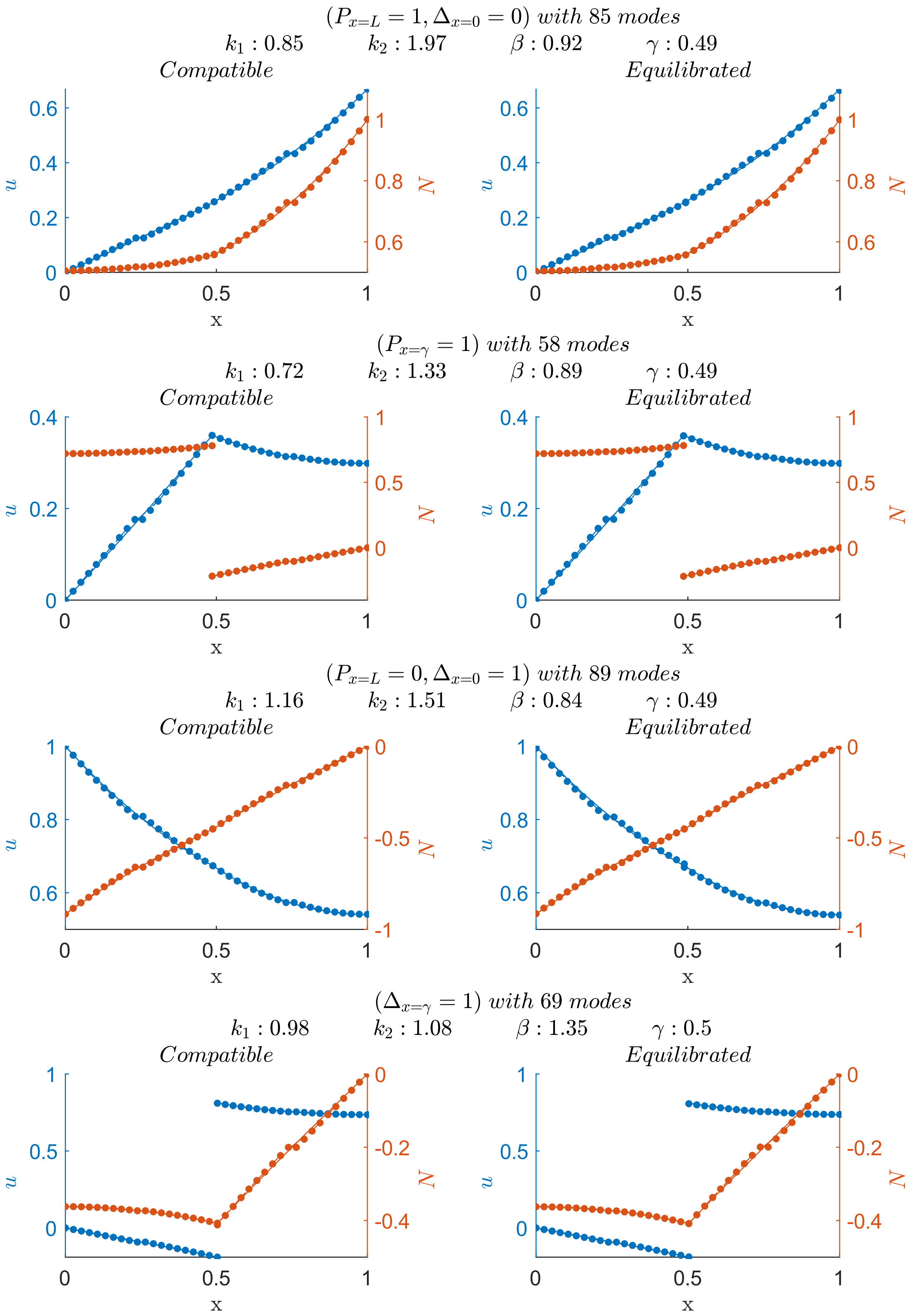
$$\Upsilon = \int_{\Omega_\mu} \Pi(\mu)\, \mathrm{d}\Omega_\mu. \tag{6.7}$$

We represent the integral in the parametric domain for the strain energies as $\int \mathcal{U}$. We will use a c superscript to indicate a variable coming from a complementary solution. If the variable is compute using the average solution, we will use an a.

Notice that, as we are only working with a global problem, we will drop all the references for *global* in the subscripts.

The first result presents the exact, FEM and PGD solutions in terms of the displacement and axial force along the length of the bars. The polynomial degrees for this simulation were set to 9 for the parameters space and 10 for the physical space. The physical space was also divided into two intervals. This choice of parameters are intended to provide a good approximation of the solutions, which is necessary for better visualization. Figure 6.5 presents the results for the displacement u and axial forces N, with values that minimize the relative error of the integral of the energy for each of the different load types. The approximations for the compatible and equilibrated models are presented side by side. The results for the PGD, the exact and FEM solutions are very close to each other, with the lowest average relative error being $3.14 \times 10^{-10}\%$ for $(P_{x=L} = 1, \Delta_{x=0} = 0)$.

Notice that the simulations that impose a load at the connection between the two sections show a discontinuity between the two solutions, as expected for this type of problem.



$(P_{x=L} = 1, \Delta_{x=0} = 0)$ *with* 85 *modes*

$k_1 : 0.85$ $k_2 : 1.97$ $\beta : 0.92$ $\gamma : 0.49$

Compatible *Equilibrated*

$(P_{x=\gamma} = 1)$ *with* 58 *modes*

$k_1 : 0.72$ $k_2 : 1.33$ $\beta : 0.89$ $\gamma : 0.49$

Compatible *Equilibrated*

$(P_{x=L} = 0, \Delta_{x=0} = 1)$ *with* 89 *modes*

$k_1 : 1.16$ $k_2 : 1.51$ $\beta : 0.84$ $\gamma : 0.49$

Compatible *Equilibrated*

$(\Delta_{x=\gamma} = 1)$ *with* 69 *modes*

$k_1 : 0.98$ $k_2 : 1.08$ $\beta : 1.35$ $\gamma : 0.5$

Compatible *Equilibrated*

FIGURE 6.5: Displacements u (blue) and axial forces N (orange) for different load types, using variables values that lead to the best approximations. The solid lines are the exact solution, the dashed lines the FEM approximations and the dots the PGD approximations. The polynomial degrees are 9 for the parameters space and 10 for the physical domain, with the physical domain being divided using two elements per section.

The solutions with the poorest approximations for this simulation are presented in Figure 6.6, where the largest relative error of the integral of the energy among the four solutions is 0.3322% for ($\Delta_{x=\gamma} = 1$). The results presented in the figure highlight that a solution obtained from an equilibrated system approximates better with the axial forces N while dealing poorly with the displacements u. A complementary behavior is observable for the solutions computed from the compatible system, with better approximations for the displacement than for the axial forces. Observe that the parameters values that lead to these solutions are mostly at the edge of their respective domains. It is important to notice that the approximations for the FEM are still close to the exact solutions, since they are not influenced by the approximations in the parameters space.

We now present solutions in terms of the error in energy and the energy of the error. For simplicity, from this point on, the calculations are performed considering only the example with an applied force at the free end ($P_{x=L} = 1$, $\Delta_{x=0} = 0$). The simulations were performed using quadratic approximations in space, in order to better visualize the bounds of the solutions. The bounding characteristics discussed in the previous section are observed in Figure 6.7, which presents selected results for our problem. The left part of the Figure shows the integral of the energies of the PGD and FEM approximations, while the right part shows the integral of the energy of the error.

The energies for the FEM solutions are very close to the exact solutions and are represented as details in the figure. Notice that the exact energy of the error is zero and therefore it is not shown in the figure while the exact integral of the error in energy is greater than zero and is presented in the left figure as the a black line. The energies of the compatible and equilibrated solutions will have a convergence from above and bellow, respectively, when the problem is force-driven, which is the behavior observed. On the other hand, for the bound of the energy of the error we have an upper bound, meaning that it should be always greater than the values found for the exact solutions. From the figure it is also possible to see that, as the number of PGD modes increases, the two complementary PGD solutions converge to the approximation of the FEM, always preserving their bounding characteristics.

Figure 6.8 shows the relative difference between the approximations obtained from the PGD and FEM models. The integral of the energies of the PGD model will converge to the FEM energy as the number of modes in the solution increases. The results obtained for the integral of the energy of the solutions and the energy of the error are virtually the same. This sort of

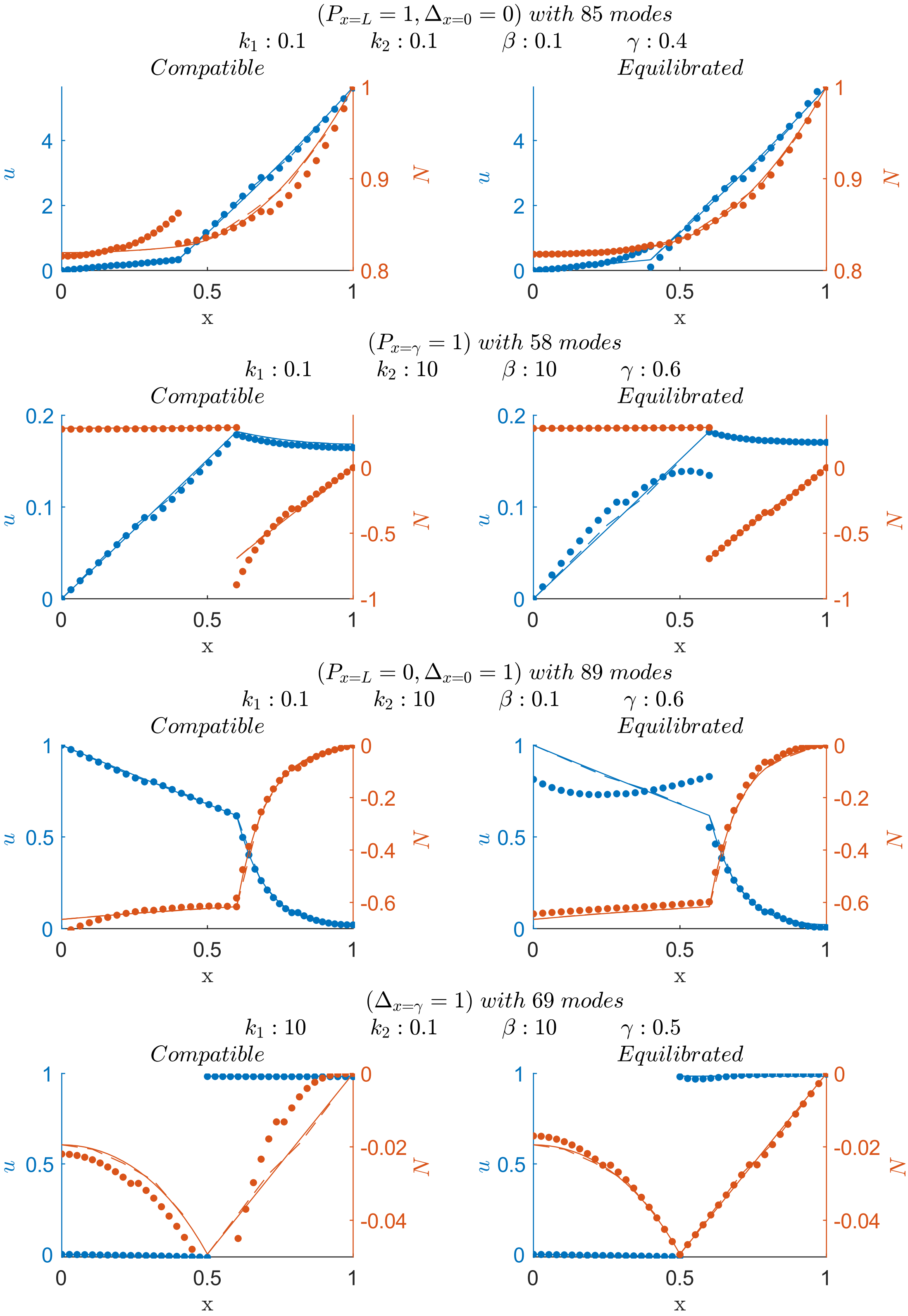


FIGURE 6.6: Displacements u (blue) and axial forces N (orange) for different load types, using variables values that lead to the worst approximations. The solid lines are the exact solution, the dashed lines the FEM approximations and the dots the PGD approximations. The polynomial degrees are 9 for the parameters space and 10 for the physical domain, with the physical domain being divided using two elements per section.

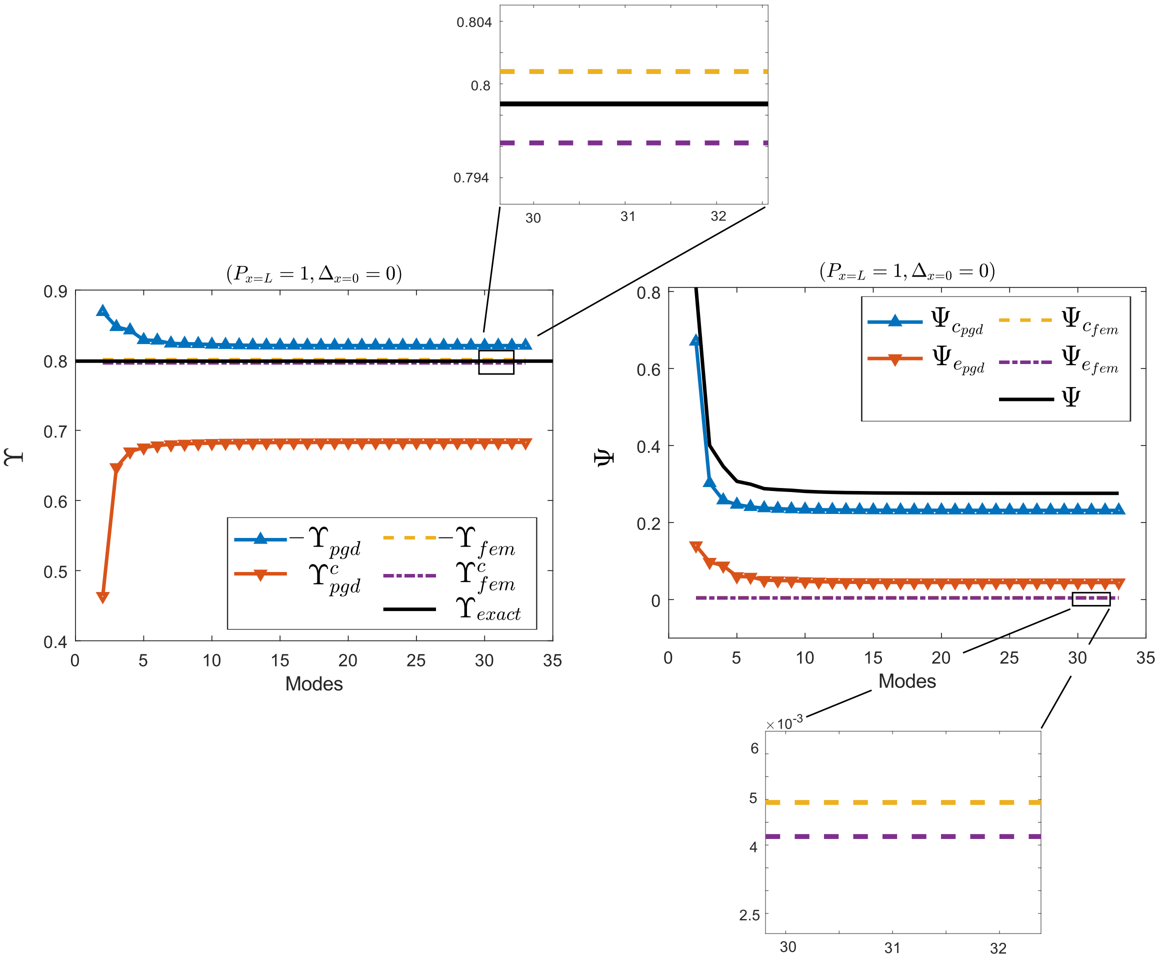


FIGURE 6.7: Integral of the energies for both PGD and FEM approximations, with Υ and Ψ being the integrals in the parametric domain of the total potential energies and the energy of the error, respectively.

behavior is expected for force or displacement driven problem, due to the orthogonality of the FEM solutions, but is not the case for mixed problems.

The results are now shown in terms of specific solutions instead of the integral. The solutions chosen to be used are the ones that present the best and worst results for each case. Figure 6.9 shows the energies for the solutions and for the errors, side by side. The bounds for the solutions are respected, although it is possible to see that the solution does not converge smoothly as the number of PGD modes increase. This is because the PGD modes were set to improve the integral of the solutions and not necessarily one specific solution, meaning that there may be modes that can lead to worse specific solution in same cases. Again, the worse solutions are located at the edges of the parameters domain.

Figure 6.10 shows the relative difference between the approximations obtained from the PGD model and the ones found using the FEM. Again, the results for the energy of the solutions and

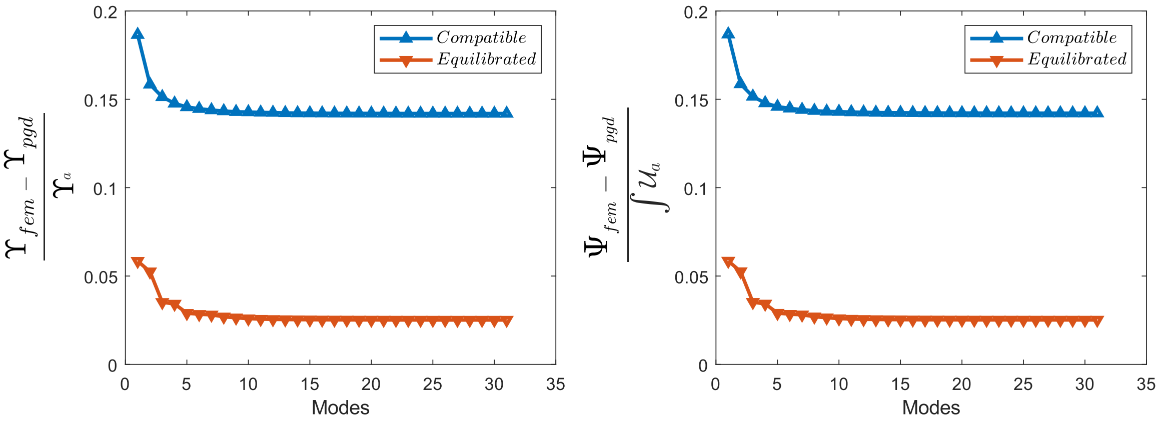


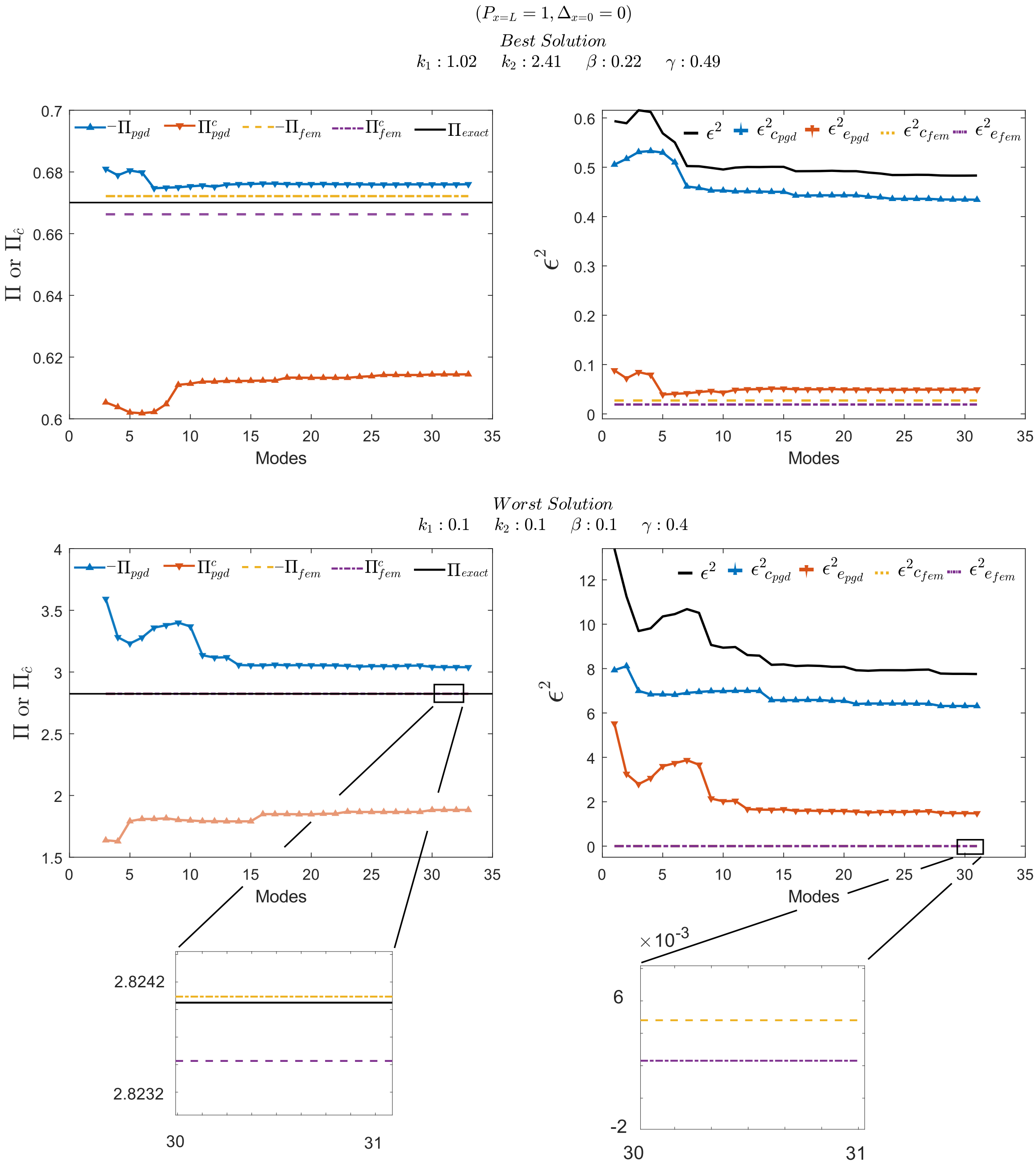
FIGURE 6.8: Relative difference between the PGD and FEM integral of the solutions energies and the average total absolute energy of the FEM solution (left). Relative difference between the PGD and FEM integral of the error energies and the average strain energy of the FEM error (right)

the error energy are the same, due to the force driven nature of the problem.

Figure 6.11 shows the behavior of the relative difference between the integral of the energies of the PGD and FEM approximations and the average potential or strain energies as the degree or the number of elements per section used to obtain the solutions increase. The solutions behave as expected, with better results being achieved as the degree or the number of elements increases.

Notice that for higher degree approximations, the solution starts to lose its convergence rate and also the initial solutions are worse than the ones obtained with smaller degrees. The decrease of the convergence rate is caused by a limit in the number of PGD modes due to the tolerance of the simulations. An indication that the solution is being constrained by the tolerance is that the number of PGD modes for a higher degree or number of elements may decrease. This can be seen for the solution with $p = 5$ and $h = 4$ and $h = 5$. Decreasing the tolerance can recover the rate of convergence, although this will not fix the poor results for the initial modes. We believe these poor results are a consequence of the complexity of the higher degree solutions. This is suggested by the fact that this behavior is not observed when increasing the number of elements of the solution. The results in terms of the error in energy and energy of the error are again the same and were omitted from Figure 6.11.

The error in energy is simpler to visualize, as it is based in the energy of the solutions which has a physical meaning. Also, each compatible and equilibrated solution has its own energy, allowing for individual analyses of the error. In the other hand, the bounds that are seen in the error in energy are limited to situations when we have force or displacement driven problems,



$(P_{x=L} = 1, \Delta_{x=0} = 0)$

Best Solution

$k_1 : 1.02 \quad k_2 : 2.41 \quad \beta : 0.22 \quad \gamma : 0.49$

Worst Solution

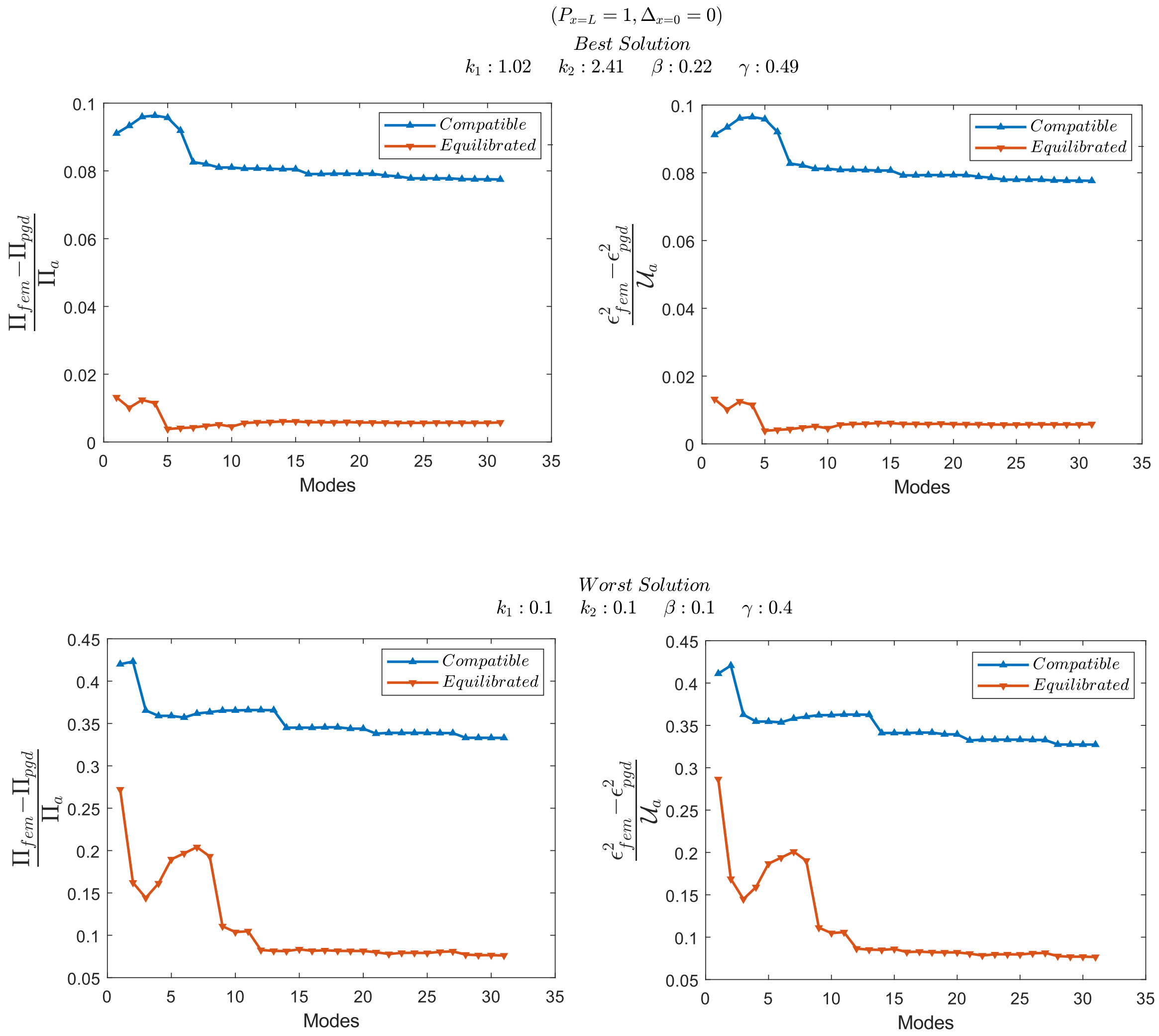
$k_1 : 0.1 \quad k_2 : 0.1 \quad \beta : 0.1 \quad \gamma : 0.4$

Figure 6.9: Best and worst energies values for both PGD and FEM approximations.

which is not the case for the energy of the error. That being said, from this point on we will base or solutions solely in the energy of the error, as it provides error bounds for general problems.

6.1.3 Solution Adaptivity

We want to obtain an adaptivity indicator that is capable of capturing the best regions to refine, without differentiating between the physical or the parametric space. To achieve this, we collect in variable χ both the parameters and the physical space. This means that, from this point on,



$(P_{x=L} = 1, \Delta_{x=0} = 0)$

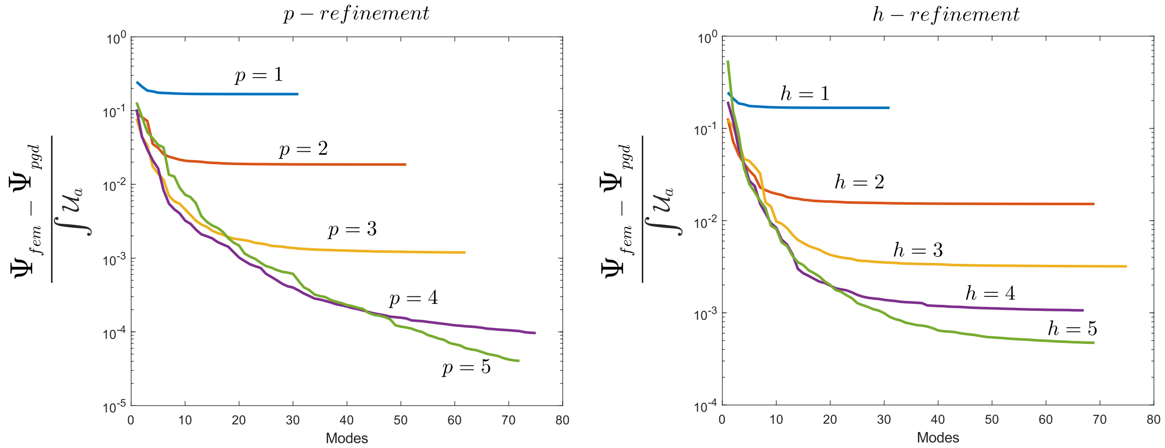
Best Solution

$k_1 : 1.02 \quad k_2 : 2.41 \quad \beta : 0.22 \quad \gamma : 0.49$

Worst Solution

$k_1 : 0.1 \quad k_2 : 0.1 \quad \beta : 0.1 \quad \gamma : 0.4$

FIGURE 6.10: Best and worst relative difference between the PGD and FEM solutions energies and the average total absolute energy of the FEM solution (left). Relative difference between the PGD and FEM error energies and the average strain energy of the FEM error (right)



$p - refinement$

$h - refinement$

FIGURE 6.11: Relative difference for the integral of the solutions energy and the integral of the error energy for the PGD approximation with different number of degrees and elements.

a reference to *the parameters* χ *of the problem* may be referring to a material parameter μ or to the parameter x of the physical space.

The solutions presented will be shown in terms of the number of elements n_h or the degrees of the polynomial approximations p. Our specific problem has a total number of parameters n_χ equal to five: four μ's and the physical space, which we divide in the two sections, b_1 and b_2.

A question that arises is how do we use the complementary solutions to decide where to refine the model. We know that the integral of the energy of the error covers the whole parameters and physical space domain and therefore it is always the same, no matter in which order we performs the integrals.

Recalling that the subscript k indicates solutions that come from k̲inematically admissible displacements and s that the solutions come from s̲tatically admissible axial forces, the parametric form for the error density, ρ, from Equation (3.2), considering the domain decomposition particularities, is:

$$\begin{aligned}\rho(\xi,\boldsymbol{\mu}) = J((N_k(\xi,\boldsymbol{\mu}) - N_s(\xi,\boldsymbol{\mu}))(\varepsilon_k(\xi,\boldsymbol{\mu}) - \varepsilon_s(\xi,\boldsymbol{\mu}))+ \\ (f_k(\xi,\boldsymbol{\mu}) - f_s(\xi,\boldsymbol{\mu}))(u_k(\xi,\boldsymbol{\mu}) - u_s(\xi,\boldsymbol{\mu}))).\end{aligned} \tag{6.8}$$

where f is the force transmitted by the support. We can look at the effects of each parameter independently by performing the integral of the error density ρ in all parameters but one. If this function is constant, it means that the parameter is not affecting the error bound, indicating that we need to look at the derivatives of the functions. This lead to the idea of working with the derivative of ρ with respect to all parameters. After numerical tests to assess how to relate the derivative of a parameter χ with the effect of that parameter on the error, the following expression for an error indicator ι was selected:

$$\iota(\chi) = \int\int (\chi - C(\chi)) \frac{\partial \rho(\xi,\boldsymbol{\mu})}{\partial \chi} \,\mathrm{d}\Omega\,\mathrm{d}\Omega_{\boldsymbol{\mu}}, \tag{6.9}$$

where:

$$C(\chi) = \frac{\int\int \chi\,\rho(\xi,\boldsymbol{\mu})\,\mathrm{d}\Omega\,\mathrm{d}\Omega_{\boldsymbol{\mu}}}{\int\int \rho(\xi,\boldsymbol{\mu})\,\mathrm{d}\Omega\,\mathrm{d}\Omega_{\boldsymbol{\mu}}}. \tag{6.10}$$

This expression was selected after testing several dimensionally consistent alternatives, including combinations of the derivatives of ρ with respect to the parameters. From this study, we concluded that (6.9) is the alternative that best captures the regions where refinement is required.

Note that this expression is not used to control if the process has converged, only where to refine next. We have not yet found a theoretical background that supports its application.

Considering ρ, which is non negative, as a pseudo mass density, we can interpret $\iota(\chi)$ as the first order moment of the derivative of the error with respect to the center of mass of the domain. The components in (6.8) can be approximated as:

$$\begin{aligned} u_k(\xi,\boldsymbol{\mu}) &= \boldsymbol{\phi}(\xi)\hat{\mathbf{u}}_k(\boldsymbol{\mu}); & N_s(\xi,\boldsymbol{\mu}) &= \boldsymbol{\phi}(\xi)\hat{\mathbf{N}}_s(\boldsymbol{\mu}); \\ F_k(\xi,\boldsymbol{\mu}) &= k\, u_k(\xi,\boldsymbol{\mu}); & \varepsilon_s(\xi,\boldsymbol{\mu}) &= \frac{1}{\beta}N_s(\xi,\boldsymbol{\mu}); \\ \varepsilon_k(\xi,\boldsymbol{\mu}) &= \frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\hat{\mathbf{u}}_k(\boldsymbol{\mu}); & F_s(\xi,\boldsymbol{\mu}) &= \frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\hat{\mathbf{N}}_s(\boldsymbol{\mu}); \\ N_k(\xi,\boldsymbol{\mu}) &= \beta\,\varepsilon_k(\xi,\boldsymbol{\mu}); & u_s(\xi,\boldsymbol{\mu}) &= \frac{1}{k}\,F_s(\xi,\boldsymbol{\mu}). \end{aligned} \tag{6.11}$$

These approximations can be expressed both in terms of FEM or PGD, only needing to set $\hat{\mathbf{u}}_k$ and $\hat{\mathbf{N}}_s$ accordingly. In order to compute (6.9) we also need to compute the derivative $\frac{\partial\rho(\xi,\boldsymbol{\mu})}{\partial\chi}$ and, therefore, we need the derivatives of (6.8). We will have different expressions for derivatives depending on what type of parameter χ is representing.

When χ represents the physical domain parameter x, and assuming that the material parameters k and β are defined such that $k=\mu_1$ and $\beta=1$ if $x\leq\mu_4$, and $k=\mu_2$ and $\beta=\mu_3$ if $x>\mu_4$ we have:

$$\begin{aligned} \frac{\partial u_k(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{J}\frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\hat{\mathbf{u}}_k(\boldsymbol{\xi}); & \frac{\partial N_s(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{J}\frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\hat{\mathbf{N}}_s(\boldsymbol{\mu}); \\ \frac{\partial F_k(\xi,\boldsymbol{\mu})}{\partial x} &= k\,\frac{1}{J}\frac{\partial u_k(\xi,\boldsymbol{\mu})}{\partial\xi}; & \frac{\partial \varepsilon_s(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{\beta}\,\frac{1}{J}\frac{\partial N_s(\xi,\boldsymbol{\mu})}{\partial\xi}; \\ \frac{\partial \varepsilon_k(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{J}\left(\frac{\mathrm{d}^2\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi^2}\right)\frac{\partial\xi}{\partial x}\hat{\mathbf{u}}_k(\boldsymbol{\mu}); & \frac{\partial F_s(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{J}\left(\frac{\mathrm{d}^2\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi^2}\right)\frac{\partial\xi}{\partial x}\hat{\mathbf{N}}_s(\boldsymbol{\mu}); \\ \frac{\partial N_k(\xi,\boldsymbol{\mu})}{\partial x} &= \beta\,\frac{1}{J}\frac{\partial \varepsilon_k(\xi,\boldsymbol{\mu})}{\partial\xi}; & \frac{\partial u_s(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{k}\,\frac{1}{J}\frac{\partial F_s(\xi,\boldsymbol{\mu})}{\partial x}. \end{aligned} \tag{6.12}$$

And when χ represents one of the parameters μ_i, with $i = 1, 2, ..., \mathcal{D}$, we have:

$$\begin{aligned}
\frac{\partial u_k(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \boldsymbol{\phi}(\xi)\frac{\partial \hat{\mathbf{u}}_k(\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial N_s(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \boldsymbol{\phi}(\xi)\frac{\partial \hat{\mathbf{N}}_s(\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial F_k(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \frac{\mathrm{d}k}{\mathrm{d}\mu_i}u_k(\xi,\boldsymbol{\mu}) + k\frac{\partial u_k(\xi,\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial \varepsilon_s(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \frac{\mathrm{d}\beta^{-1}}{\mathrm{d}\mu_i}N_s(\xi,\boldsymbol{\mu}) + \frac{1}{\beta}\frac{\partial N_s(\xi,\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial \varepsilon_k(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \left(\frac{\mathrm{d}}{\mathrm{d}\mu_i}\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)\frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\hat{\mathbf{u}}_k(\boldsymbol{\mu}) + \frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\partial \hat{\mathbf{u}}_k(\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial F_s(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \left(\frac{\mathrm{d}}{\mathrm{d}\mu_i}\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)\frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\hat{\mathbf{N}}_s(\boldsymbol{\mu}) + \frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\partial \hat{\mathbf{N}}_s(\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial N_k(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \frac{\mathrm{d}\beta}{\mathrm{d}\mu_i}\varepsilon_k(\xi,\boldsymbol{\mu}) + \beta\frac{\partial \varepsilon_k(\xi,\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial u_s(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \frac{\mathrm{d}k^{-1}}{\mathrm{d}\mu_i}F_s(\xi,\boldsymbol{\mu}) + \frac{1}{k}\frac{\partial F_s(\xi,\boldsymbol{\mu})}{\partial \mu_i}.
\end{aligned} \tag{6.13}$$

Notice that $\left(\frac{\mathrm{d}}{\mathrm{d}\mu_i}\frac{\mathrm{d}\xi}{\mathrm{d}x}\right) = 0$ unless μ_i is the geometric parameter $\mu_4 = \gamma$. For the derivatives of the PGD approximations, we have:

$$\begin{aligned}
\frac{\partial \hat{\mathbf{u}}_k(\boldsymbol{\mu})}{\partial \mu_i} &= \sum_{m=1}^{\mathcal{N}} \bar{\mathbf{u}}^m \frac{\mathrm{d}\,U_i^m(\mu_i)}{\mathrm{d}\mu_i} \prod_{j=1;j\neq i}^{\mathcal{D}} U_j^m(\mu_j)\\
\frac{\partial \hat{\mathbf{N}}_s(\boldsymbol{\mu})}{\partial \mu_i} &= \sum_{m=1}^{\mathcal{N}} \bar{\mathbf{N}}^m \frac{\mathrm{d}\,\tilde{N}_i^m(\mu_i)}{\mathrm{d}\mu_i} \prod_{j=1;j\neq i}^{\mathcal{D}} \tilde{N}_j^m(\mu_j)
\end{aligned} \tag{6.14}$$

We can now apply Equations (6.11) and (6.13) into Equation (6.9) to obtain the error indicator ι. Notice that by multiplying the derivative by the variable, we are able to keep ι with units that are consistent with the energy of the error ϵ^2.

6.1.3.1 p- Adaptivity

This section studies the results for p- refinement when using the proposed adaptivity indicator. It works with a simple verification, by identifying the parameter that has the highest value among all the parameters studied and increasing the polynomial degree of that parameter by one, as this parameter is expected to influence the solution error the most. The adaptivity process using the indicator proposed is compared with the uniform mesh adaptivity and is shown in Figure 6.12. Using the adaptivity indicator leads to better results than simply uniformly increasing the

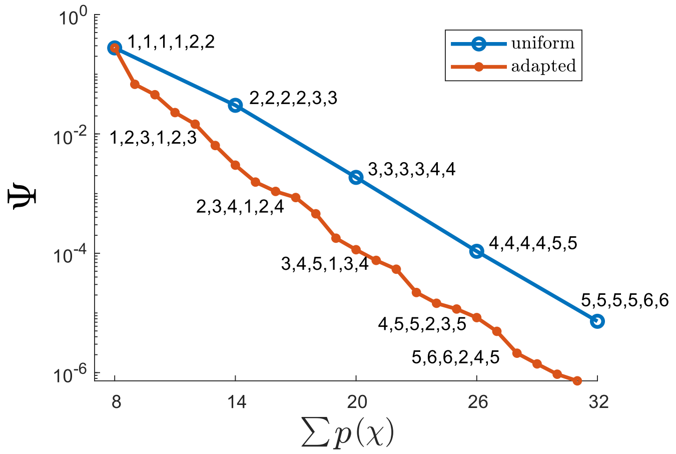


Figure 6.12: Comparison between the uniform mesh refinement and the adaptivity method for the integral of the energy of the error as a function of the combination of polynomial degrees. The detail numbers are the the degrees of the approximation in each parameter, respectively k_1, k_2, β_1, γ_1, b_1, b_2.

degrees of approximation functions, as it chooses the parameter that has the major influence in the solution to improve its degree. The drawback of this process is the need to repeat the simulation for every new degree improved.

One way to determine if the method to drive the adaptivity process is reliable is to determine all the possible solutions for a given set of degrees for the polynomial approximation and verify if the method is capable to accurately define which parameter should have its degree improved.

The following computation process was done:

1. Start with linear functions for the parameters and quadratic for the physical space;
2. Increase the polynomial degree of a parameter by one and compute the error;
3. Return the parameter to the original degree and repeat the previous step for the next parameter, until all variables are tested;
4. Pick the parameter that leads to the smallest error and permanently increase the polynomial degree of that parameter by one;
5. Go back to step two until the sum of the degrees of the parameter reaches a predetermined value.

The results for different tolerances (τ_{pgd}) were compared with a reference solution (τ_{ref}) which was obtained by testing all possible solutions and choosing the optimal result, using a tolerance of

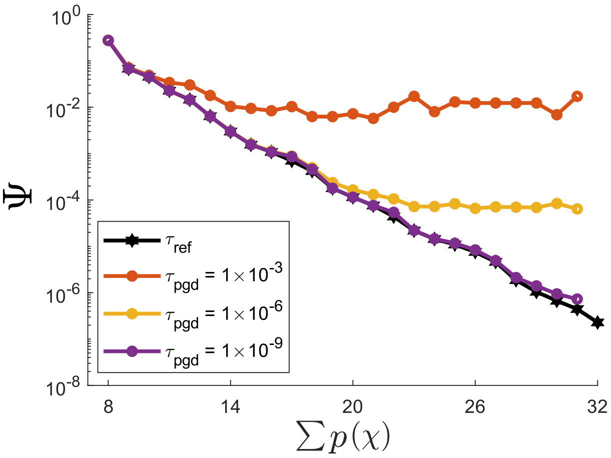


FIGURE 6.13: Comparison between the reference simulation τ_{ref} and the results with different tolerances for the integral of the energy of the error as a function of the combination of polynomial degrees.

1×10^{-12}. The results can be seen in Figure 6.13, where the error tends to stagnate after a certain tolerance is achieved. The adaptivity process is capable of precisely identify the parameter that causes the most impact in the solution, achieving better results for the error sooner than increasing all the degrees at once. We consider that the adaptivity method works if it is capable of selecting the parameter that required its polynomial degree increased to achieve the smallest error. Therefore, as long as the tolerance is small enough, the adaptivity method proposed is capable of choosing the optimal parameter to be refined.

6.1.3.2 h- Adaptivity

We now study the simulation results for h- refinement when using the proposed adaptivity indicator. Again, just like for the p- refinement, we use the parameter that gives the highest value ι as the indicator mechanism for the adaptivity process. One key difference for the h-refinement it that it is also necessary to know where the element should be divided. The most natural place to choose is the middle of the element, but we can instead use the center of gravity of the energy error for that parameter, which is already being computed for the adaptivity indicator. To confirm that the center o gravity of the parameter is the best place to break the element, a simulation with 50 different break points in the interval $]-1,1[$ of the variable β was performed and is presented in Figure 6.14. It is possible to observe that the point that leads to the real minimum value is really close to the center of gravity of the parameter. This behavior is seen in all others parameters, for different numbers of elements or degrees, leading us to believe that the center of gravity provides a good indication on where to divide the element.

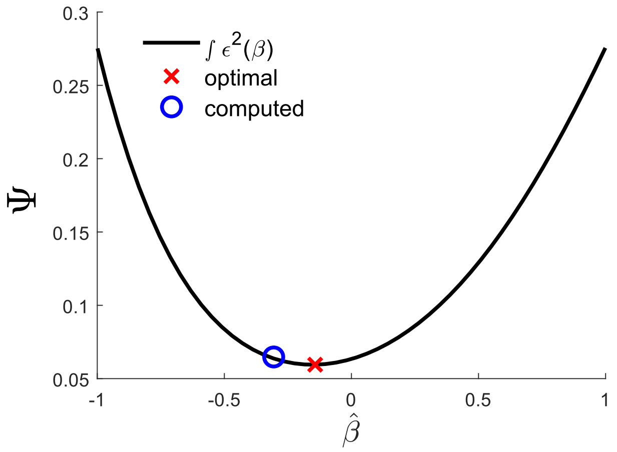


FIGURE 6.14: Energy of the error for variable β considering different element break positions. Details for the optimal break point and the computed using the gravity center of the energy of the error curve for β.

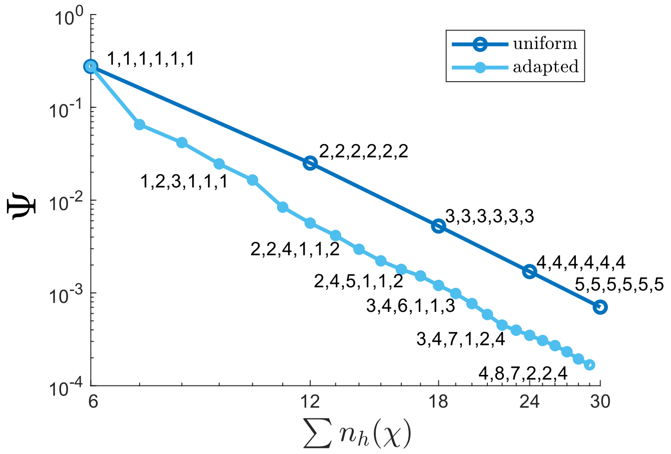


FIGURE 6.15: Comparison between the uniform mesh refinement and the adaptivity method for the integral of the energy of the error as a function of the element partitions. The detail numbers are the number of elements in each parameter, respectively k_1, k_2, β_1, γ_1, b_1, b_2.

The adaptivity process using the proposed indicator is compared with the uniform mesh adaptivity and is shown in Figure 6.15. Using the adaptivity indicator leads to better results than uniformly increasing the number of elements, as it chooses the variable that has the major influence in the solution to divide its elements. Just like for the p- refinement, the drawback of this process is the need to repeat the simulation for every new element division.

We repeat the verification process performed for the p- refinement, determining all possible solutions to the problem for a given sets of elements and verifying if the method is able to accurately define which parameter should have its element divided.

The following computation process was performed:

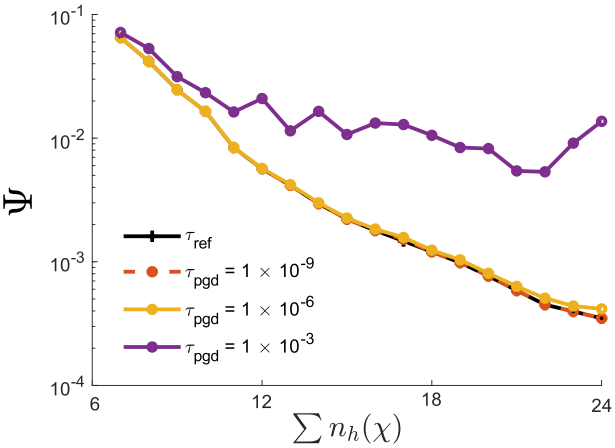


FIGURE 6.16: Comparison between the reference simulation τ_{ref} and the results with different tolerances for the integral of the energy of the error as a function of the element partitions.

1. Start with all parameter with single element;
2. Divide the element of the parameter and compute the total error;
3. Return the parameter to the original number of elements and repeat the previous step for the next parameter, until all the variables were studied;
4. Pick the parameter that results in the smallest error and permanent split the element of that parameter;
5. Go back to step two until the simulation reaches a predetermined value.

The results for different tolerances (τ_{pgd}) were compared with a reference solution (τ_{ref}) which was obtained by testing all possible solutions and choosing the optimal result, using a tolerance of 1×10^{-9}. These results are presented in Figure 6.16, with the total error stagnating after a certain tolerance is achieved. The adaptivity process is capable of precisely identify the parameter that causes the most impact in the solution, achieving better results for the total error sooner than increasing all the elements at once. We consider the adaptivity method good if it is capable of selecting the parameter that needed its element split to achieve the smallest error. Therefore, just like for the p- refinement, as long as the tolerance is small enough, the adaptivity method proposed is capable of choosing the optimal parameter to be refined.

6.2 2D Problem with Separable PGD Solutions

The problem we will work on is the one quarter section of a perforated plate, subjected to a constant load on its side, as presented in Figure 6.17. A characteristic of this geometry is the presence of geometric singularities that lead to stress concentrations. The approximate solution presented next are obtained using the `algebraicPGDtools` .

For the parametric problem, we are considering that the plate is composed of up to three horizontal layers S_i, where each layers or combination of layers has a different Young's modulus E_i, defined in the range $[0.1, 2.1]$. For the homogeneous plate the properties of the three layers are always the same ($E_1 = E_2 = E_3$). The problem with two materials is defined with S_1 and S_2 being combined to form a larger layer ($E_1 = E_2$). Finally, the case of the plate with three materials is defined with all three layers being independent.

The parametric range of the parameters is selected so that both errors, from the spatial discretization and from the PGD approximation, affect the solution. The PGD parametric domain is divided in 50 points for the simulations. Changing this value will affect the error associated with the space of the parameters. However, we observed that increasing the number of points does not change significantly the results in the examples presented. This number is problem dependent as the region with the most error influence can easily be concentrated in some random place. Usually, the more points we add in the parametric domain, the more accurate are the approximations obtained. Using a more clever distribution, such as using the Gauss points, also increases the accuracy of the results with fewer points needed.

The Poisson ratio could also be one of the parametric variables in our problem, but for simplicity we are fixing it at $\nu = 0.3$. Tolerances of 1×10^{-3}, 1×10^{-4} and 1×10^{-4} are used, respectively for the PGD model, the fixed point iteration and the adaptive process. We are setting the maximum number of iterations for the fixed point at 10. This value may be low for more complex problems, but is sufficient in our case. Additional errors in the approximation may be compensated by the addition of more PGD modes. The adaptive process will also stop if the new mesh yields more than 3000 elements.

The results obtained from the direct FE solution and from the PGD model were compared for different combinations of the values of the Young's modulus, E. This is illustrated, for the case of a plate composed of two materials, in Figure 6.18 for the amplitude of the displacements of the compatible solutions and in Figure 6.19 for the von Mises stresses of the equilibrated solutions. In

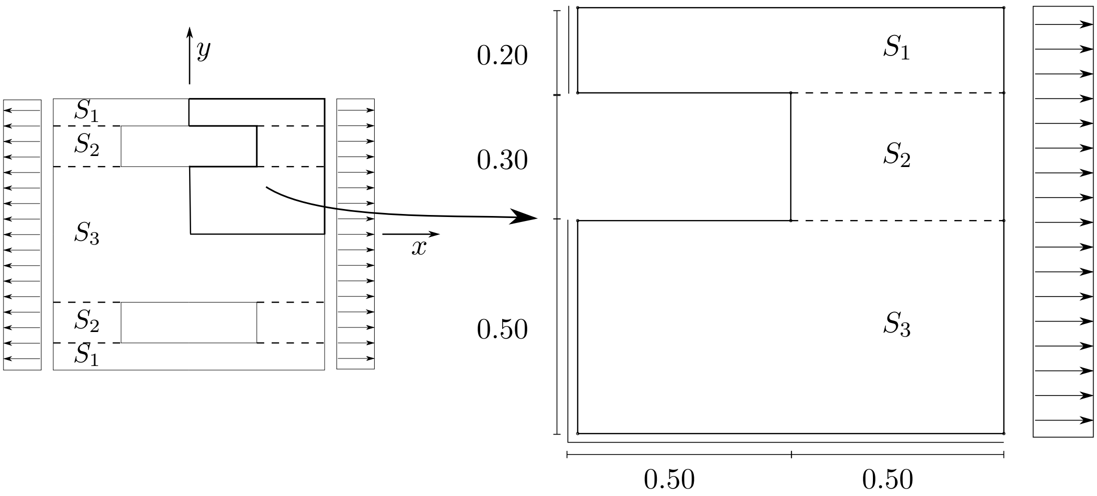


FIGURE 6.17: Perforated plate. Original geometry and loading, symmetry simplification, and definition of the layers.

this example, we are using an uniform mesh with 176 elements for the FEM and PGD problems, with the PGD approximation needing 21 modes to converge. We are considering a high degree for the polynomial approximation in space (6 for the equilibrated solution and 7 for the compatible), in order to reduce the error coming from the mesh and better visualize the results. The results presented show that the differences between the PGD and FEM approximations are relatively small.

The displacements in case (a) of an homogeneous plate in Figure 6.18 are, as expected, larger on the right hand side boundary, which is the location where load is being applied. Notice that, since the geometry of the problem is non-symmetric, the displacement values at the top right corner are larger than the ones at the bottom. Case (b) is obtained when considering that the bottom section of the plate is more flexible ($E = 0.1$), while the top section is stiffer ($E = 2.1$). This case has a unique characterization of the displacements, with larger values on the bottom right part of the plate, but still fairly large displacements at the top right part. This is mainly due to the bending effect, which induces vertical displacements of the top right part. Case (c) is the opposite of (b), with the top section being more flexible ($E = 0.1$) and the bottom section being stiffer ($E = 2.1$). Here, as the top edge of the plate is free, we have a more consistent behaviour for the displacements, with the top section having high displacements values at the right edge and the bottom section remaining still.

The results in Figure 6.19 are based on the same conditions. For the homogeneous case (a), we have regions of stress concentration only at the re entrant corners. In cases (b) and (c) we have

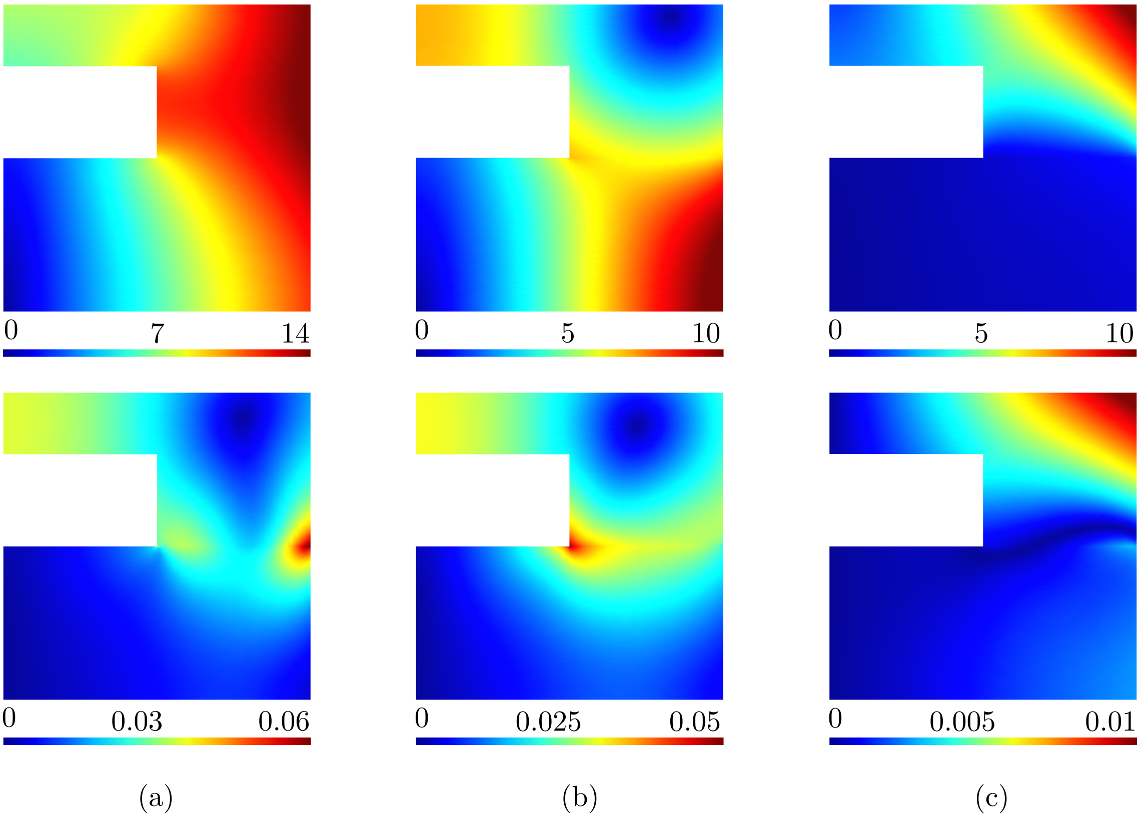
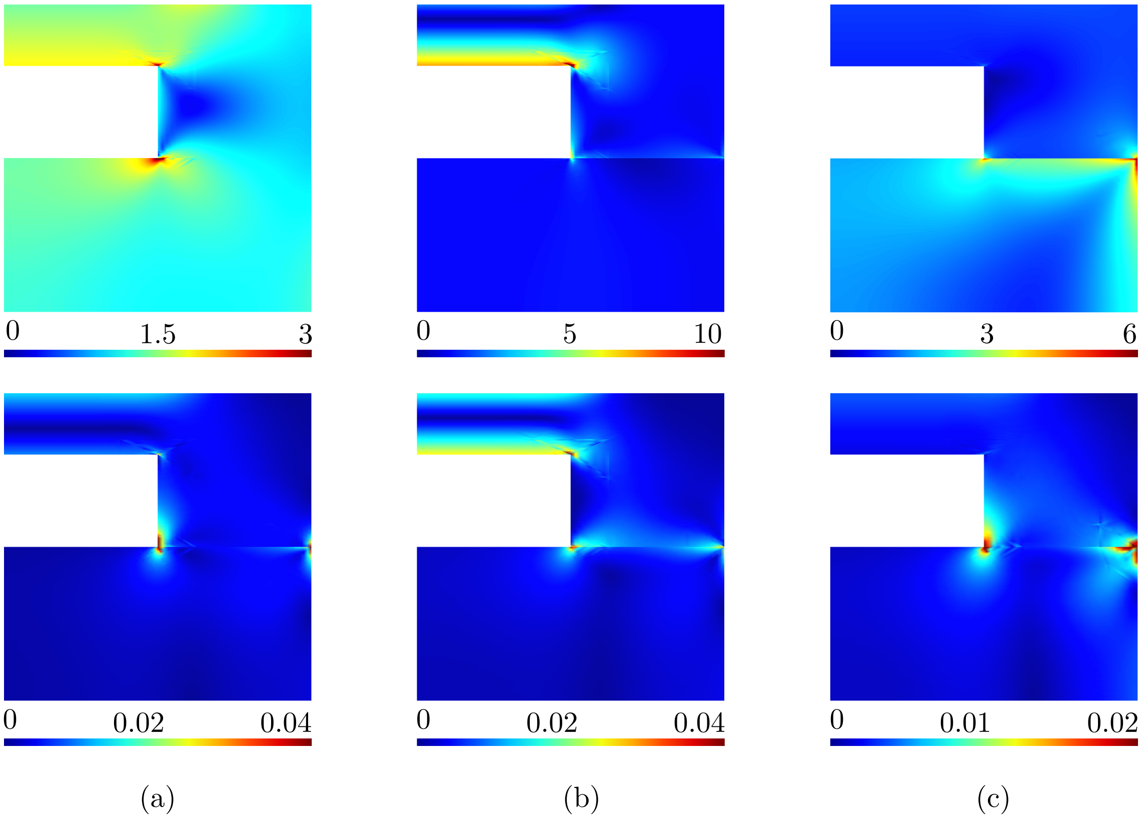


Figure 6.18: Illustration of the displacements of the compatible solutions for the case of (a) an homogeneous plate, (b) a plate with a flexible bottom section, and (c) a plate with flexible top section. The top row presents amplitude of the displacements of the PGD solutions, and the bottom row the amplitude of the difference between the PGD and the FEM solutions. We obtain both compatible solutions in the same mesh with 172 elements using polynomial approximation of degree 7, with the PGD converging after 21 modes.

an additional singularity, faintly captured by this mesh, at the right of the transition between the two materials, which is associated with the jump in the mechanical properties.

Notice that, as this is a force driven problem, the equilibrated approximations are usually more accurate than the ones obtained from the compatible approximations. This difference in the quality of the approximations is more evident when considering lower degree approximations in space or meshes with fewer elements, but these type of solutions are out of the scope of this study and are omitted here.

For other simulations, considering different values for the parameters in the PGD, we obtained compatible and equilibrated solutions similar to the ones obtained with the corresponding FE model, as long as the PGD solution has a sufficiently large number of modes. This was observed even when considering coarse meshes and low approximation degrees, in which case both solutions are of poor quality.



(a) (b) (c)

Figure 6.19: Illustration of the stresses of the equilibrated solutions for the case of (a) an homogeneous plate, (b) a plate with a flexible bottom section, and (c) a plate with a more flexible top section. The top row presents the von Mises stresses of the PGD solutions, and the bottom row the von Mises stresses of the difference between the PGD and the FEM solutions. We obtain both equilibrated solutions in the same mesh with 172 elements using polynomial approximation of degree 6, with the PGD converging after 21 modes.

After verifying that the problem behaves as expected in terms of stresses and displacements and that the solutions obtained are in agreement with the ones computed from the FEM model, we analysed our solutions in terms of the bound of their error. Figure 6.20 shows the error Ψ as a function of the number of elements n_e, for different degrees of the space approximation n_d. The problem we are presenting here is again the one considering that all the sections are composed of the same material, with the meshes being uniformly refined and starting from a mesh with number of elements $n_e = 24$.

We can verify that the problem being studied is well behaved because as we uniformly increase the number of elements in the mesh, the bound of the errors tends to reduce in a uniform and predicable manner. As expected, all the curves tend to a linear behaviour when graphed in a logarithmic scale. The slope of the lines does not increase when the degree of the approximation is increased because in this problem convergence is always limited by the singularities.

The bound of the energy of the error is computed from the sum of the contributions of each

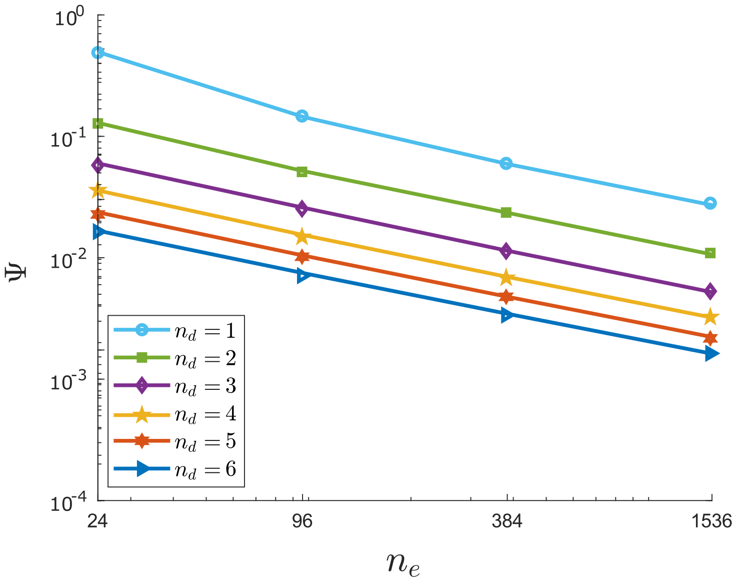


FIGURE 6.20: The global error Ψ as a function of the number of elements, for different degrees of the polynomial approximation in space d. There is no degradation in the convergence obtained for the PGD solutions.

element in the finite element mesh, which can be used to drive an adaptivity scheme. Figure 6.21 shows the results of performing a mesh refinement on three different situation: the first one considers the case where all three layers S_i are composed of the same material; the second considers two layers; the last case considers three layers. The mesh refinement is performed following the criteria defined in Section 5, limiting the maximum number of elements in the meshes to be lower than 3000.

As we increase the number of different materials in the plate, we obtain different meshes. This is a direct consequence of the error distribution in the plate shifting from one with stress concentrations only around the geometry singularities, to another that also depends on the difference between the materials. It is also worth mentioning that, although it is not possible to make a straight comparison between the different problems due to the difference in the number of elements in each mesh, it is noticeable that as we increase the complexity of the problem, the global error Ψ has a tendency to increase, what can be seen from the different value of Ψ for (a) and (c), in Figure 6.21.

For the case where the plate is composed of two materials, Figure 6.22 shows the detail of the mesh at the stress concentrations points, where the majority of the elements are concentrated. Since the strength of the mechanical singularity is related to the ratio between the moduli of the adjacent materials, the corresponding singularity is visible because the maximum possible ratio

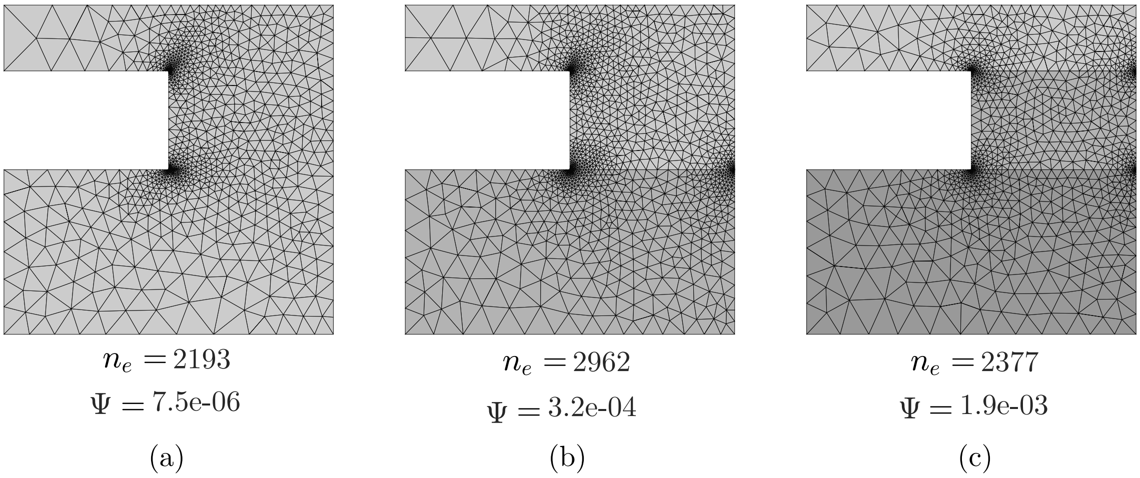


FIGURE 6.21: The final meshes obtained from the adaptivity process, the respective number of elements n_e and the global error Ψ, for the cases of a plate with (a) one, (b) two, and (c) three different materials.

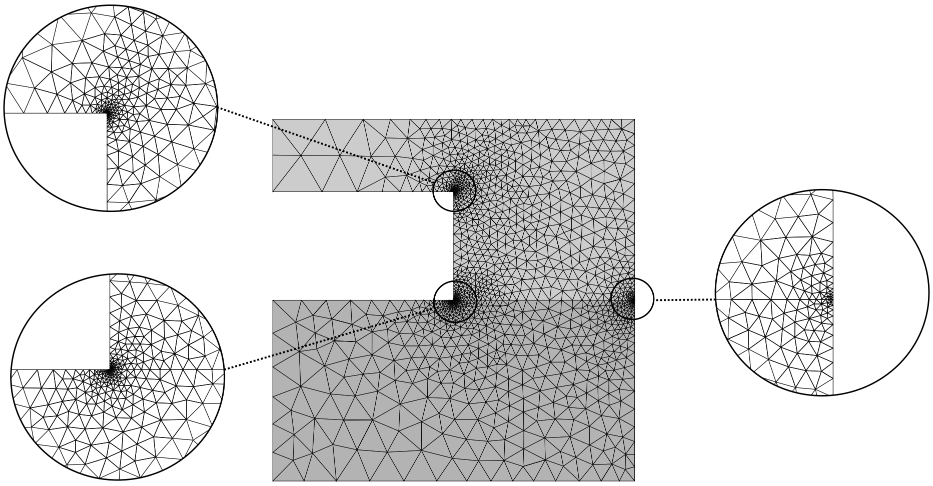


FIGURE 6.22: The final mesh obtained from the adaptivity process for the case of a plate with two different materials, with details at the stress concentration points.

between the materials defined in the PGD intervals is sufficiently large. A smaller ratio would lead to smaller errors and, as a consequence, to fewer elements on that area.

Our next analysis is assessing the benefit of using an adaptivity scheme for the mesh refinement, instead of simply performing a uniform refinement. Although a uniform mesh refinement does not require any additional effort to define a new mesh, a properly adapted mesh should be capable of achieving the same accuracy with fewer elements.

Figure 6.23 shows the results for the Ψ error for both types of refinements, with details for some of the meshes obtained during the process. The uniform meshes are obtained by simply splitting each original elements in four new ones, while the adapted mesh focuses the elements in the regions with higher errors. Again, we are limiting our meshes to a maximum of 3000 elements.

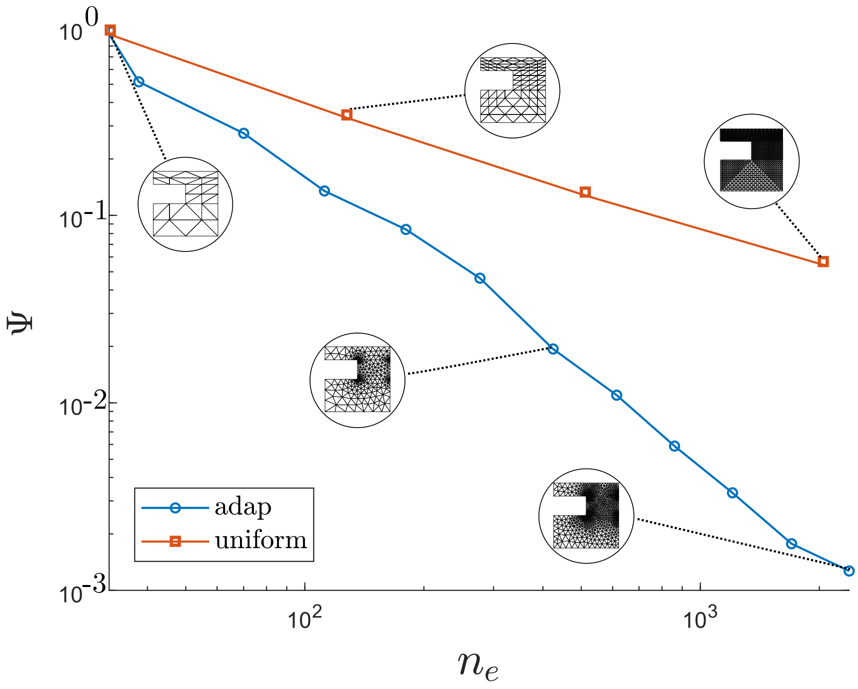


Figure 6.23: The global error Ψ as a function of the number of elements n_e in the mesh. We represent the cases of meshes obtained with an uniform and an adapted refinement scheme. The details in the figure are the meshes used at the selected points. Polynomial approximations of degree 1 and 2 are used for the compatible and equilibrated solutions, respectively.

Using the adaptivity process driven by the bound of the energy of the error leads to results with an improvement of approximately two orders of magnitude, for the same number of elements in the mesh. This is clearly due to the smart choice of using smaller elements in the regions where the the solution changes more rapidly, as can be seen in the detail of the finer adapted mesh.

For the adapted solutions the values of the error practically follow a linear trend in a logarithmic scale, where the last adapted mesh in the simulation breaks this trend, as the PGD enters the stagnation zone. Using a lower tolerance for the PGD approximations we obtain more solutions that follow the trend line.

Finally, we present in Figure 6.24 the behavior of the energy of the error Ψ as we increase the number of modes in the PGD approximations for some of the adapted meshes. This plot shows that there are two sources of errors that need to be balanced, those due to the spatial discretization of the domain and those arising from the truncation in the PGD expansion. In the case of coarse meshes, the discretization error dominates, and by increasing the number of modes, one does not diminish the total error, which quickly reaches a plateau. On the other hand, the truncation error dominates in the case of finer meshes and more modes are necessary to reach the plateau in that case. It is worth noting that, up to a given number of modes, the total error is more or less the same, independently of the mesh. This poses the question of having the

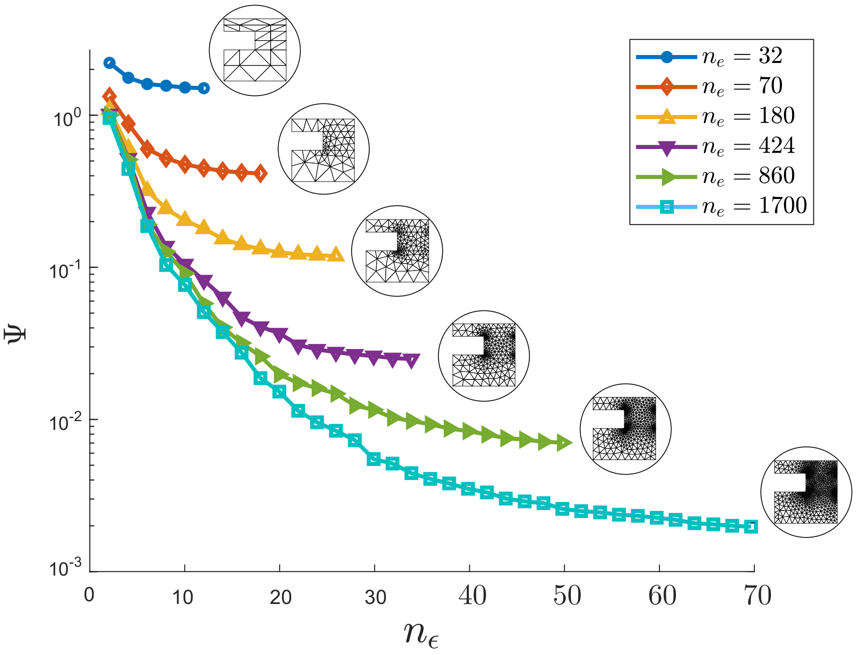


FIGURE 6.24: The global error Ψ as a function of the number of modes n_ϵ in the PGD. The curves are plotted for some of the different meshes obtained during the adaptive refinement process. The details in the figure are the meshes used, with the number of element n_e of each mesh in the legend.

first modes computed on coarse meshes and subsequent modes calculated on finer ones. When a direct projection between different meshes is possible, as in the compatible solutions, this can lead to a significant reduction of the computational cost, as the PGD process is not restarted form scratch after each mesh refinement. However, the projection of the stresses between different meshes requires the imposition of equilibrium in the interfaces of the new elements. Additional studies must be performed to assess the efficiency of the alternative approaches.

6.3 Quantities of Interest and Local Output Error Bounds

The examples presented in this section intend to demonstrate how the procedures that were described can efficiently provide guaranteed bounds of local outputs. We also assess the influence of using an error indicator associated with a local quantity to drive an adaptive refinement process. The approximate solution presented in this section are obtained using the `algebraicPGDtools` .

We use the simple academic problem described in Figure 6.25. It corresponds to the symmetry simplification of a square plate subjected to an unit horizontal traction. The plate is composed of two layers S, each made of materials with different Young's modulus E defined in a range $[0.1, 2.1]$. The Poisson ratio ν for both sections is fixed at 0.15.

Simulations with a more complex geometry were performed, but their results are not significantly different from those obtained for the simpler problem. Hence, we decided to present only the results from the square plate problem because it is easier to focus on the aspects that affect the goal-oriented refinement and the PGD approximations, such as the change of signs in the results, depending on the set of parameters chosen for the Young's moduli. Several applications using a more complex geometry, in the context of adaptivity driven by the global error, can be found in the previous section.

Although the stopping criteria used by the `algebraicPGDtools` code are a reliable way to determine when the PGD solution has converged, since we are independently computing pairs of solution, it may lead to more modes than necessary for our application [7, 21]. Hence, we use the information we have of the energy of the error as our PGD stopping criterion, assessing the convergence of the PGD approximations, considering the pair of solutions. The criterion is obtained by computing, for each additional mode of the PGD, the value of Equation (4.24) and divide it by the energy of the system. We then compare the relative difference with the previous solutions and stop the process when the difference reaches a certain threshold tolerance τ. Note that this means that both solutions of the global problem will have the same number of PGD modes. Furthermore, this criterion is applicable for either the real or the virtual problem.

This verification, which is performed for every additional PGD mode, becomes more costly as the number of modes increases, remaining a small fraction of the total running time. It is worth mentioning that computing the equilibrated solutions is usually more costly than computing the compatible approximations, by a factor close to two, for both the pre-processing and the solution of the PGD. This is highly dependent on the size of the finite element mesh, increasing for larger meshes. Naturally, these are just rough estimates, as the codes used to obtain these solutions are not aimed for efficiency.

To stop the adaptivity refinement process we consider either if the ratio between the error for the current mesh and the energy of the problem is bellow a certain threshold or if a given number of elements in the mesh is exceeded. A more detailed explanation on the stopping criteria used can be found in 5.

The parametric domain, unless otherwise stated, has $n_h = 50$ points. The tolerance for the PGD model, the fixed point iteration and the adaptivity process are all set to 1×10^{-4}. As we will demonstrate at the end of this section, this value guarantees that the error from the PGD approximations is negligible. The maximum number of fixed points iterations is 15, which may be low for more complex problems but, as we are working with an smooth problem, gives good results without slowing the computations. If during the mesh refinement process the tolerance is not reached, the adaptivity process stops when the mesh exceed $n_e = 6000$ elements. This number of elements is chosen due to computational constraints. The degree of the polynomial approximations in space are always linear for the equilibrated solutions and quadratic for the compatible ones, so that the stress fields of both solutions are described by linear functions.

We consider two different QoIs:

- The self compliant problem associated with the horizontal load, where the QoI is the integral of the horizontal displacement at the right side of the plate. For this QoI the real and virtual load are the same, being represented by the full arrows in Figure 6.25;
- The integral of the vertical displacement on a line at top boundary of the plate, due to the horizontal load. Here, the real load corresponds to the full arrows, and the virtual loads to the hollow arrows in Figure 6.25.

Figure 6.25 also presents the displacement fields obtained for the global problem when Young's moduli at the opposite edges of the parametric domain are considered, for a reference case with polynomial degrees of approximation of the space equal to 6 and 7 for equilibrated and compatible solution, respectively, $n_e = 320$ elements, $n_h = 50$ and a number of modes equal to 50 for the real and virtual solutions. The number of modes used for this specific simulation is arbitrarily high, so that the errors shown are based only on the mesh. All the other simulations have a variable number of modes, which are controlled by the selected tolerance.

As shown in the detail of the figure, this problem is particularly interesting because the output changes sign and, consequently, it is zero for certain values of the parameters. This is important because, when the outputs are obtained from a poor solutions, the information obtained may be grossly misleading. The only way to be certain of the quality of the output is by looking at its bounds. Also note that, because the output is zero for certain combinations of the parameters, it is not reasonable to work with the relative error.

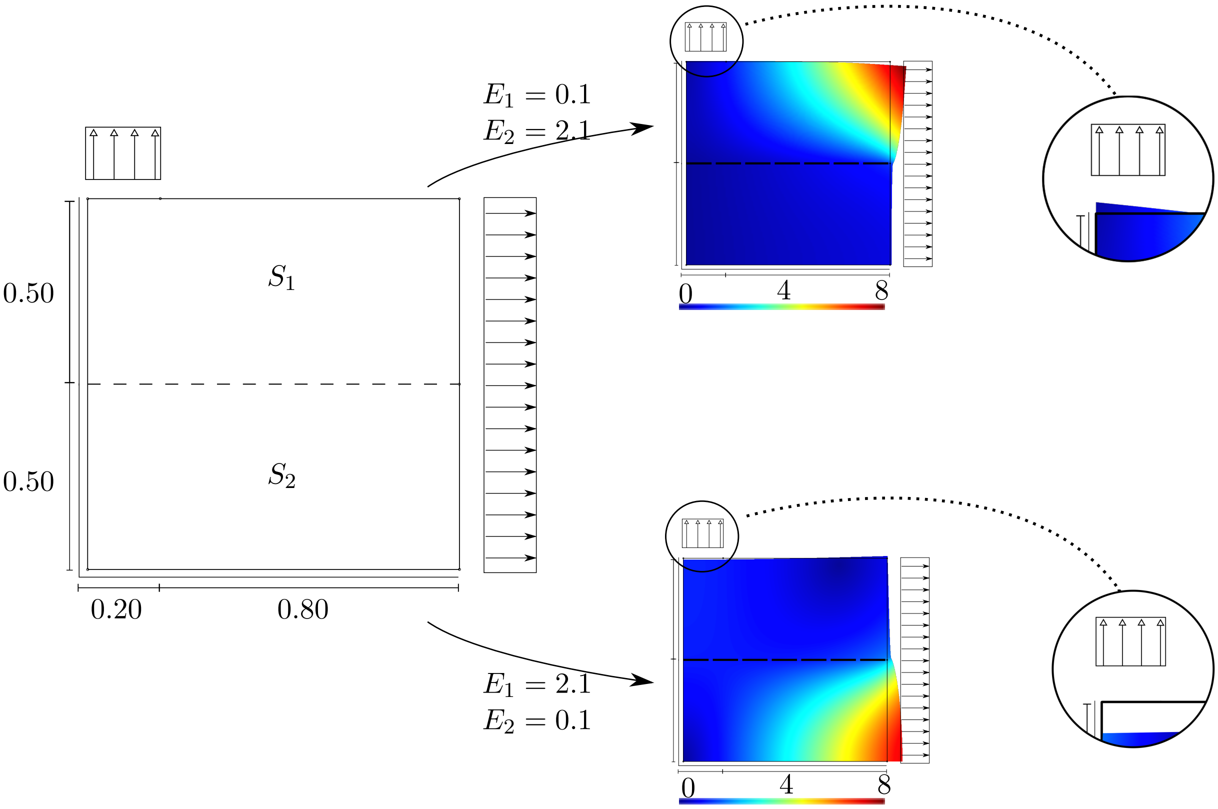


FIGURE 6.25: Symmetric simplification of a square plate, with the representations of the real (solid arrows) and virtual (hollow arrows) loads for the second QoI. The displacement fields for the case with the Young's moduli at the opposite edges of the parametric domain are presented, with details for the region where the displacement changes signs. The displacements at the detail plots are exaggerated for a better visualisation.

6.3.1 The Self-Compliant Problem

We begin with a self-compliant problem, comparing the results obtained by applying either a global or a QoI driven adaptivity technique. The apparent equivalence between both approaches is discussed, and the results are used as a validation of the codes used.

When using the same load for the real and for the virtual problem the energies are the same, so that the local output bounds are proportional to the global energy. This can be demonstrated by using the same value for the energy of the error in Equations (4.28) or (4.30) for the real and virtual problems.

Figure 6.26 presents the values obtained for the integrals of the global (left) and local output (right) errors using refinements driven by the global or local output error indicators. As expected, the values obtained are practically identical. Note that the slope for Ψ_{global^2} is almost twice that for Ψ_{local^2}, which is expected, because the local indicator is based on a product of two energies.

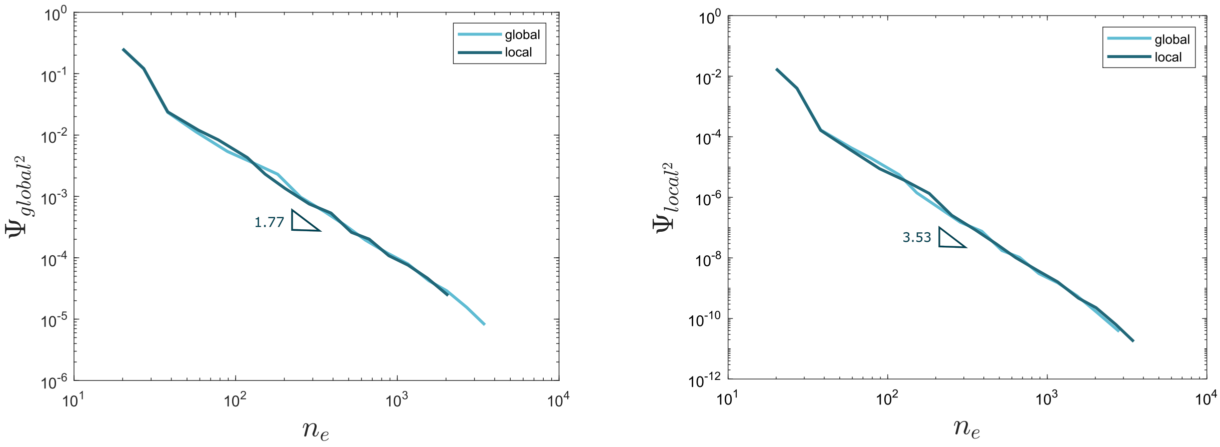


FIGURE 6.26: Self-compliant problem. The integrals of the global (left) and local output (right) errors as a function of the number of elements in the mesh. We present the cases of errors for the sequence of meshes obtained with an adaptive refinement based on the local output error indicator and a refinement driven by the global error indicator.

The difference in the results for either integrals come from the fact that the meshes used by the refinements driven by the local output and global indicators are different. This is because, although it may seem that the integrals Ψ_{global^2} and Ψ_{local^2} have only a difference in scale, in reality one cannot immediately be obtained from the other, as the integral of the square of a function is not the square of its integral.

6.3.2 The Displacement at the Top as the QoI

For this problem we present in Figure 6.27 the meshes obtained by using a uniform mesh refinement and by driving the adaptivity process with the integrals of the global or local output error indicators. We observe that:

- The uniform mesh refinement process has no information regarding the quality of the solution and, consequently, the final mesh obtained has the highest error and no particular distribution of the elements;
- The mesh obtained by using the global error indicator has information of the error of the solution of the real problem and captures the transition of mechanical properties, with an error that is much lower than that obtained with uniform refinement;
- The mesh resulting from using the local output error indicator is capable of capturing both the mechanical singularity and the effect of the virtual load in the solution of the problem, leading to the lowest values of the local output error.

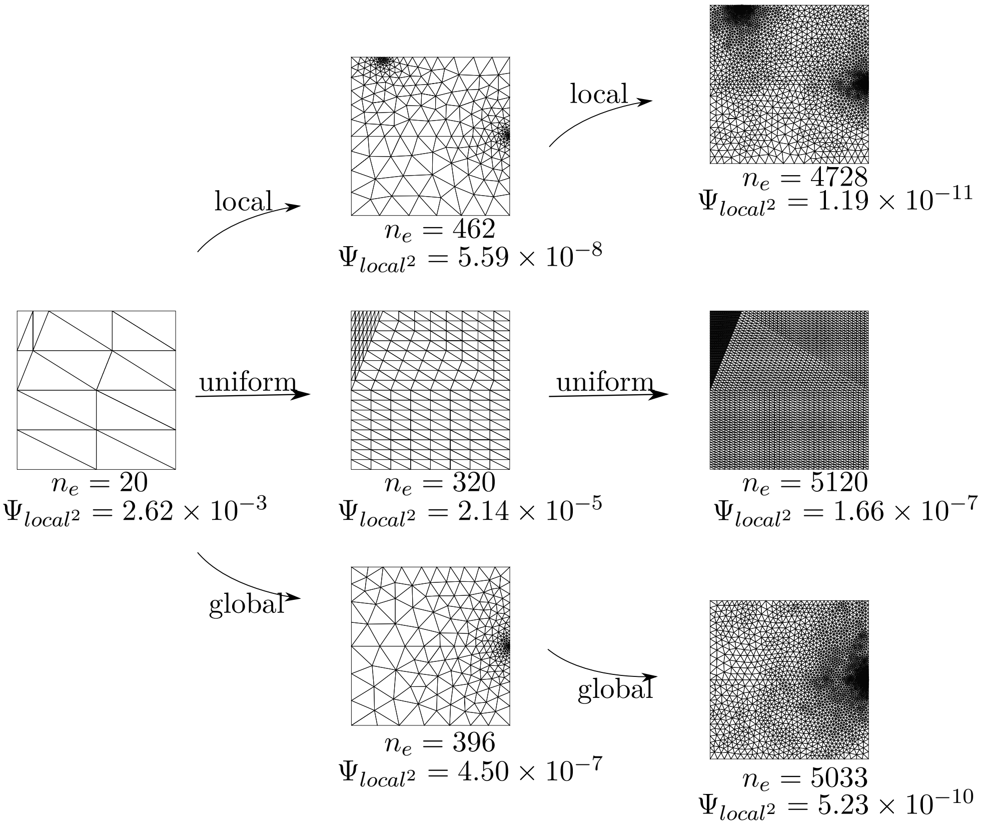

local
local
uniform
uniform
global
global
$n_e = 20$
$\Psi_{local^2} = 2.62 \times 10^{-3}$
$n_e = 462$
$\Psi_{local^2} = 5.59 \times 10^{-8}$
$n_e = 320$
$\Psi_{local^2} = 2.14 \times 10^{-5}$
$n_e = 396$
$\Psi_{local^2} = 4.50 \times 10^{-7}$
$n_e = 4728$
$\Psi_{local^2} = 1.19 \times 10^{-11}$
$n_e = 5120$
$\Psi_{local^2} = 1.66 \times 10^{-7}$
$n_e = 5033$
$\Psi_{local^2} = 5.23 \times 10^{-10}$

FIGURE 6.27: The meshes obtained during the adaptive refinement process using the displacement at the top as the QoI. We consider the cases of a refinement driven by the local output error indicator (top), an uniform mesh refinement (middle) and an adaptive refinement process driven by the global error indicator (bottom).

Notice that the number of elements n_e for the final meshes is not the same and hence the error values obtained cannot be compared directly, only in a qualitative manner.

We summarize in Figure 6.28 the values obtained for the integrals of the global and local output errors using the different types of refinement. As expected, using either the global or the local output error indicators is always better than uniformly refining the mesh, since having information about the error results in more elements being used where they lead to lower error values. Also, using the local output error indicator leads to better results for the local output error while using the global error indicator yields better results in the global error values.

This effect is well defined in a sufficiently smooth problem, such as the one that we are studying. However, this may not be the case for other problems, particularly when other error sources influence the solutions, for example geometrical singularities. It was observed that when working with local outputs for the example in [32], the meshes obtained based on the global or local output adaptivity indicators are very similar.

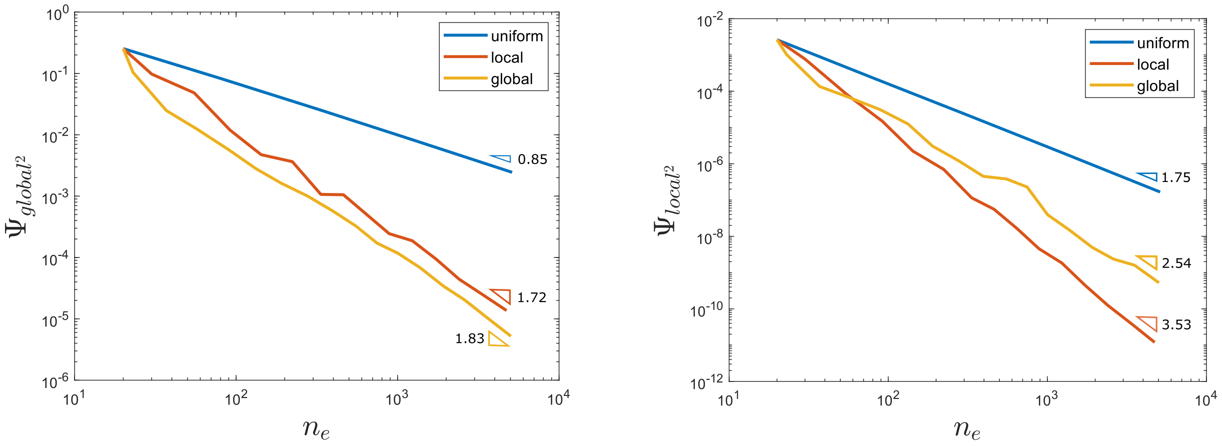


FIGURE 6.28: The integrals of the global (left) and local output (right) errors as a function of the number of elements in the mesh, using the displacement at the top as the QoI. We present the cases of errors for the sequence of meshes obtained with an uniform refinement, an adaptive refinement based on the local output error indicator and a refinement driven by the global error indicator.

We present in Figure 6.29 the local output errors obtained from the different refinements when considering the different combinations for the values of the Young's modulus E. The meshes used are the most refined ones presented in Figure 6.27. Using the uniform refinement yields the worse results, even though its mesh has the highest number of elements, while driving the refinement with the local output error indicator provides the best results, even with fewer elements. The edges of the parametric domain have the worse approximations, particularly when one of the sections has a lower Young's modulus, because that results in higher relative displacements between section.

The lowest values for the error are in the region where the materials of the two sections are the same. In this region, the exact stresses and strains are uniform and, hence, can be represented by the finite element solution so that the error should be zero. This is not the case because the PGD introduces truncation errors, still the error in this regions is lower.

Figure 6.30 presents the output and its bounds for an initial mesh (top), an uniformly refined mesh (left) and a mesh obtained by using the local output error indicator to drive the adaptivity process (right), always using the same range in similar plots. The figure also presents the output for the refined meshes for the case when the Young's modulus of the second section is $E_2 = 1$, detailing the region when both moduli are equal.

Again, the edges of the parametric domain have a much higher influence in the output obtained, with the bounds in the initial mesh being much higher when one the sections has a soft material. Using the goal-oriented refinement lead to a larger improvement in the quality of the bounds

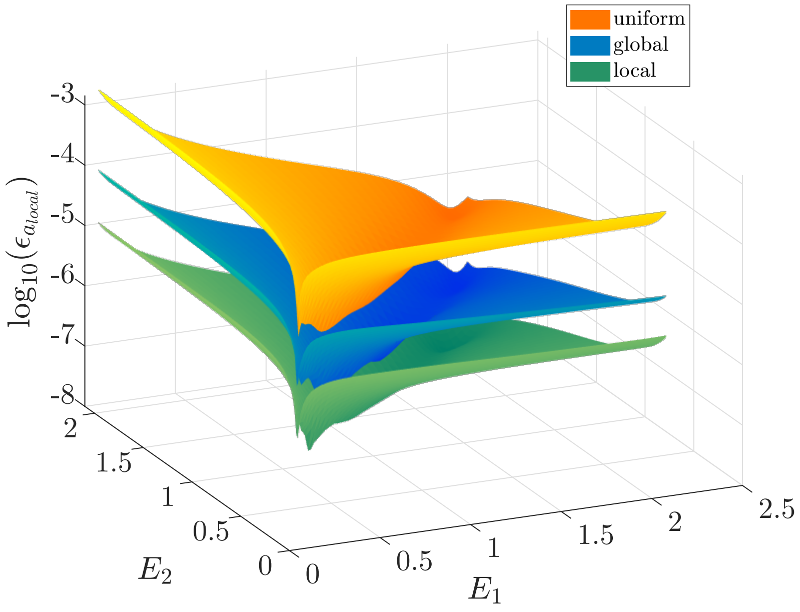


FIGURE 6.29: Error bounds of the displacement at the top for all the combinations of parameters values. We present the cases where the meshes obtained by an uniform refinement, a refinement based on the global error indicator and a refinement driven by the local output error indicator. The meshes used are the most refined ones presented in Figure 6.27

than the uniform refinement. This behaviour is better visualized in the plot of the cross section, where we see in the detail that, for the case where both the materials are equal, the uniform refinement has larger errors than the refinement based on the local output error adaptivity indicator. The number of elements n_e in both meshes is different ($n_e = 80$ for the uniform and $n_e = 92$ for the adapted), but the qualitative analysis still holds.

Notice that the sign of the output for the lowest values of E_1 is still uncertain in the bounds provided by both meshes, indicating that further refinement is necessary.

Finally, we combine in Figure 6.31 the local output obtained from fine solutions, with a high number of elements and points in the parametric domain ($n_e = 1280$, $n_h = 100$), which can be considered a reference solution, with the bounds of the local output obtained from a coarse solution ($n_e = 20$, $n_h = 5$).

It is observed that at the points of parametric discretization, some of which are detailed in the figure, the bounds are always effective. However, at points of the parametric domain that were not discretized, this is not always the case.

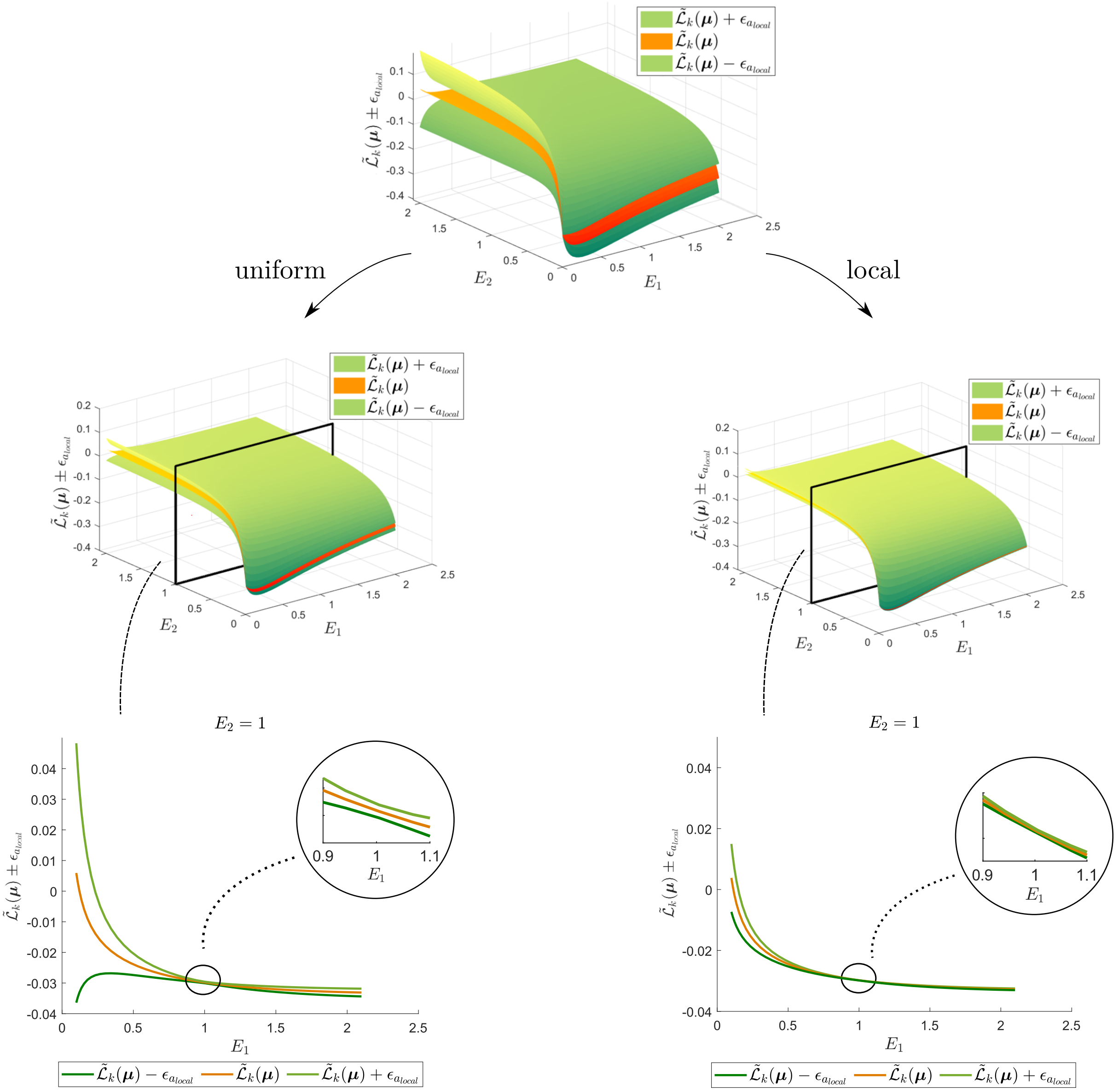


FIGURE 6.30: Displacements outputs and their respective bounds for all the possible parameters values. We present the case for the outputs and bounds obtained with an initial mesh (top) and the cases with a uniformly refinement mesh (left) and a mesh refined using the local output error indicator (right). The details in the uniformly and locally refined meshes present the values of the output and bounds when the Young's modulus $E_2 = 1$ and the regions where the two moduli are equal.

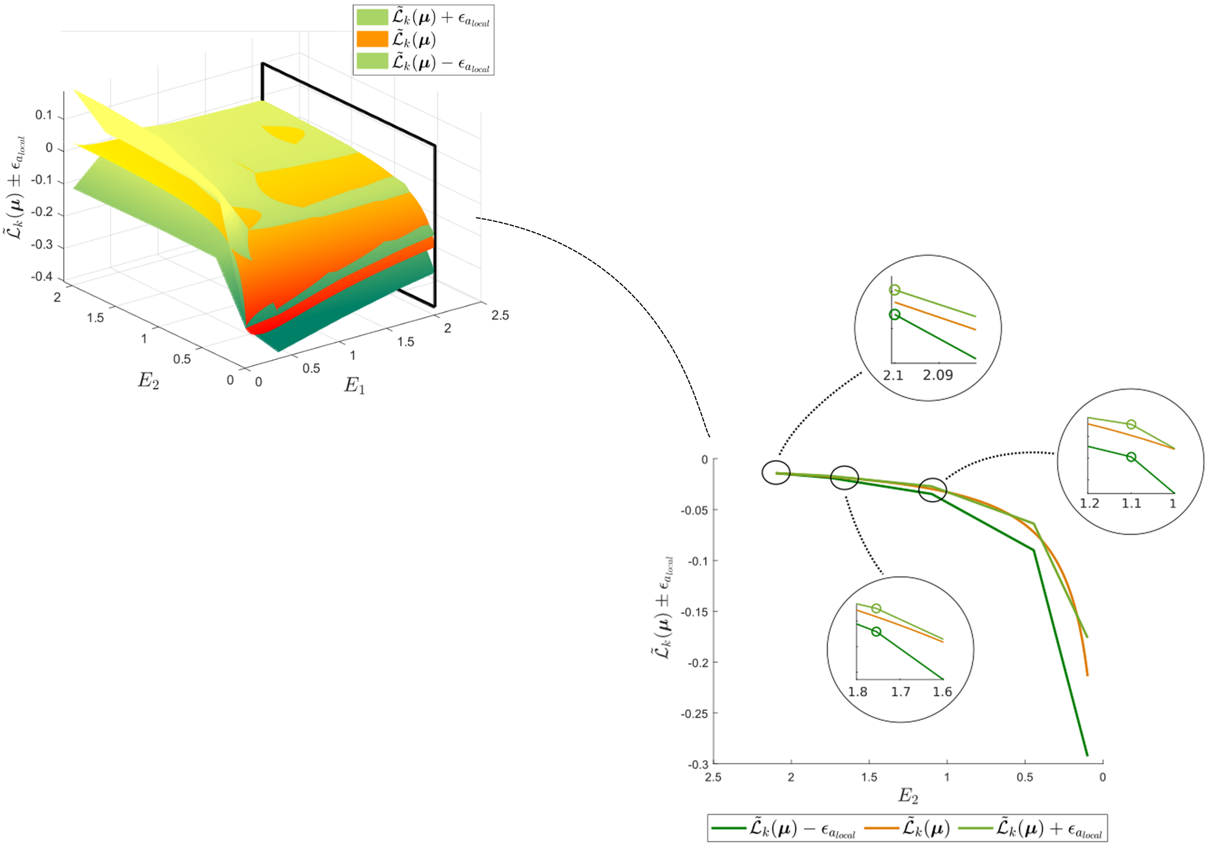


FIGURE 6.31: A reference local displacement output combined with the bounds of the local output obtained from a coarse mesh. The details present the case where the Young's modulus $E_1 = 2.1$ and the points where the parametric domain is discretized.

6.3.3 The effect of PGD convergence on mesh refinement

In this section we assess how the convergence of the PGD, controlled by tolerance τ, affects the mesh refinement process. We run two different examples, the refinement driven by the global error indicator and the refinement driven by the local output error indicators, each one with tolerances for the PGD approximations τ that vary from 10^{-1} until 10^{-4}.

We present the results in terms of the integrals Ψ in Figure 6.32, with the values seen for the tolerance of 10^{-4} are the same seen in Figure 6.28. First and foremost, we note that even for the less strict tolerance values, which do not seem to converge, we always obtain upper bounds for the integrals of the errors (global or local output). The downside of using less strict tolerances is having bounds that are excessively large, although this does not mean that the meshes obtained do not take advantage of the knowledge from the error in each element.

The final meshes obtained from each refinement are shown in Figure 6.33. From this figure we can see that, even for a tolerance value of 10^{-1}, the key features of the problem, such as the

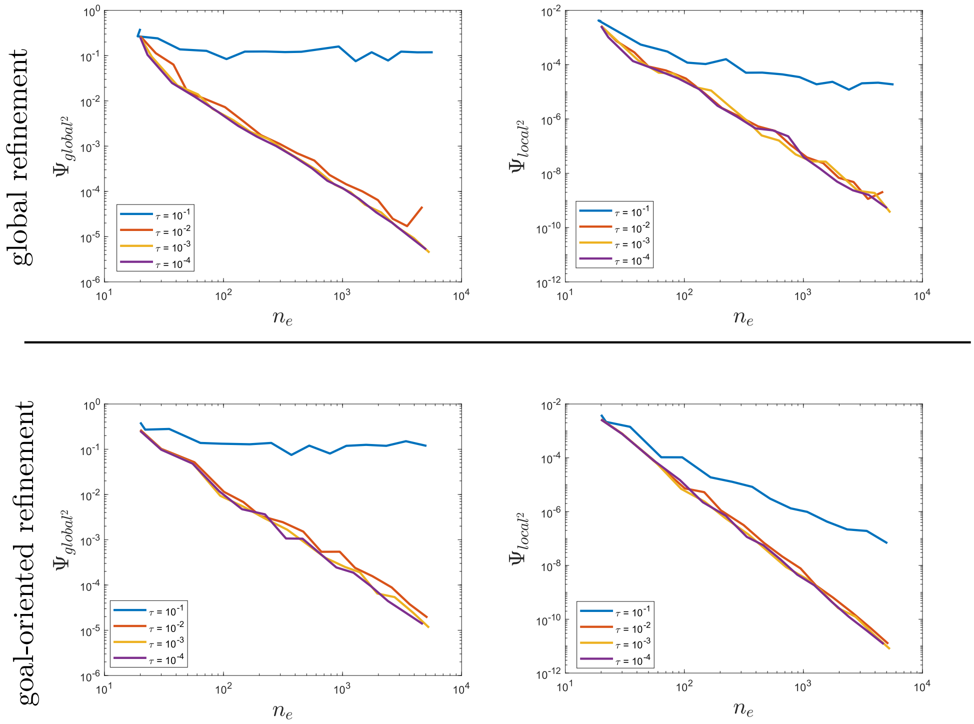


FIGURE 6.32: The integrals of the global (left) and local output (right) error bounds as a function of the number of elements in the mesh, using the displacement at the top as the QoI. We present the cases where the PGD approximations have different τ values, for refinements driven by the global error indicator (top) and by the local output error indicator (bottom).

material singularities, are captured. Naturally, the accuracy is much lower than the one obtained for the meshes that use a tolerance of 10^{-4}. This is reflected in the density of elements in the less refined zones.

The gray dashed lines in Figure 6.34 show, for the goal-oriented refinement process, what happens when, from a mesh refined with a tolerance of 10^{-1}, we compute the PGD solution and Ψ with a tolerance of 10^{-4}. This supports the idea that the meshes obtained from coarse PGD solutions are able to capture the most important information from the problem. If we continue the mesh refinement, we will eventually match the results obtained by using a tolerance of 10^{-4} from the beginning. This is shown, for two specific solutions, by the green and yellow dashed lines in that figure, where we differentiate between the PGD tolerance used during the mesh adaptivity process, and the one that is used to compute the outputs, we use respectively τ_{adapt} and τ_{sol}. Note that, before Figure 6.34, we always considered $\tau = \tau_{\text{adapt}} = \tau_{\text{sol}}$.

In Figure 6.35 we present the error bounds of the displacement outputs obtained with the finest meshes from Figure 6.34 (as in Figure 6.29) in this case using different values for τ_{adapt} and τ_{sol}. This result confirms the global behavior in Figure 6.34, however it should be noted that for

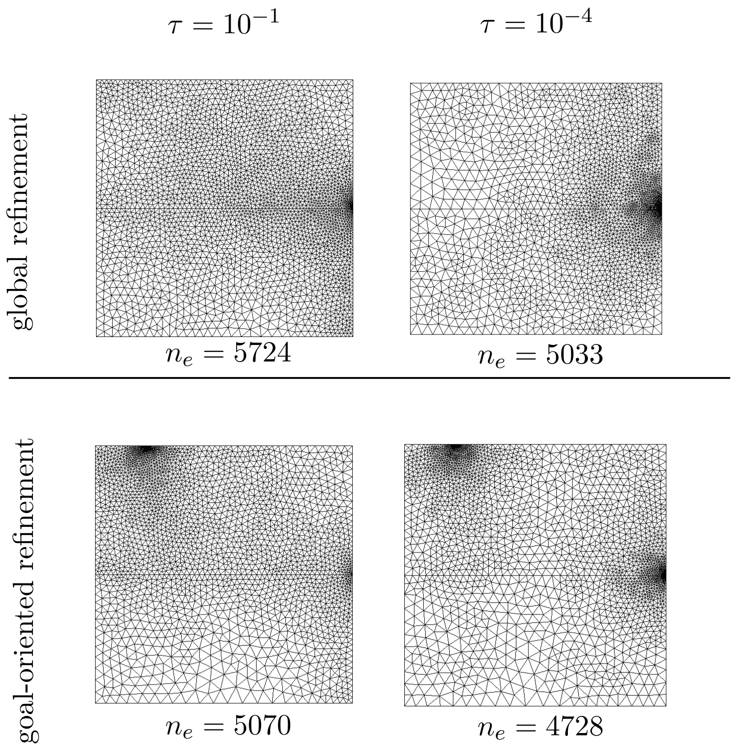


Figure 6.33: The final meshes obtained during the adaptive refinement process using the displacement at the top as the QoI, considering different τ tolerances for the PGD approximations. We consider the cases of a refinement driven by the global error indicator (top) and by the local output error indicator (bottom).

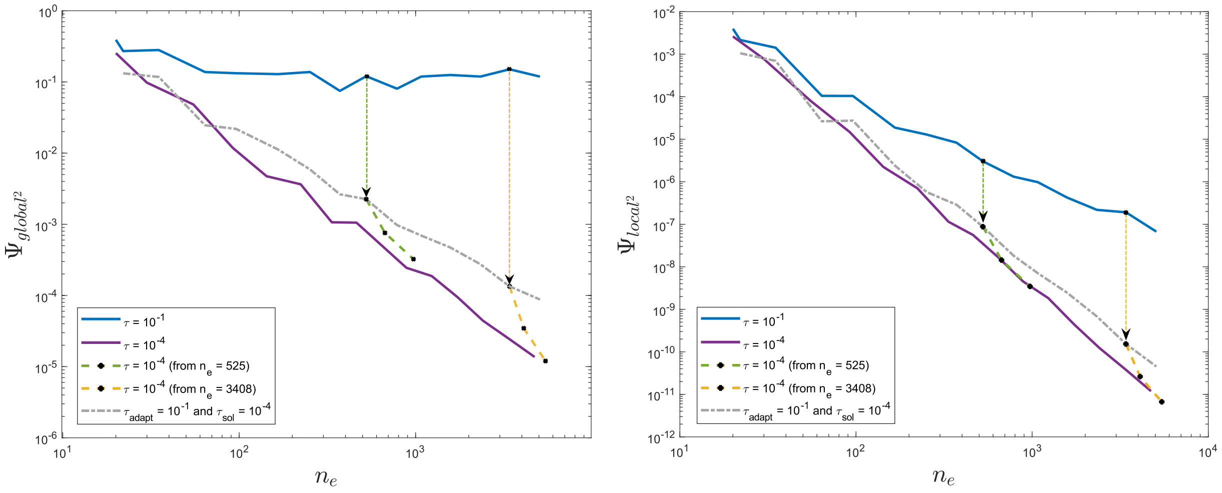


Figure 6.34: Details for the integrals of the global (left) and local output (right) errors as a function of the number of elements in the mesh, for the goal-oriented refinement process using the displacement at the top as the QoI. We present the cases where the PGD approximations have different τ values, and the case where the bounds are obtained with better tolerances than the ones used in the adaptivity process. Details are given for the values obtained by changing the tolerance after $n_e = 525$ and $n_e = 3408$.

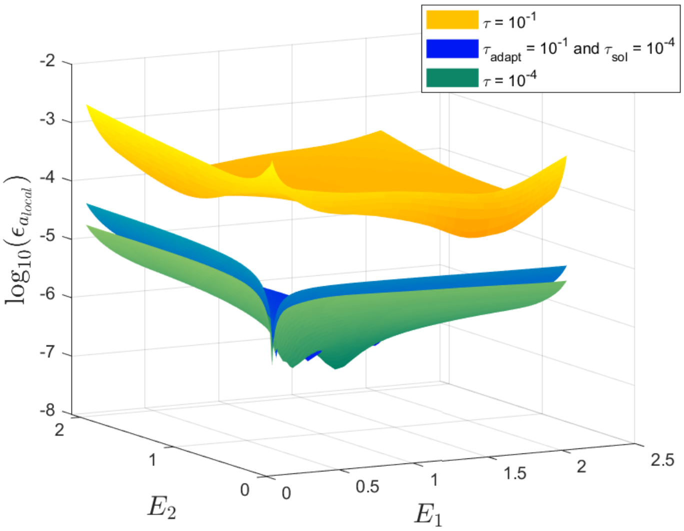


FIGURE 6.35: The values for the error bounds of the displacement at the top for all the combinations of parameters values. We present the cases where the PGD approximations have different τ values, and the case where the output solutions are obtained with a better tolerance then the one used in the adaptivity process. The meshes used are the most refined ones presented in Figure 6.33 for the goal-oriented refinement.

some values of the parameters the non-optimal mesh leads to smaller errors, as can be observed by the blue regions below the green surface. These are located in the regions where $E_1 \approx E_2$, where the problem is nearly homogeneous and the error of the finite element solution is zero. The homogeneous behavior is not captured by the PGD approximations, resulting in marginally different solutions for different meshes.

The behavior demonstrated in Figure 6.34 and 6.35 opens the question of whether it is possible to use a larger tolerance at the beginning of the computations, which is increased after a certain number of steps. The figures seem to indicate that this is a valid option. The unsolved question is how to define when to modify the tolerance. If we truncate too soon, we may have the error coming from the PGD approximations influencing the refinement. Also, if the tolerance is too large, we may completely lose the convergence of the refinement process.

Following this reasoning it would be desirable to have a procedure that controls the total computational cost of the adaptive refinement process, which will strongly depend on the implementation.

We note, however, that the consideration in the initial steps of more optimistic, although less strict, bounds may be a way to move forward, since the availability of a guaranteed interval of the output is critical only for the final mesh.

Chapter 7

Conclusion

In this Thesis we work with a linear elastic problem and start by defining the governing equations for the continuous problem. We describe the finite element formulations used, both compatible and equilibrated, in order to obtain approximated solutions. Computing the compatible and equilibrated solutions is the first step for the dual analysis. The equilibrium formulation uses a hybrid approach, for which a new form of computing the system equations was developed. We present the details of this approach and the main steps of its implementation.

Given the complementary solutions, we proceed to explain how to use them to obtain different error measures. We begin with the straightforward definition of the energy of the error, which is a bound of the global error of each solution, focusing afterwards on a procedure to obtain local outputs, detailing how to obtain a displacement. We show how the local output error bounds the local output and present the necessary steps to compute it.

The parametric form of our problem is presented next, with the introduction of the routines in the `algebraicPGDtools` as a form of efficiently obtaining PGD solutions of problems that are separable PGD solutions. We explain the requirements for the application of the `algebraicPGDtools`, with details on how to write the constitutive relations in a separable form. We use the complementary solutions in a separable PGD format to define the error measures also in a parametric PGD form. The `algebraicPGDtools` gives us flexibility in terms of how to implement our code, and key aspects of the implementation are described.

We proceed to explain how the elemental contribution of the integral of the error in the parametric domain can be used to drive an adaptivity process. The software that we use is described, along with the specific stop criteria used for both the PGD and the adaptivity process.

We then apply the concepts described to simple academic problems. We begin with a one dimensional problem, validating the use of the energy of the error as an error measure and we present several examples to evaluate the behavior of the PGD approximations. We compute the integrals of the energy of the error and the potential energies in the parametric domain for both PGD and FEM and compare the results with the exact solutions.

The integral of the energy of the error was observed to be a good error measure, as it bounds the solutions without any additional conditions, while the error of the energies requires force or displacement driven problems to obtain bounds of the solutions. We further explore the possibilities of the energy of the error by taking advantaged of the in house PGD solver used in this example, developing an adaptivity indicator based on empiric data obtained from several simulations performed. This adaptivity indicator is capable of capturing the optimal parameter (or space region) to be refined, both in terms of p- and h- refinements, leading to lower error values than those obtain with a simply uniform refinement and providing a criterion for adaptation that (almost) always leads to the greater reduction of the error.

Next we move to a two dimensional problem, using the solutions obtained using the routines in the `algebraicPGDtools` toolbox, and focusing on the effects of the global errors in the physical space and its use for mesh adaptivity. We observe the behavior of the PGD approximated complementary solutions in a 2D domain, as well as the error measures that it provide, with the concepts described being tested on an simple application example of a perforated plate. The PGD approximations obtained, in terms of stresses and displacements, were always in agreement with the ones from the FEM approximations.

We performed further studies with the degree of the approximation in space and with the mesh adaptivity strategy, with the results showing that it is viable to use the elemental contributions to the energy of the error obtained from the PGD approximations to adapt the meshes. The errors are analyzed both in terms of their role as a stoping criteria for the PGD method, but also with respect to their accuracy in capturing the global errors of the problem studied. The effects of the errors are translate in the mesh obtained through the mesh refinement process. The new meshes obtained led to improvements of two orders of magnitude when compared to the uniformly refined ones.

Finally, we present the study on a problem with a quantity of interest. We start by validating results with a self-compliant problem, showing the similarities between the global and local output errors measures obtained in this case. We then apply the concepts described to an academic problem which has the characteristic of resulting in solutions where, depending on the parameters, the local output can have opposite signs. This type of behavior highlights the importance of having reliable bounds, since the wrong definition of a sign can lead to misleading conclusions. We focus on assessing the quality of the outputs obtained from meshes derived by adaptive refinements based on either the global or the local output error indicators.

The local output error indicator proved to be efficient in capturing the nature of the local quantity being studied, leading to meshes with the lowest values of the error of the local output, when compared with an uniform refinement and with the use of a global error indicator. We use the meshes obtained with different refinement criteria to present the output and bounds of its error for all the combinations of parameters, detailing critical zones for the simulation such as the edges of the parametric domain. We also assess the region where the errors should be the lowest, which is when the parameters have equal values. Finally, we demonstrate how the bounds for a reference local output are guaranteed as long as they coincide with the points of parametric discretization.

7.1 Future Developments

The proposed approach can be applied to other problems. We envisage, namely, to consider:

- Three-dimensional elastic problems: It is obvious that there are several problems that can only be properly described if we consider a three dimensional approach. The problems we presented here focus on 1D and 2D applications because of the simplicity the verify the results from the former and the ease of the implementation of the later. A natural progression for this research is then to expand the results presented here for 3D problems. This requires efficiently computing equilibrated solutions in 3D, which is an extension of the formulation proposed in [23].
- Mesh adaptivity for multiple loads and multiple outputs: The goal-oriented problem we presented here focused on seeking only one quantity of interest. This is not a limitation of the method, but certain additional steps must be made to obtain such solutions. This can be achieved by either considering a generalization of the combined error proposed in [28] or simply the minimal mesh element size approach proposed by Zienkiewicz [38].

- Different physical models: This thesis focus mainly on linear elastic problems, but the results and analyzes can be generalized to other problems, for which we can have a dual pair of solutions. This means that the error bounds and indicators presented here should work in a similar way when considering, for instance, a thermal problem.

- General adaptivity indicator: When working with the one dimensional problem, presented an adaptivity indicator capable of capturing the optimal region to perform a refinement, without any distinction of the physical and parametric domains. This is easy to achieve when dealing with a one dimensional problem, because both parametric and physical domains are described with one dimension. Expanding this indicator when considering higher dimensions of the physical space is not straight forward, as a decomposition of the physical space is probably necessary to maintain the consistency of the dimensions. We are capable of defining a working indicator for the parametric domain only, but further studies must be performed to obtain relatable indicators for the physical space.

- Geometry as a parameter for general problems: It is still a challenge to use the geometry as one parameter for the PGD, mostly because the steps required to make it work are very intrusive. We briefly presented in the one dimensional problem how we could take into account a geometric parameter, but the steps we took required several simplifications for the problem, that could not be directly translated to a higher dimension problem. Recognizing that imposing co-diffusivity in the equilibrated solutions is a major issue, we are convinced that the proposed approach based on the Piola-Kirchhoff stresses may facilitate the consideration of geometric parameters.

Bibliography

[1] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[2] Stefan Boschert and Roland Rosen. *Digital Twin—The Simulation Aspect*, pages 59–74. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32156-1. doi: 10.1007/978-3-319-32156-1_5. URL https://doi.org/10.1007/978-3-319-32156-1_5.

[3] Francisco Chinesta and Pierre Ladevèze. *Separated Representations and PGD-Based Model Reduction*, volume 554 of *CISM International Centre for Mechanical Sciences*. Springer Vienna, Vienna, 2014. ISBN 978-3-7091-1793-4. doi: 10.1007/978-3-7091-1794-1. URL http://link.springer.com/10.1007/978-3-7091-1794-1.

[4] Amine Ammar, Etienne Pruliere, Julien Férec, Francisco Chinesta, and Elias Cueto. Coupling finite elements and reduced approximation bases. *European Journal of Computational Mechanics*, 18(5-6):445–463, jan 2009. ISSN 1779-7179. doi: 10.3166/ejcm.18.445-463. URL https://www.tandfonline.com/doi/full/10.3166/ejcm.18.445-463.

[5] A. Ammar, F. Chinesta, P. Diez, and A. Huerta. An error estimator for separated representations of highly multidimensional models. *Computer Methods in Applied Mechanics and Engineering*, 199:1872–1880, 2010.

[6] Wil Schilders, Henk a. van der Vorst, and Joost Rommes. Model Order Reduction: Theory, Research Aspects and Applications. *Symbolic Analysis and Reduction of VLSI ...*, 13, 2008. ISSN 00392871. doi: 10.1007/978-3-540-78841-6. URL http://link.springer.com/10.1007/978-3-540-78841-6.

[7] David Modesto, Sergio Zlotnik, and Antonio Huerta. Proper generalized decomposition for parameterized Helmholtz problems in heterogeneous and unbounded domains: Application to harbor agitation. *Computer Methods in Applied Mechanics and Engineering*, 295:127–149, 2015.

[8] Amine Ammar, Antonio Huerta, Francisco Chinesta, Elías Cueto, and Adrien Leygue. Parametric solutions involving geometry: A step towards efficient shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 268:178–193, 2014.

[9] Amaury Courard, David Néron, Pierre Ladevèze, and Ludovic Ballere. Integration of PGD-virtual charts into an engineering design process. *Computational Mechanics*, 57:637–651, 2016.

[10] Ruben Sevilla, Sergio Zlotnik, and Antonio Huerta. Parametric solutions involving geometry integrated with computer-aided design. pages 1–45, feb 2019.

[11] Ruben Sevilla, Sergio Zlotnik, and Antonio Huerta. Solution of geometrically parametrised problems within a cad environment via model order reduction. *Computer Methods in Applied Mechanics and Engineering*, 358:112631, 2020. ISSN 0045-7825.

[12] Pierre Ladevèze and Ludovic Chamoin. On the verification of model reduction methods based on the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 200:2032–2047, 2011.

[13] Francisco Chinesta, Roland Keunings, and Adrien Leygue. *The proper generalized decomposition for advanced numerical simulations: A primer*. Springer International Publishing, Cham, 2014.

[14] Sergio Zlotnik, Pedro Díez, David Gonzalez, Elías Cueto, and Antonio Huerta. Effect of the separated approximation of input data in the accuracy of the resulting PGD solution. *Advanced Modeling and Simulation in Engineering Sciences*, 2:28, 2015.

[15] Ludovic Chamoin and Pierre Ladevèze. Robust control of PGD-based numerical simulations. *European Journal of Computational Mechanics*, 21(3-6):195–207, dec 2012. ISSN 1779-7179. doi: 10.1080/17797179.2012.721499. URL https://www.tandfonline.com/doi/full/10.1080/17797179.2012.721499.

[16] J.F. Debongnie, H.G. Zhong, and P. Beckers. Dual analysis with general boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 122(1-2):183–192, apr 1995. ISSN 00457825. doi: 10.1016/0045-7825(94)00726-4. URL https://linkinghub.elsevier.com/retrieve/pii/0045782594007264.

[17] W Prager and J. L. Synge. Approximations in elasticity based on the concept of function space. *Quarterly of Applied Mathematics*, 5:241–269, 1947.

[18] B. M. Fraeijs de Veubeke. Upper and Lower Bounds in Matrix Structural Analysis. In *B.M. Fraeijs De Veubeke Memorial Volume of Selected Papers*, pages 53–100. Springer Netherlands, Dordrecht, 1964. doi: 10.1007/978-94-009-9147-7_2. URL http://link.springer.com/10.1007/978-94-009-9147-7{_}2.

[19] Pierre-Eric Allier, Ludovic Chamoin, and Pierre Ladevèze. Proper Generalized Decomposition computational methods on a benchmark problem: introducing a new strategy based on Constitutive Relation Error minimization. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):17, dec 2015. ISSN 2213-7467. doi: 10.1186/s40323-015-0038-4. URL http://www.amses-journal.com/content/2/1/17.

[20] Ludovic Chamoin, Pierre Eric Allier, and Basile Marchand. Synergies between the constitutive relation error concept and PGD model reduction for simplified V&V procedures. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1), 2016. ISSN 22137467. doi: 10.1186/s40323-016-0073-9.

[21] Ludovic Chamoin, Florent Pled, Pierre-Eric Allier, and Pierre Ladevèze. A posteriori error estimation and adaptive strategy for PGD model reduction applied to parametrized linear parabolic problems. *Computer Methods in Applied Mechanics and Engineering*, 327:118–146, 2017.

[22] J. P. Moitinho Almeida and Edward A.W. Maunder. *Equilibrium Finite Element Formulations*. John Wiley & Sons, Ltd, Chichester, UK, feb 2017. ISBN 9781118925782. doi: 10.1002/9781118925782. URL http://doi.wiley.com/10.1002/9781118925782.

[23] J P M Almeida and Jonatha Reis. An efficient implementation of stress based finite element approximations for two-dimensional elasticity. 2019. Submitted for publication.

[24] Eric Florentin and Pedro Díez. Adaptive reduced basis strategy based on goal oriented error assessment for stochastic problems. *Computer Methods in Applied Mechanics and Engineering*, 225-228:116–127, 2012. ISSN 00457825. doi: 10.1016/j.cma.2012.03.016. URL http://dx.doi.org/10.1016/j.cma.2012.03.016.

[25] Kenan Kergrene, Serge Prudhomme, Ludovic Chamoin, and Marc Laforest. A new goal-oriented formulation of the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 327:256–276, dec 2017. ISSN 00457825. doi: 10.1016/j.cma.2017.09.018. URL https://doi.org/10.1016/j.cma.2017.09.018https://linkinghub.elsevier.com/retrieve/pii/S0045782517306485.

[26] Kenan Kergrene, Ludovic Chamoin, Marc Laforest, and Serge Prudhomme. On a Goal-Oriented Version of the Proper Generalized Decomposition Method. *Journal of Scientific Computing*, feb 2019. ISSN 0885-7474. doi: 10.1007/s10915-019-00918-1. URL https://doi.org/10.1007/s10915-019-00918-1http://link.springer.com/10.1007/s10915-019-00918-1.

[27] J. P.Moitinho de Almeida and O. J.B.Almeida Pereira. Upper bounds of the error in local quantities using equilibrated and compatible finite element solutions for linear elastic problems. *Computer Methods in Applied Mechanics and Engineering*, 195(4-6):279–296, 2006. ISSN 00457825. doi: 10.1016/j.cma.2004.09.012.

[28] O. J. B. Almeida Pereira and J. P. Moitinho de Almeida. Dual adaptive finite element refinement for multiple local quantities in linear elastostatics. *International Journal for Numerical Methods in Engineering*, (February):n/a–n/a, oct 2010. ISSN 00295981. doi: 10.1002/nme.2834. URL http://arxiv.org/abs/1010.1724http://doi.wiley.com/10.1002/nme.2834.

[29] J P M Almeida. A basis for bounding the errors of proper generalised decomposition solutions in solid mechanics. *International Journal for Numerical Methods in Engineering*, 94(10):961–984, jun 2013. ISSN 00295981. doi: 10.1002/nme.4490. URL http://doi.wiley.com/10.1002/nme.4490.

[30] Pedro Díez, Sergio Zlotnik, Alberto García-González, and Antonio Huerta. Algebraic PGD for tensor separation and compression: An algorithmic approach. *Comptes Rendus Mécanique*, 1:1–14, may 2018. ISSN 16310721. doi: 10.1016/j.crme.2018.04.011. URL https://doi.org/10.1016/j.crme.2018.04.011http://linkinghub.elsevier.com/retrieve/pii/S1631072118300809.

[31] P. Díez, S. Zlotnik, A. García-González, and A. Huerta. Encapsulated pgd algebraic toolbox operating with high-dimensional data. *Archives of Computational Methods in Engineering*, 2019.

[32] Moitinho de Almeida J.P. Díez P. Zlotnick S. Reis, J. Error Estimation for PGD Solutions: A Dual Approach. *In preparation*, 2020. URL http://dx.doi.org/10.1016/j.enganabound.2013.08.016https://linkinghub.elsevier.com/retrieve/pii/S0955799713001756.

[33] Alberto Sibileau, Alberto García-González, Ferdinando Auricchio, Simone Morganti, and Pedro Díez. Explicit parametric solutions of lattice structures with proper generalized decomposition (PGD). *Computational Mechanics*, pages 1–21, jan 2018. ISSN 0178-7675. doi: 10.1007/s00466-017-1534-9. URL http://link.springer.com/10.1007/s00466-017-1534-9.

[34] J.P.M. de Almeida and E.A. Maunder. *Equilibrium Finite Element Formulations*. Wiley, 2016.

[35] I. Babuška and W. C. Rheinboldt. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12:1597–1615, 1978.

[36] Pedro Díez and Antonio Huerta. A unified approach to remeshing strategies for finite element h-adaptivity. *Computer Methods in Applied Mechanics and Engineering*, 176(1-4): 215–229, jul 1999. ISSN 00457825. doi: 10.1016/S0045-7825(98)00338-7. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782598003387.

[37] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[38] O.C. Zienkiewicz and J.Z. Zhu. The three R's of engineering analysis and error estimation and adaptivity. *Computer Methods in Applied Mechanics and Engineering*, 82(1-3):95–113, sep 1990. ISSN 00457825. doi: 10.1016/0045-7825(90)90160-N. URL https://linkinghub.elsevier.com/retrieve/pii/004578259090160N.

Appendices

Appendix A: Error Estimation for PGD Solutions: A Dual Approach

Reis, J., de Almeida, J.P.M., Díez, P. and Zlotnik, S. (2020)
Error Estimation for PGD Solutions: A Dual Approach.
International Journal for Numerical Methods in Engineering.
Accepted Author Manuscript.
doi:10.1002/nme.6452

Appendix B: An Efficient Implementation of Stress Based Finite Element Approximations for Two-Dimensional Elasticity

Moitinho de Almeida, J. and Reis, J. (2020)
An Efficient Methodology for Stress Based Finite Element Approximations in Two-Dimensional Elasticity.
International Journal for Numerical Methods in Engineering.
Accepted Author Manuscript.
doi:10.1002/nme.6458

ATTENTION¡¡
Pages 110 to 132 of the thesis, containing the article mentioned above
are available at the editor’s web
https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6458

Appendix C: Error Estimation for PGD Solutions: Dual Analysis and Adaptivity for Quantities of Interest

Reis, J., de Almeida, J.P.M., Díez, P. and Zlotnik, S.
Error Estimation for PGD Solutions: Dual Analysis and Adaptivity for Quantities of Interest.
Submitted for publication.

ATTENTION¡¡
Pages 134 to 158 of the thesis, containing the article mentioned above
are available at the editor's web
https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6559

Appendix D: Error Estimation and Adaptivity for PGD based on Complementary Solutions

Reis, J., de Almeida, J.P.M., Díez, P. and Zlotnik, S.
Error Estimation and Adaptivity for PGD based on Complementary Solutions.
Submitted for publication.

RESEARCH

Error Estimation and Adaptivity for PGD based on Complementary Solutions Applied to a Simple 1D Problem

Jonatha Reis[1,2], J P Moitinho de Almeida[1], Pedro Díez[2,3*] and Sergio Zlotnik[2]

[*]Correspondence:
pedro.diez@upc.edu
[2]Laboratori de Calcul Numeric (LaCaN), E.T.S. de Ingenieros de Caminos, Canales y Puertos, Universitat Politecnica de Catalunya, Jordi Girona 1, 08034, 24105 Barcelona, Spain
Full list of author information is available at the end of the article

Abstract

Reduced order methods are powerful tools for the design and analysis of sophisticated systems, reducing computational costs and speeding up the development process. Among these reduced order methods, the Proper Generalized Decomposition is a well-established one, commonly used to deal with multi-dimensional problems that often suffer from the *curse of dimensionality*. Although the PGD method has been around for some time now, it still lacks mechanisms to assess the quality of the solutions obtained.

This paper explores the dual error analysis in the scope of the PGD, using complementary solutions to compute error bounds and drive an adaptivity process, applied to a simple 1D problem. The energy of the error obtained from the dual analysis is used to determine the quality of the PGD approximations. We define a new adaptivity indicator based on the energy of the error and use it to drive parametric h- and p- adaptivity processes. The results are positive, with the indicator accurately capturing the parameter that will lead to lowest errors.

Keywords: Error bounds; Error estimation; Proper Generalized Decomposition; Equilibrium formulation

Introduction

Computational methods have become an essential part of most high end engineering projects. They greatly simplify the design and analysis of highly complex systems and are a must for companies that aim to become and stay competitive. When modeling most demanding systems a high computational effort is needed, leading to a low response time between the identification of the details of the model to be considered and the availability of the response. Furthermore, the modification of details in the model leads to re-analyses of the process, further delaying the determination of the desired response. This is a major issue in our current quick paced reality, which relies in the premise that the faster you can come up with a reliable solution, the better. Reduced order methods appear as an answer for this demand.

The main idea behind reduced order methods is to formulate a model that takes only the essential parts of a simulation, reducing the computational time needed to perform a complex analysis while aiming to maintain the accuracy of the results. The Proper Generalized Decomposition (PGD), which we use in this work, is one of the several reduced order methods that exist. The main characteristics of this

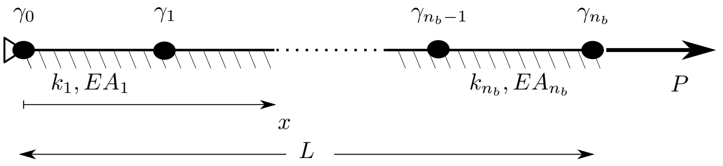


Figure 1: Linear elastic straight bar on an elastic support, with n_b sections. The bar is fixed at the start and has a concentrated load at the end.

method is its a priory nature, which avoids the need to perform the full simulation beforehand in arbitrarily selected instances [1].

One of the challenges that the application of PGD faces is that the method lacks a posteriori estimations tools and adaptivity strategies [2, 3, 4, 5]. Another issue with PGD is its complexity when dealing with geometry as one of the parameters [6, 7].

This paper has the objective of showing the application of a PGD driven adaptivity process to a simple 1D problem. The particular aspect of this implementation is that we simultaneously seek two complementary PGD approximations, one compatible and one equilibrated, which we will use to bound their error (as in [8]) and also to drive the adaptivity process (in the physical and in the parameter space).

The governing equations that describe the problem being considered are presented for the case of a bar divided in two sections, followed by a discussion on error assessment strategies to evaluate the solutions obtained. We proceed to explain the steps used to obtain approximated solutions, first for the finite element method and later using the proper generalized decomposition. We present the parametric form for the approximated solutions and a form of assessing the PGD solutions error, this time specific for each parameter. We apply this error measurement strategy to obtain a novel error indicator which can drive both h- and p- adaptivity processes.

Governing Differential Equations

We consider a simple problem for which an analytic solution can be obtained: a linear elastic straight bar divided in n_b sections, each section with a length (γ), an uniform axial stiffness (EA) and an uniform elastic support (k), connected in sequence, which form a one dimensional lattice structure. Note that when the stiffness of the support is zero, the solution of this problem is trivial.

Although arbitrary imposed displacements or applied forces may be considered at the nodes that connect the different sections, we restrict our study to the problem where the displacement (Δ) is fixed at the start of the first bar and a non-zero force (P) is considered at the free end of the bar, as presented in Figure 1.

We want to obtain a solution for this problem, expressed in terms of either the displacements (u) or of the axial forces (N), that is characterized by the values of γ, EA and k for each section, plus the value of the imposed displacement and of the applied force. This results in, at most, $(3 \times n_b) + 2$ parameters.

We know that the strain in an arbitrary point at the bar is

$$\varepsilon = \frac{\mathrm{d}u}{\mathrm{d}x}. \tag{1}$$

The corresponding axial force is

$$N = EA\,\varepsilon, \tag{2}$$

where E is the Young's modulus at the specified point and A its cross section area. Assuming that the force transmitted by the support is $F = ku$, with k being the elastic stiffness of the support, and knowing that this force has to be balanced by the axial force, we have:

$$\frac{\mathrm{d}N}{\mathrm{d}x} - F = 0. \tag{3}$$

The exact solution of the problem has to satisfy:

$$EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} - F = 0, \tag{4}$$

subjected to

$$\begin{cases} u\,|_{x=0} & = & \Delta; \\ EA\frac{\mathrm{d}u}{\mathrm{d}x}\,|_{x=L} & = & P. \end{cases} \tag{5}$$

Notice that for a bar divided in multiple sections we will need to impose the continuity of displacements and of the corresponding axial force in the $n_b - 1$ connections nodes of the sections.

We can also write the solution for this problem in terms of the axial force, considering first equilibrium and then compatibility. We know that for a given axial force distribution, equilibrium will be satisfied if relation (3) is true, implying that:

$$u = \frac{1}{k}\frac{\mathrm{d}N}{\mathrm{d}x}. \tag{6}$$

Compatibility and the constitutive relations require that the strain corresponding to this displacement has to be equal to the strain associated with the axial force, such that:

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{k}\frac{\mathrm{d}N}{\mathrm{d}x}\right) = \frac{N}{EA}. \tag{7}$$

If we write the problem in terms of the axial force, which has to be continuous, it becomes

$$\frac{1}{k}\frac{\mathrm{d}^2N}{\mathrm{d}x^2} - \frac{N}{EA} = 0, \tag{8}$$

subjected to

$$\begin{cases} \frac{1}{k}\frac{\mathrm{d}N}{\mathrm{d}x}\,|_{x=0} & = & \Delta; \\ N\,|_{x=L} & = & P. \end{cases} \tag{9}$$

Again, we need to impose the continuity of the axial forces and of the corresponding displacements at the connection nodes of a bar divided in multiple sections.

Error Assessment Measurements

In this section we discuss two possible ways to compute the error for approximate solutions. This is made for an arbitrary problem with domain Ω and boundary Γ, which for the bar that we are considering are is its length and the endpoints, respectively. We will first talk about the error in energy and later of the energy of the error. After a brief explanation, these errors indicators will be applied to our problem.

Both the error in energy and the energy of the error can work as global estimators of the solution's behavior, providing error bounds.

Error in Energy

Following [9], the potential energy Π of a mechanical system can be defined as a function of a kinematically admissible displacement u, such that:

$$\Pi(u) = \mathcal{U}(u) - \mathcal{V}(u) = \int_\Omega \mathcal{W}(u) \mathrm{d}\Omega - \mathcal{V}(u), \tag{10}$$

with $\mathcal{U}$ describing the total strain energy, $\mathcal{W}$ the strain energy density, and $\mathcal{V}$ is the work done by the applied forces, defined as:

$$\mathcal{V}(u) = \int_\Omega \bar{b}^T u \, \mathrm{d}\Omega + \int_{\Gamma_t} \bar{t}^T u \, \mathrm{d}\Gamma, \tag{11}$$

where $\bar{b}$ are the body forces, $\bar{t}$ the boundary traction and Γ_t the static boundary. The complementary potential energy Π^c for a statically admissible stress field N can be written in a similar way as

$$\Pi^c(N) = \mathcal{U}^c(N) - \mathcal{V}^c(N) = \int_\Omega \mathcal{W}^c(N) \mathrm{d}\Omega - \mathcal{V}^c(N), \tag{12}$$

where $\mathcal{U}^c$ is the complementary strain energy, $\mathcal{W}^c$ the complementary strain energy density, and $\mathcal{V}^c$ is the work done by the imposed displacements. We can define $\mathcal{V}^c$ as

$$\mathcal{V}^c(N) = \int_{\Gamma_u} (\mathcal{N}^T N)^T \tilde{u} \, \mathrm{d}\Gamma, \tag{13}$$

with Γ_u representing the kinematic boundary, $\mathcal{N}$ the boundary operator, which for the 1D problem is ±1, and $\tilde{u}$ the displacement at the boundary. Considering linear elastic constitutive relations and that the possible influence of initial strains is excluded, we can write the strain energy density as:

$$\mathcal{W}(u) = \frac{1}{2} \left(u'^T EA \; u' + u^T k \; u \right), \tag{14}$$

and the complementary strain energy density:

$$\mathcal{W}^c(N) = \frac{1}{2} \left(N^T \frac{1}{EA} N + N'^T \frac{1}{k} N' \right). \tag{15}$$

We can use the dual solutions to write the average values for the total potential energies and the strain energies, such that:

$$\Pi_a = \frac{1}{2}\Big(\Pi(u) - \Pi^c(N)\Big); \qquad \mathcal{U}_a = \frac{1}{2}\Big(\mathcal{U}(u) - \mathcal{U}^c(N)\Big). \tag{16}$$

The sum of the potential energy and the complementary potential energy is zero for the exact solution [9]. For force driven or displacement driven problems, such as the one that we will consider, the error of each approximate solution is bounded by the sum of their energies. Compatible solutions for force-driven problems have the total energy equal to minus the strain energy, which is a negative value. We choose to work with the symmetric of the total energy, which is positive and converges from bellow, and can be used as an error measure. On the other hand, compatible solutions for displacement-driven problems will present a total energy which is always equal to the strain energy, therefore converging from above. Equilibrium solutions will have an analogous opposite behavior, where force-driven problems will converge from above and displacement-driven from bellow (More about this in [10]).

Energy of the Error

We can compute the exact energy of the error of a compatible or of an equilibrium solution, respectively as:

$$\epsilon_k^2 = \int \left((N_k - N)(\varepsilon_k - \varepsilon) + (F_k - F)(u_k - u)\right)\,\mathrm{d}x; \tag{17}$$

$$\epsilon_s^2 = \int \left((N_s - N)(\varepsilon_s - \varepsilon) + (F_s - F)(u_s - u)\right)\,\mathrm{d}x, \tag{18}$$

where the subscript k indicates solutions that come from kinematically admissible displacements, s solutions from statically admissible axial forces and the lack of a subscript indicates the exact solution. An upper bound of both values is [11]:

$$\epsilon^2 = \int \left((N_k - N_s)(\varepsilon_k - \varepsilon_s) + (F_k - F_s)(u_k - u_s)\right)\,\mathrm{d}x. \tag{19}$$

Unlike the energies of the solutions, this expression is applicable to any type of problem (force driven, displacement driven or mixed). Therefore, from now on we always use the energy of the error instead of the error in an energy.

We call the integrand in (19) an *error density*, defined as:

$$\rho = (N_k - N_s)(\varepsilon_k - \varepsilon_s) + (F_k - F_s)(u_k - u_s). \tag{20}$$

Therefore, we can rewrite equation (19) as:

$$\epsilon^2 = \int \rho\,\mathrm{d}x. \tag{21}$$

It can be shown that [9]:

$$\epsilon^2 \geq \epsilon_k^2; \qquad \text{and} \qquad \epsilon^2 \geq \epsilon_s^2. \tag{22}$$

Compatible and Equilibrated Finite Elements Approximations

For our problem, any general function $f(x) \in \mathcal{H}^1$ (continuous and differentiable) that satisfies $f(0) = \Delta$ is a compatible displacement field, with a normal force distribution $N = u'EA$. Also, any general function $g(x) \in \mathcal{H}^1$ is an equilibrated normal force distribution, provided $g(L) = P$, with a displacement field $u = g'/k$.

When considering the displacement field, we can write the bilinear form of the strain energy product as:

$$a_k(u_\alpha, u_\beta) = \int_0^L \left(k u_\alpha u_\beta + EA u'_\alpha u'_\beta\right) \mathrm{d}x, \tag{23}$$

and the work of the concentrated force is

$$b_k(u_\beta) = P \; u_\beta(L), \tag{24}$$

where $a_k(u_\alpha, u_\beta)$ and $b_k(u_\beta)$ are the bilinear and linear form products, corresponding to compatible (k̲inematically admissible) solutions.

For the equilibrated solution (s̲tatically admissible), we will have:

$$a_s(N_\alpha, N_\beta) = \int_0^L \left(\frac{1}{k} N'_\alpha N'_\beta + \frac{1}{EA} N_\alpha N_\beta\right) \mathrm{d}x, \tag{25}$$

and the work of the imposed displacement is

$$b_s(N_\beta) = \Delta \; N_\beta(0). \tag{26}$$

It is also possible to consider a m̲ixed form

$$a_m(u_\alpha, N_\beta) = \int_0^L \left(u_\alpha N'_\beta + u'_\alpha N_\beta\right) \mathrm{d}x. \tag{27}$$

We use a Galerkin approach to determine the approximate solutions. Assuming a polynomial degree of approximation p, we have:

$$u_h(x) = \sum_{d=0}^{p} \phi_d(x) \; \hat{u}_d = \boldsymbol{\phi}(x) \, \hat{\mathbf{u}}; \qquad N_h(x) = \sum_{d=0}^{p} \phi_d(x) \hat{N}_d = \boldsymbol{\phi}(x) \, \hat{\mathbf{N}}, \tag{28}$$

where vectors $\hat{\mathbf{u}}$ and $\hat{\mathbf{N}}$ represent the coefficients of the approximations for the displacements and axial forces, respectively, and $\boldsymbol{\phi}(x)$ are the polynomial basis.

Substituting (28) into (23) and (25) will lead to the definition of the stiffness and flexibility matrices, $\boldsymbol{\mathcal{K}}$ and $\boldsymbol{\mathcal{F}}$, respectively,

$$\boldsymbol{\mathcal{K}} = \int_0^L \Big(k \, \boldsymbol{\phi}(x) \boldsymbol{\phi}^T(x) + EA \; \boldsymbol{\phi'}(x) \boldsymbol{\phi'}^T(x)\Big) \, \mathrm{d}x; \tag{29}$$

$$\boldsymbol{\mathcal{F}} = \int_0^L \left(\frac{1}{k} \boldsymbol{\phi'}(x) \boldsymbol{\phi'}^T(x) + \frac{1}{EA} \boldsymbol{\phi}(x) \, \boldsymbol{\phi}^T(x)\right) \mathrm{d}x. \tag{30}$$

We can define a matrix that projects the equilibrated solutions onto the compatible ones, as in (27), such that:

$$\boldsymbol{\mathcal{S}} = \int_0^L \left(\boldsymbol{\phi}(x)\boldsymbol{\phi'}^T(x) + \boldsymbol{\phi'}(x)\boldsymbol{\phi}^T(x)\right) \mathrm{d}x. \tag{31}$$

The vectors of applied forces and imposed displacements are obtained substituting (28) into (24) and (26), so that:

$$\mathbf{Q} = \boldsymbol{\phi}^T(L)P; \qquad \mathbf{e} = \boldsymbol{\phi}^T(0)\Delta. \tag{32}$$

The resulting finite element compatible and equilibrated systems are, respectively

$$\boldsymbol{\mathcal{K}}\hat{\mathbf{u}} = \mathbf{Q}; \qquad \boldsymbol{\mathcal{F}}\hat{\mathbf{N}} = \mathbf{e}. \tag{33}$$

We can obtain the finite element approximation for the displacements u and axial forces N by solving the equations in (33) and using the results at equations (28).

Compatible and Equilibrated PGD Approximations

A PGD formulation is going to be used to obtain the approximations of the solutions of each complementary problem, as in [8]. We define $\boldsymbol{\mu}$, a vector of $\mathcal{D}$ parameters $\mu_1, \mu_2, \ldots, \mu_{\mathcal{D}}$, each defined in its own domain $\Omega_{\mu_i} \subset \mathbb{R}$, with $i = 1, 2, \ldots, \mathcal{D}$. Then, the vector of parameters $\boldsymbol{\mu}$ is defined so that $\boldsymbol{\mu} \subset \Omega_\mu = \Omega_{\mu_1} \otimes \Omega_{\mu_2} \otimes \cdots \otimes \Omega_{\mu_{\mathcal{D}}} \subset \mathbb{R}^{\mathcal{D}}$. Each solution will depend on the $\mathcal{D}$ parameters and will be approximated as a sum of $\mathcal{N}$ modes of $\mathcal{D}$ independent functions of each variable. We can write a general PGD approximated solution as:

$$f_{pgd}(x, \boldsymbol{\mu}) = \sum_{d=0}^{p} \phi_d(x)\hat{f}_{d_{pgd}}(\boldsymbol{\mu}) = \boldsymbol{\phi}(x)\hat{\mathbf{f}}_{pgd}(\boldsymbol{\mu}), \tag{34}$$

where,

$$\hat{\mathbf{f}}_{pgd}(\boldsymbol{\mu}) = \sum_{m=1}^{\mathcal{N}} \bar{\mathbf{f}}^m \prod_{j=1}^{\mathcal{D}} F_j^m(\mu_j), \tag{35}$$

with $\bar{\mathbf{f}}$ representing the coefficients of the approximation in the physical space. A linear combination of polynomials of degree p_j is used to represent the function:

$$F_j^m(\mu_j) = \sum_{d_j=0}^{p_j} \Phi_{d_j}(\mu_j)\hat{F}_{j,d_j}^m, \tag{36}$$

which is a generalization of (28) where $\hat{F}$ represents the coefficients of the approximation in the parameters.

A fixed point iteration is used to determine the terms in each product, or more precisely, the coefficients $\hat{F}$ and $\bar{\mathbf{f}}$. This iteration procedure will try to minimize the extended Galerkin residual of (23) or (25). For a generic term $n > 1$, $\tilde{\mathbf{f}}$, is written as

$$\tilde{\mathbf{f}} = \sum_{m=1}^{n-1} \bar{\mathbf{f}}^m \prod_{j=1}^{\mathcal{D}} F_i^m(\mu_j) + \bar{\mathbf{f}}^n \prod_{j=1}^{\mathcal{D}} F_j^n(\mu_j). \tag{37}$$

Defining

$$\tilde{\mathbf{f}}^{n-1} = \sum_{m=1}^{n-1} \bar{\mathbf{f}}^m \prod_{j=1}^{\mathcal{D}} F_j^m(\mu_j) \quad \text{and} \quad \Delta\tilde{\mathbf{f}}^n = \bar{\mathbf{f}}^n \prod_{j=1}^{\mathcal{D}} F_j^n(\mu_j), \tag{38}$$

we can further simply $\tilde{\mathbf{f}}$ as:

$$\tilde{\mathbf{f}} = \tilde{\mathbf{f}}^{n-1} + \Delta\tilde{\mathbf{f}}^n. \tag{39}$$

The extended residual is obtained by considering the integration domain of (23) or (25) including all elements and all parameters. We want to impose that

$$a(\tilde{\mathbf{f}}, \mathbf{g}) - b(\mathbf{g}) = 0 \quad \forall \mathbf{g}, \tag{40}$$

or

$$a(\Delta\tilde{\mathbf{f}}^n, \mathbf{g}) - (b(\mathbf{g}) - a(\tilde{f}^{n-1}, \mathbf{g})) = 0 \quad \forall \mathbf{g}. \tag{41}$$

In order to compute a specific function, a fixed point iteration scheme is assumed with all other functions being constants.

In the fixed point scheme only the physical space or one variable are considered at a time, e.g. μ_α, so that the test space (the $\mathbf{g}$'s) are restricted to the basis used for that variable.

To determine the values of $\hat{\mathbf{f}}^n$ for all elements, which define the solution in the physical space, we solve:

$$\tilde{\mathbf{f}} = \tilde{\mathbf{f}}^{n-1} + \bar{\mathbf{f}}^n \prod_{j=1}^{\mathcal{D}} F_j^n(\mu_j). \tag{42}$$

For a specific function $F_\alpha^n(\mu_\alpha)$ the iteration procedure determines the coefficients $\hat{F}_\alpha^n$ that satisfy:

$$\tilde{\mathbf{f}} = \tilde{\mathbf{f}}^{n-1} + \bar{\mathbf{f}}^n \; F_\alpha^n(\mu_\alpha) \prod_{j=1;j\neq\alpha}^{\mathcal{D}} F_j^n(\mu_j). \tag{43}$$

Finally, the parametric PGD approximation in terms of the displacements is:

$$\hat{\mathbf{u}}_{pgd}(\boldsymbol{\mu}) = \sum_{m=1}^{\mathcal{N}} \bar{\mathbf{u}}^m \prod_{j=1}^{\mathcal{D}} U_j^m(\mu_j), \tag{44}$$

and in terms of axial forces:

$$\hat{\mathbf{N}}_{pgd}(\boldsymbol{\mu}) = \sum_{m=1}^{\mathcal{N}} \bar{\mathbf{N}}^m \prod_{j=1}^{\mathcal{D}} \tilde{N}_j^m(\mu_j). \tag{45}$$

As is typical for the PGD method, the greater the number of terms in the sum, the best should be the approximation achieved. The converge criteria adopted here is the difference between the strain energy of the current pair of solutions (compatible and equilibrium). The process is stopped when either the error is lower than a required tolerance, or when the process stagnates.

When non homogeneous boundary conditions are considered ($u(0) \neq 0$ and $N(L) \neq 0$, respectively for the compatible and for the equilibrium models) their effect are taken into account in the first term of the sum in (36), so that for the other terms the boundary conditions are homogeneous.

Parametric Problem and Approximated Error Measurements

We defined $\boldsymbol{\mu}$ as a vector of $\mathcal{D}$ parameters $\mu_1, \mu_2, \ldots, \mu_{\mathcal{D}}$. Since for our specific example the integrals in the physical space are already considered in the system matrices presented in section 2, we can write the parametric finite element approximations of the potential energies directly in terms of the stiffness and flexibility matrices and of the vectors of the applied force and imposed displacement, such that:

$$\Pi_{fem}(\hat{\mathbf{u}}(\boldsymbol{\mu})) = \frac{1}{2}\hat{\mathbf{u}}^T(\boldsymbol{\mu})\,\boldsymbol{\mathcal{K}}(\boldsymbol{\mu})\,\hat{\mathbf{u}}(\boldsymbol{\mu}) - \hat{\mathbf{u}}^T(\boldsymbol{\mu})\mathbf{Q}, \tag{46}$$

and

$$\Pi_{fem}^c(\hat{\mathbf{N}}(\boldsymbol{\mu})) = \frac{1}{2}\hat{\mathbf{N}}^T(\boldsymbol{\mu})\boldsymbol{\mathcal{F}}(\boldsymbol{\mu})\,\hat{\mathbf{N}}(\boldsymbol{\mu}) - \hat{\mathbf{N}}^T(\boldsymbol{\mu})\mathbf{e}. \tag{47}$$

We apply the same concept to obtain the bound of the energy of the error, so that:

$$\epsilon_{fem}^2(\boldsymbol{\mu}) = \hat{\mathbf{u}}^T(\boldsymbol{\mu})\,\boldsymbol{\mathcal{K}}(\boldsymbol{\mu})\,\hat{\mathbf{u}}(\boldsymbol{\mu}) + \hat{\mathbf{N}}^T(\boldsymbol{\mu})\boldsymbol{\mathcal{F}}(\boldsymbol{\mu})\,\hat{\mathbf{N}}(\boldsymbol{\mu}) - 2\,\hat{\mathbf{N}}^T(\boldsymbol{\mu})\boldsymbol{\mathcal{S}}\,\hat{\mathbf{u}}(\boldsymbol{\mu}), \tag{48}$$

or we can simply define it in terms of the energy density as:

$$\epsilon_{fem}^2(\boldsymbol{\mu}) = \int \rho_{fem}(\boldsymbol{\mu})\,\mathrm{d}x. \tag{49}$$

When considering the error measurements in the parametric form, we will obviously have a different result for each combination of parameters we use. Therefore, it is necessary to have an additional way to determine the quality of the solutions obtained, that also considers the effects of the parametric domain. One simple solution is to integrate the error measurements obtained in the parametric domain, resulting in a single value that accounts for the quality of all possible solutions. For

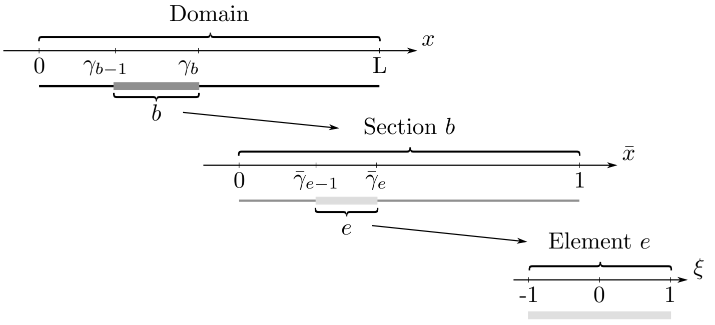


Figure 2: Geometric mappings: from bar domain to section and from section domain to element.

the energy of the error in particular, it is worth noting that if the integral in space is extended to the parameters, the bounding properties still hold, so that:

$$\begin{aligned} &\int_{\Omega_{\boldsymbol{\mu}}} \epsilon_{fem}^2(\boldsymbol{\mu})\,\mathrm{d}\Omega_{\boldsymbol{\mu}} \geq \int_{\Omega_{\boldsymbol{\mu}}} \epsilon_{k_{fem}}^2(\boldsymbol{\mu})\,\mathrm{d}\Omega_{\boldsymbol{\mu}}; \quad \text{or} \\ &\int_{\Omega_{\boldsymbol{\mu}}} \epsilon_{fem}^2(\boldsymbol{\mu})\,\mathrm{d}\Omega_{\boldsymbol{\mu}} \geq \int_{\Omega_{\boldsymbol{\mu}}} \epsilon_{s_{fem}}^2(\boldsymbol{\mu})\,\mathrm{d}\Omega_{\boldsymbol{\mu}}. \end{aligned} \tag{50}$$

We can obtain similar approximated error measures using the PGD approximations. We need only to substitute the finite element approximated displacements $\hat{\mathbf{u}}$ and axial forces $\hat{\mathbf{N}}$ for their PGD counterparts $\hat{\mathbf{u}}_{pgd}$ and $\hat{\mathbf{N}}_{pgd}$.

Practical Aspects of the Discretization of the Physical Domain

A global coordinate system ($x \in [0, L]$) is used, where the coordinates of the initial and final nodes of section b are γ_{b-1} and γ_b, with $\gamma_0 = 0$ and $\gamma_{n_b} = L$. Therefore, the geometric parameters correspond to the γ_i, with $i = 1, \ldots, (n_b - 1)$.

Each section, which is divided in $n_{e[b]}$ elements, uses an intermediate coordinate system ($\bar{x}_{[b]} \in [0, 1]$). The coordinates of the initial and final nodes of element e are $\bar{\gamma}_{[b]e-1}$ and $\bar{\gamma}_{[b]e}$, with $\bar{\gamma}_{[b]0} = 0$ and $\bar{\gamma}_{[b]n_e} = 1$. In the following, we replace $\bar{\gamma}_{[b]e}$ with $\bar{\gamma}_e$, $\bar{x}_{[b]e}$ with $\bar{x}_e$ and $\bar{x}_{[b]}$ with $\bar{x}$, unless the reference to the section of the element is necessary.

In the elements domain, a local coordinate system is used ($\xi \in [-1, 1]$), which is linearly mapped to $\bar{x}$, which is then linearly mapped to x.

Using ϕ_0 and ϕ_1 to represent the linear interpolation functions associated with the end nodes of an interval we have:

$$\bar{x}(\xi) = \phi_0(\xi)\,\bar{\gamma}_{e-1} + \phi_1(\xi)\,\bar{\gamma}_e \quad \text{and} \quad x(\bar{x}) = \phi_0(\bar{x})\,\gamma_{b-1} + \phi_1(\bar{x})\,\gamma_b. \tag{51}$$

These transformations are illustrated in Figure 2.

Each section may be divided in several finite elements. Normally, for problems with fixed geometry, only one mapping is used, from the frame of element (ξ) to the global coordinates (x).

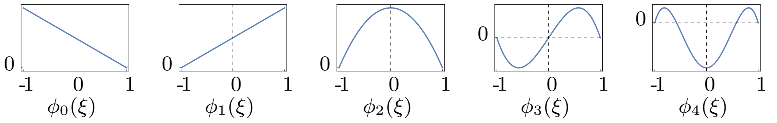


Figure 3: Basis for one dimensional approximations.

Since both mappings are linear the Jacobian of each transformation is constant, and the Jacobian of the double mapping, J, is equal to their product. The derivative of an arbitrary function, f, is obtained as

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\mathrm{d}f}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}\bar{x}}\frac{\mathrm{d}\bar{x}}{\mathrm{d}x} = \frac{2}{(\bar{\gamma}_e - \bar{\gamma}_{e-1})}\frac{1}{(\gamma_b - \gamma_{b-1})}\frac{\mathrm{d}f}{\mathrm{d}\xi} = \frac{1}{J_e}\frac{\mathrm{d}f}{\mathrm{d}\xi}, \tag{52}$$

and its integral is

$$\int f \,\mathrm{d}x = \int f J_e \,\mathrm{d}\xi. \tag{53}$$

Our basis, in local coordinates, are obtained by combining the linear interpolation functions with the Legendre polynomials $\mathcal{L}_i(\xi)$:

$$\begin{cases} \phi_0(\xi) = \frac{1-\xi}{2}, \\ \phi_1(\xi) = \frac{1+\xi}{2}, \\ \phi_i(\xi) = \phi_0(\xi)\ \phi_1(\xi)\ \mathcal{L}_{i-2}(\xi), \quad \text{for } i > 1 \end{cases} \tag{54}$$

We can re-write, for example, the stiffness and flexibility matrices in the local frame of the elements, by applying (52) and (53):

$$\mathcal{K}_{e_{i,j}} = \int_{-1}^{1} \left(k\ \phi_i \phi_j J_e + EA\ \phi_i' \phi_j' \frac{1}{J_e} \right)\ \mathrm{d}\xi; \tag{55}$$

$$\mathcal{F}_{e_{i,j}} = \int_{-1}^{1} \left(\frac{1}{k}\phi_i' \phi_j' \frac{1}{J_e} + \frac{1}{EA}\phi_i \phi_j J_e \right)\ \mathrm{d}\xi. \tag{56}$$

The complete system of equations for each problem combines the assembled stiffness (or flexibility) matrices of each element of each section, together with additional constraints that impose displacement continuity (5) or equilibrium of normal forces (9) between sections. This corresponds to impose:

- continuity of displacements and (weak) equilibrium of nodal forces, or
- equilibrium of normal forces and (weak) continuity of the displacements.

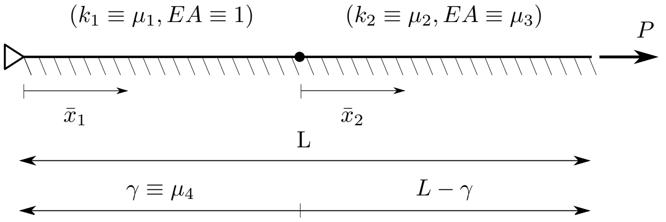

$(k_1 \equiv \mu_1, EA \equiv 1)$
$(k_2 \equiv \mu_2, EA \equiv \mu_3)$
P
$\bar{x}_1$
$\bar{x}_2$
L
$\gamma \equiv \mu_4$
$L - \gamma$

Figure 4: A problem with two sections, connected at an intermediate point, defined by γ.

For the compatible formulation, to have a continuous displacement at the node between sections b and $b+1$, we must impose that:

$$u_{[b]n_e}(1) = u_{[b+1]1}(-1) \implies \sum_{i=0}^{p_e} \phi_i(1)\hat{u}_{[b]n_e,i} = \sum_{i=0}^{p_e} \phi_i(-1)\hat{u}_{[b+1]1,i} \implies \hat{u}_{[b]n_e,1} = \hat{u}_{[b+1]1,0}. \quad (57)$$

This condition can be expressed in matrix form as

$$\mathbf{L}_{end}\hat{\mathbf{u}}_b - \mathbf{L}_{start}\hat{\mathbf{u}}_{b+1} = 0. \quad (58)$$

The transpose of these matrices reflects the effect of the force at the node (transmitted between sections b and $b+1$) in the equations of weak equilibrium of the adjacent sections.

For the equilibrated formulation, the process is complementary: we impose continuity of the nodal forces and the displacement of the interface is accounted for in the weak compatibility conditions of each adjacent bar. The matrices involved are the same.

Note that, in case we wanted to consider a concentrated force applied at a node, or a relative imposed displacement, the equations need to be modified accordingly.

In the practical implementation of the problem, we apply these mapping considerations for all the equations described so far, but we will omit them from now on to reduce repetition of the text.

Characterization of the Test Case

To apply the concepts presented here, we consider the problem shown in Figure 4, with the following characteristics:

- The bar is composed of two sections, with a total summed length L of one;
- Each section has its own support stiffness k_b;
- The axial stiffness EA for the first section has a unit value;
- The axial stiffness of the second section is equal to β;
- The length of the first section is equal to γ.

The parameters for the PGD with their respective limits are:

$$\begin{cases} \mu_1 \equiv k_1, & \text{with } k_1 \in [0.1, 10]; \\ \mu_2 \equiv k_2, & \text{with } k_2 \in [0.1, 10]; \\ \mu_3 \equiv \beta, & \text{with } \beta \in [0.1, 10]; \\ \mu_4 \equiv \gamma, & \text{with } \gamma \in [0.4, 0.6]. \end{cases} \tag{59}$$

The calculations are performed considering an applied force at the free end ($P_{x=L} = 1$, $\Delta_{x=0} = 0$).

Unless otherwise stated, the polynomial approximations are linear for all parameters. Notice that these degrees can lead to solutions which are not accurate, but they serve the purpose of this paper, which is to identify regions where the simulations can be improved.

The tolerances τ adopted for the fixed point iteration scheme and the PGD enrichment process are $\tau_{fix} = 1 \times 10^{-3}$ and $\tau_{pgd} = 1 \times 10^{-6}$, respectively, with the maximum amount of iterations being 3 for the fixed point and 3001 for the PGD. Setting such low number of iterations for the fixed point scheme may lead to more terms for the convergence of the PGD, but drastically decreases the computational time, which is a phenomenon also observed in [12, 13], but should not be generalized for all problems, as this behavior is case dependent.

To further improve the convergence of the solutions, the limits for the parameters k_1, k_2 and β were mapped to a logarithmic scale $k_1 = 10^{\hat{k}_1}$, $k_2 = 10^{\hat{k}_2}$ and $\beta = 10^{\hat{\beta}}$, with ranges $\hat{k}_1 \in [-1, 1]$, $\hat{k}_2 \in [-1, 1]$ and $\hat{\beta} \in [-1, 1]$. This mapping does not affect the PGD approximations, serving only to diminish the influence of the edges in the solutions. Additional simulations must be performed with different limits to assess the effect of this modification.

We now present solutions in terms of the error in energy and the energy of the error. The simulations were performed using quadratic approximations in space, in order to better visualize the bounds of the solutions. The bounding characteristics discussed in the previous section are observed in Figure 5, which presents selected results for our problem. The left part of the Figure shows the integral of the energies of the PGD and FEM approximations, while the right part shows the integral of the energy of the error. The energies for the FEM solutions are very close to the exact solutions and are represented as details in the figure. Notice that the exact energy of the error is zero and therefore it is not shown in the figure while the exact integral of the error in energy is greater than zero and is presented in the left figure with a black line. The energies of the compatible and equilibrated solutions will have a convergence from above and bellow, respectively, when the problem is force-driven, which is the behavior observed. On the other hand, for the bound of the energy of the error we have an upper bound, meaning that it should be always greater than the values found for the exact solutions. From the figure it is also possible to see that, as the number of PGD modes increases, the two complementary PGD solutions converge to the approximation of the FEM, always preserving their bounding characteristics.

Figure 6 shows the relative difference between the approximations obtained from the PGD and FEM models. The integral of the energies of the PGD model will

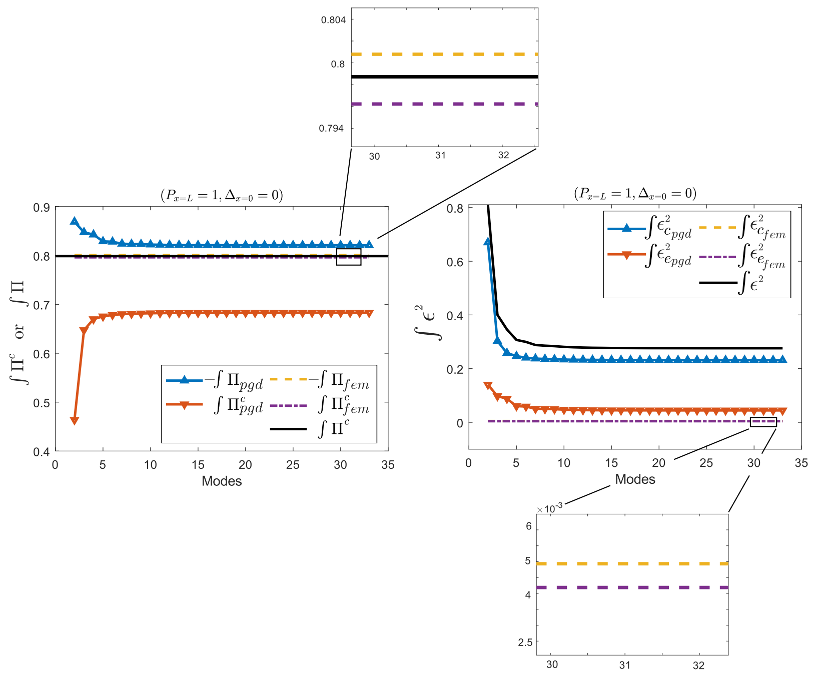


Figure 5: Integral of the energies for both PGD and FEM approximations.

converge to the FEM energy as the number of modes in the solution increases. The results obtained for the integral of the energy of the solutions and the energy of the error are virtually the same. This sort of behavior is expected for force or displacement driven problem, due to the orthogonality of the FEM solutions, but is not the case for mixed problems.

Figure 7 shows the behavior of the relative difference between the integral of the energies of the PGD and FEM approximations and the average potential or strain energies as the degree or the number of elements per section used to obtain the solutions increase. The solutions behave as expected, with better results being achieved as the degree or the number of elements increases.

Notice that for higher degree approximations, the solution starts to lose its convergence rate and also the initial solutions are worse than the ones obtained with smaller degrees. The decrease of the convergence rate is caused by a limit in the number of PGD modes due to the tolerance of the simulations. An indication that the solution is being constrained by the tolerance is that the number of PGD modes for a higher degree or number of elements may decrease. This can be seen for the solution with $p = 5$ and $h = 4$ and $h = 5$. Decreasing the tolerance can recover the rate of convergence, although this will not fix the poor results for the initial modes. We believe these poor results are a consequence of the complexity of the higher degree solutions. This is suggested by the fact that this behavior is not observed when increasing the number of elements of the solution. The results in terms of the error in energy and energy of the error are again the same and were omitted from Figure 7.

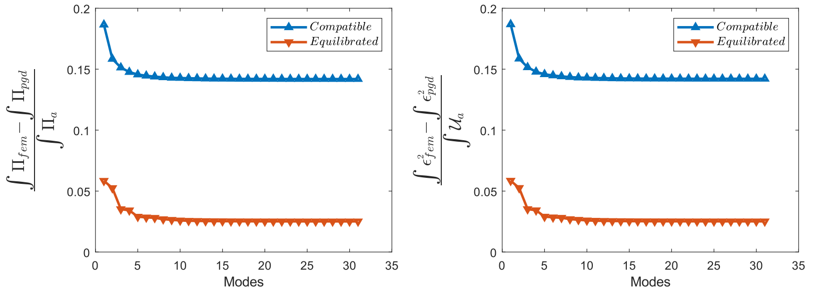


Figure 6: Relative difference between the PGD and FEM integral of the solutions energies and the average total absolute energy of the FEM solution (left). Relative difference between the PGD and FEM integral of the error energies and the average strain energy of the FEM error (right)

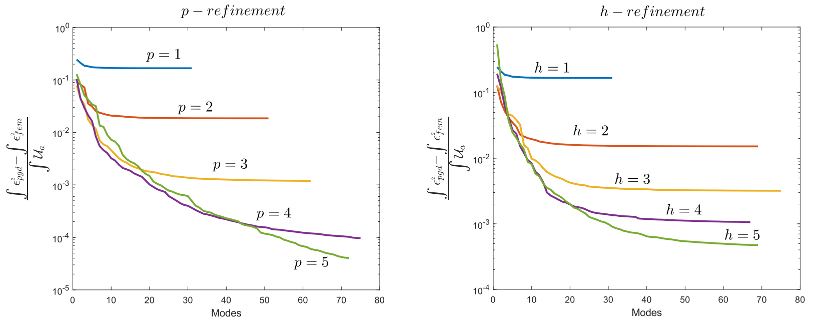


Figure 7: Relative difference for the integral of the solutions energy and the integral of the error energy for the PGD approximation with different number of degrees and elements.

The error in energy is simpler to visualize, as it is based in the energy of the solutions which has a physical meaning. Also, each compatible and equilibrated solution has its own energy, allowing for individual analyses of the error. In the other hand, the bounds that are seen in the error in energy are limited to situations when we have force or displacement driven problems, which is not the case for the energy of the error. That being said, from this point on we will base or solutions solely in the energy of the error, as it provides error bounds for general problems.

Solution Adaptivity

Adaptivity mesh techniques are designed to improve approximated solutions by modifying either the element mesh configuration or their degrees of freedom [9]. One of the key points in the mesh adaptivity process is the proper definition of the relation between the solution convergence and the element size and/or degree of the approximation functions. This convergence dependency is expected to be precise when a smooth solution is being studied and the mesh that represents the solution domain has a large enough number of elements.

We can usually divide the adaptivity process in three categories: the modification of the size of the elements in a mesh (h- adaptivity); the modification of the approximation degree for the functions (p- adaptivity); and a combination of both these process (hp- adaptivity). This paper explores the aspects of the h- and p- adaptivities, but not the combination of both.

We want to obtain an adaptivity indicator that is capable of capturing the best regions to refine, without differentiating between the physical or the parametric space. To achieve this, we collect in variable χ both the parameters and the physical space. This means that, from this point on, a reference to *the parameters* χ *of the problem* may be referring to a material parameter μ or to the parameter x of the physical space.

The solutions presented will be shown in terms of the number of elements n_h or the degrees of the polynomial approximations p. Our specific problem has a total number of parameters n_χ equal to five: four μ's and the physical space, which we divide in the two sections, b_1 and b_2.

Adaptivity indicator

A question that arises is how do we use the complementary solutions to decide where to refine the model. We know that the integral of the energy of the error covers the whole parameters and physical space domain and therefore it is always the same, no matter in which order we performs the integrals. The idea was, then, to look at the effects of each parameter independently by performing the integral of the error density ρ in all parameters but one. If this function is constant, it means that the parameter is not affecting the error bound, indicating that we need to look at the derivatives of the functions. This lead to the idea of working with the derivative of ρ with respect to all parameters. After numerical tests to assess how to relate the derivative of the augmented set of parameters χ with the effect of that parameter on the error, the following expression for an error indicator ι was selected:

$$\iota(\chi) = \int\int (\chi - C(\chi)) \frac{\partial \rho(\xi, \boldsymbol{\mu})}{\partial \chi} \,\mathrm{d}\Omega \,\mathrm{d}\Omega_{\boldsymbol{\mu}}, \tag{60}$$

where:

$$C(\chi) = \frac{\int\int \chi \,\rho(\xi, \boldsymbol{\mu}) \,\mathrm{d}\Omega \,\mathrm{d}\Omega_{\boldsymbol{\mu}}}{\int\int \rho(\xi, \boldsymbol{\mu}) \,\mathrm{d}\Omega \,\mathrm{d}\Omega_{\boldsymbol{\mu}}}. \tag{61}$$

This expression was selected after testing several dimensionally consistent alternatives, including combinations of the derivatives of ρ with respect to the parameters. From this study, we concluded that (60) is the alternative that best captures the regions where refinement is required. Note that this expression is not used to control if the process has converged, only where to refine next. We have not yet found a theoretical background that supports its application.

Considering ρ, which is non negative, as a pseudo mass density, we can interpret $\iota(\chi)$ as the first order moment of the derivative of the error with respect to the center of mass of the domain. Recalling that the subscript k indicates solutions that come from $\underline{\text{k}}$inematically admissible displacements and s that the solutions come from $\underline{\text{s}}$tatically admissible axial forces, we can write ρ for an element, considering the domain decomposition particularities, as:

$$\begin{aligned}\rho(\xi,\boldsymbol{\mu}) = J((&N_k(\xi,\boldsymbol{\mu}) - N_s(\xi,\boldsymbol{\mu}))(\varepsilon_k(\xi,\boldsymbol{\mu}) - \varepsilon_s(\xi,\boldsymbol{\mu}))+ \\ &(F_k(\xi,\boldsymbol{\mu}) - F_s(\xi,\boldsymbol{\mu}))(u_k(\xi,\boldsymbol{\mu}) - u_s(\xi,\boldsymbol{\mu}))).\end{aligned} \tag{62}$$

where the components in (62) can we approximated in a similar manner as in (28), so that:

$$\begin{aligned} u_k(\xi,\boldsymbol{\mu}) &= \boldsymbol{\phi}(\xi)\hat{\mathbf{u}}_k(\boldsymbol{\mu}); & N_s(\xi,\boldsymbol{\mu}) &= \boldsymbol{\phi}(\xi)\hat{\mathbf{N}}_s(\boldsymbol{\mu}); \\ F_k(\xi,\boldsymbol{\mu}) &= k\, u_k(\xi,\boldsymbol{\mu}); & \varepsilon_s(\xi,\boldsymbol{\mu}) &= \frac{1}{\beta}N_s(\xi,\boldsymbol{\mu}); \\ \varepsilon_k(\xi,\boldsymbol{\mu}) &= \frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\hat{\mathbf{u}}_k(\boldsymbol{\mu}); & F_s(\xi,\boldsymbol{\mu}); &= \frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\hat{\mathbf{N}}_s(\boldsymbol{\mu}); \\ N_k(\xi,\boldsymbol{\mu}) &= \beta\, \varepsilon_k(\xi,\boldsymbol{\mu}); & u_s(\xi,\boldsymbol{\mu}) &= \frac{1}{k}\, F_s(\xi,\boldsymbol{\mu}). \end{aligned} \tag{63}$$

These approximations can be expressed both in terms of FEM or PGD, only needing to set $\hat{\mathbf{u}}_k$ and $\hat{\mathbf{N}}_s$ accordingly. In order to compute (60) we also need to compute the derivative $\frac{\partial\rho(\xi,\boldsymbol{\mu})}{\partial\chi}$ and, therefore, we need the derivatives of (62). We will have different expressions for derivatives depending on what type of parameter χ is representing.

When χ represents the physical domain parameter x, and assuming that the material parameters k and β are defined such that $k = \mu_1$ and $\beta = 1$ if $x \leq \mu_4$, and $k = \mu_2$ and $\beta = \mu_3$ if $x > \mu_4$ we have:

$$\begin{aligned} \frac{\partial u_k(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{J}\frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\hat{\mathbf{u}}_k(\boldsymbol{\xi}); & \frac{\partial N_s(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{J}\frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\hat{\mathbf{N}}_s(\boldsymbol{\mu}); \\ \frac{\partial F_k(\xi,\boldsymbol{\mu})}{\partial x} &= k\,\frac{1}{J}\frac{\partial u_k(\xi,\boldsymbol{\mu})}{\partial\xi}; & \frac{\partial \varepsilon_s(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{\beta}\,\frac{1}{J}\frac{\partial N_s(\xi,\boldsymbol{\mu})}{\partial\xi}; \\ \frac{\partial \varepsilon_k(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{J}\left(\frac{\mathrm{d}^2\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi^2}\right)\frac{\partial\xi}{\partial x}\hat{\mathbf{u}}_k(\boldsymbol{\mu}); & \frac{\partial F_s(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{J}\left(\frac{\mathrm{d}^2\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi^2}\right)\frac{\partial\xi}{\partial x}\hat{\mathbf{N}}_s(\boldsymbol{\mu}); \\ \frac{\partial N_k(\xi,\boldsymbol{\mu})}{\partial x} &= \beta\,\frac{1}{J}\frac{\partial \varepsilon_k(\xi,\boldsymbol{\mu})}{\partial\xi}; & \frac{\partial u_s(\xi,\boldsymbol{\mu})}{\partial x} &= \frac{1}{k}\,\frac{1}{J}\frac{\partial F_s(\xi,\boldsymbol{\mu})}{\partial x}. \end{aligned} \tag{64}$$

And when χ represents one of the parameters μ_i, with $i = 1, 2, ..., \mathcal{D}$, we have:

$$\begin{aligned}
\frac{\partial u_k(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \boldsymbol{\phi}(\xi)\frac{\partial \hat{\mathbf{u}}_k(\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial N_s(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \boldsymbol{\phi}(\xi)\frac{\partial \hat{\mathbf{N}}_s(\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial F_k(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \frac{\mathrm{d}k}{\mathrm{d}\mu_i}u_k(\xi,\boldsymbol{\mu}) + k\frac{\partial u_k(\xi,\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial \varepsilon_s(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \frac{\mathrm{d}\beta^{-1}}{\mathrm{d}\mu_i}N_s(\xi,\boldsymbol{\mu}) + \frac{1}{\beta}\frac{\partial N_s(\xi,\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial \varepsilon_k(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \left(\frac{\mathrm{d}}{\mathrm{d}\mu_i}\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)\frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\hat{\mathbf{u}}_k(\boldsymbol{\mu}) + \frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\partial \hat{\mathbf{u}}_k(\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial F_s(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \left(\frac{\mathrm{d}}{\mathrm{d}\mu_i}\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)\frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\hat{\mathbf{N}}_s(\boldsymbol{\mu}) + \frac{\mathrm{d}\boldsymbol{\phi}(\xi)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\partial \hat{\mathbf{N}}_s(\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial N_k(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \frac{\mathrm{d}\beta}{\mathrm{d}\mu_i}\varepsilon_k(\xi,\boldsymbol{\mu}) + \beta\frac{\partial \varepsilon_k(\xi,\boldsymbol{\mu})}{\partial \mu_i};\\
\frac{\partial u_s(\xi,\boldsymbol{\mu})}{\partial \mu_i} &= \frac{\mathrm{d}k^{-1}}{\mathrm{d}\mu_i}F_s(\xi,\boldsymbol{\mu}) + \frac{1}{k}\frac{\partial F_s(\xi,\boldsymbol{\mu})}{\partial \mu_i}.
\end{aligned} \tag{65}$$

Notice that $\left(\frac{\mathrm{d}}{\mathrm{d}\mu_i}\frac{\mathrm{d}\xi}{\mathrm{d}x}\right) = 0$ unless μ_i is the geometric parameter $\mu_4 = \gamma$. For the derivatives of the PGD approximations, we have:

$$\begin{aligned}
\frac{\partial \hat{\mathbf{u}}_k(\boldsymbol{\mu})}{\partial \mu_i} &= \sum_{m=1}^{\mathcal{N}} \bar{\mathbf{u}}^m \,\frac{\mathrm{d}\,U_i^m(\mu_i)}{\mathrm{d}\mu_i}\prod_{j=1;j\neq i}^{\mathcal{D}} U_j^m(\mu_j)\\
\frac{\partial \hat{\mathbf{N}}_s(\boldsymbol{\mu})}{\partial \mu_i} &= \sum_{m=1}^{\mathcal{N}} \bar{\mathbf{N}}^m \,\frac{\mathrm{d}\,\tilde{N}_i^m(\mu_i)}{\mathrm{d}\mu_i}\prod_{j=1;j\neq i}^{\mathcal{D}} \tilde{N}_j^m(\mu_j)
\end{aligned} \tag{66}$$

We can now apply Equations (63) and (65) into Equation (60) to obtain the error indicator ι. Notice that by multiplying the derivative by the variable, we are able to keep ι with units that are consistent with the energy of the error ϵ^2.

p- Adaptivity

This section studies the results for p- refinement when using the proposed adaptivity indicator. It works with a simple verification, by identifying the parameter that has the highest value for ι among all the parameters studied and increasing the polynomial degree of that parameter by one, as this parameter is expected to influence the solution error the most.

The adaptivity process using the indicator proposed is compared with the uniform mesh adaptivity and is shown in Figure 8. Using the adaptivity indicator leads to better results than simply uniformly increasing the degrees of approximation functions, as it chooses the parameter that has the major influence in the solution to improve its degree. The drawback of this process is the need to repeat the simulation for every new degree improved.

One way to determine if the method to drive the adaptivity process is reliable is to determine all the possible solutions for a given set of degrees for the polynomial approximation and verify if the method is capable to accurately define which parameter should have its degree improved.

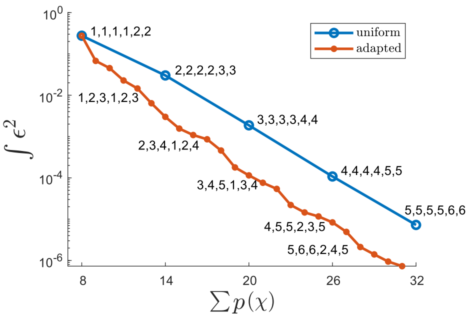


Figure 8: Comparison between the uniform mesh refinement and the adaptivity method for the integral of the energy of the error as a function of the combination of polynomial degrees. The detail numbers are the the degrees of the approximation in each parameter, respectively k_1, k_2, β_1, γ_1, b_1, b_2.

The following computation process was done:

1 Start with linear functions for the parameters and quadratic for the physical space;

2 Increase the polynomial degree of the parameter by one and compute the error;

3 Return the parameter to the original degree and repeat the previous step for the next parameter, until all the variables are studied;

4 Pick the parameter that leads to the smallest error and permanently increase the polynomial degree of that parameter by one;

5 Go back to step two until the sum of the degrees of the parameter reaches a predetermined value.

The results for different tolerances (τ_{pgd}) were compared with a reference solution (τ_{ref}) which was obtained by testing all possible solutions and choosing the optimal result, using a tolerance of 1×10^{-12}. The results can be seen in Figure 9, where the error tends to stagnate after a certain tolerance is achieved.

The adaptivity process is capable of precisely identify the parameter that causes the most impact in the solution, achieving better results for the error sooner than increasing all the degrees at once. We consider that the adaptivity method works if it is capable of selecting the parameter that required its polynomial degree increased to achieve the smallest error. Therefore, as long as the tolerance is small enough, the adaptivity method proposed is capable of choosing the optimal parameter to be refined.

h- Adaptivity

We now study the simulations results for h- refinement when using the proposed adaptivity indicator. Again, just like for the p- refinement, we use the parameter that gives the highest value ι as the indicator mechanism for the adaptivity process.

One key difference for the h- refinement is that it is also necessary to know where the element should be divided. The most natural place to choose is the middle of

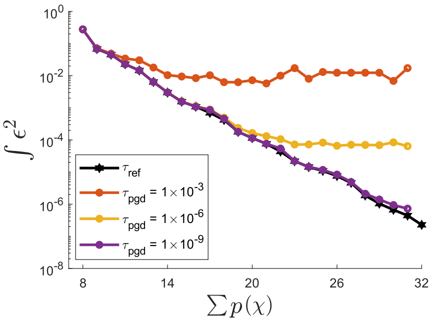


Figure 9: Comparison between the reference simulation τ_{ref} and the results with different tolerances for the integral of the energy of the error as a function of the combination of polynomial degrees.

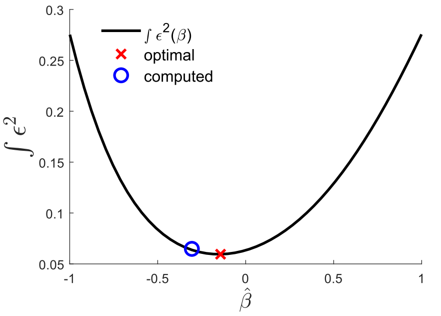


Figure 10: Energy of the error for variable β considering different element break positions. Details for the optimal break point and the computed using the gravity center of the energy of the error curve for β.

the element, but we can instead use the center of gravity of the energy error for that parameter, which is already being computed for the adaptivity indicator. To confirm that the center of gravity of the parameter is the best place to break the element, a simulation with 50 different break points in the interval $]-1,1[$ of the variable β was performed and is presented in Figure 10. It is possible to observe that the point that leads to the real minimum value is really close to the center of gravity of the parameter. This behavior is seen in all others parameters, for different numbers of elements or degrees, leading us to believe that the center of gravity provides a good indication on where to divide the element.

The adaptivity process using the proposed indicator is compared with the uniform mesh adaptivity and is shown in Figure 11. Using the adaptivity indicator leads to better results than uniformly increasing the number of elements, as it chooses the variable that has the major influence in the solution to divides its elements. Just like for the p- refinement, the drawback of this process is the need to repeat the simulation for every new element division.

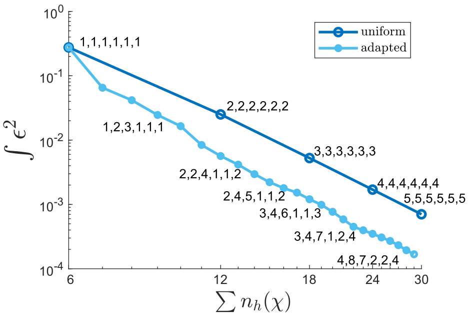


Figure 11: Comparison between the uniform mesh refinement and the adaptivity method for the integral of the energy of the error as a function of the element partitions. The detail numbers are the number of elements in each parameter, respectively k_1, k_2, β_1, γ_1, b_1, b_2.

We repeat the verification process performed for the p- refinement, determining all possible solutions to the problem for a given sets of elements and verifying if the method is able to accurately define which parameter should have its element divided.

The following computation process was performed:

1 Start with all parameter with single element;

2 Divide the element of the parameter and compute the total error;

3 Return the parameter to the original number of elements and repeat the previous step for the next parameter, until all the variables were studied;

4 Pick the parameter that results in the smallest error and permanent split the element of that parameter;

5 Go back to step two until the simulation reaches a predetermined value.

The results for different tolerances (τ_{pgd}) were compared with a reference solution (τ_{ref}) which was obtained by testing all possible solutions and choosing the optimal result, using a tolerance of 1×10^{-9}. These results are presented in Figure 12, with the total error stagnating after a certain tolerance is achieved.

The adaptivity process is capable of precisely identify the parameter that causes the most impact in the solution, achieving better results for the total error sooner than increasing all the elements at once. We consider the adaptivity method good if it is capable of selecting the parameter that needed its element split to achieve the smallest error. Therefore, just like for the p- refinement, as long as the tolerance is small enough, the adaptivity method proposed is capable of choosing the optimal parameter to be refined.

Conclusion

This paper defines new strategies to assess the error of a PGD parametric problem. We give a brief explanation on how the dual analysis of error works, defining errors measures in terms of the energy of the error and the error in energy. We proceed by introducing finite element and PGD approximations, focused for the solution

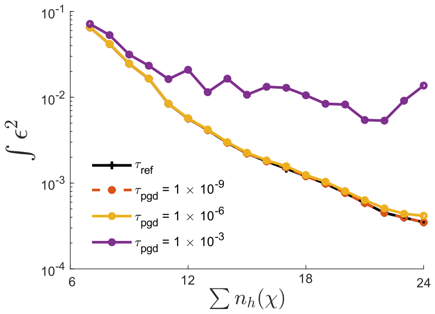


Figure 12: Comparison between the reference simulation τ_{ref} and the results with different tolerances for the integral of the energy of the error as a function of the element partitions.

of a 1D linear elastic problem, considering the details for the implementation of a geometric parameter, specific the length of the sections of the bar we are studying.

We present several examples to evaluate the behavior the PGD approximations. We compute the integrals of the energy of the error and the potential energies in the parametric domain for both PGD and FEM and compare the results with the exact solutions. The integral of the energy of the error presented itself as a good error measure, as it bounds the solutions without any additional conditions, while the error of the energies requires force or displacement driven problems to obtain bounds of the solutions.

We were able to develop an adaptivity indicator, based on empiric data obtained from several simulations performed. This adaptivity indicator is capable of capturing the optimal parameter (or space region) to be refined, both in terms of p- and h-refinements, leading to lower error values than those obtain with a simple uniform refinement.

Additional work is being done to extend the results obtained to a 2D and 3D framework [14]. We can also extend the results to obtain quantities of interest and bounds of their errors from for the PGD approximations [15], giving a direct physical meaning to the solutions obtained.

Competing interests
The authors declare that they have no competing interests.

Author's contributions
All authors discussed the content of the article, based on their expertise on the subjects presented. JR and JP prepare and run the numerical examples. All authors read and approved the final manuscript.

Acknowledgements
Jonatha Reis was supported by the European Education, Audiovisual and Culture Executive Agency (EACEA) under the Erasmus Mundus Joint Doctorate "Simulation in Engineering and Entrepreneurship Development (SEED)", FPA 2013-0043.
Pedro Díez and Sergio Zlotink are grateful for the financial support provided by the Spanish Ministry of Economy and Competitiveness (Grant agreement No. DPI2017-85139-C2-2-R), the Generalitat de Catalunya (Grant agreement No. 2017-SGR-1278), and the project H2020-RISE MATH-ROCKS GA nº 777778.

Author details
[1]Departamento de Engenharia Civil, Instituto Superior Técnico, Av. Rovisco Pais, 1049-001, Lisbon, Portugal.
[2]Laboratori de Calcul Numeric (LaCaN), E.T.S. de Ingenieros de Caminos, Canales y Puertos, Universitat

Politecnica de Catalunya, Jordi Girona 1, 08034, 24105 Barcelona, Spain. [3]International Centre for Numerical Methods in Engineering,CIMNE, Carrer del Gran Capità, S/N, 08034, 24105 Barcelona, Spain.

References

1. Ammar, A.: The proper generalized decomposition: A powerful tool for model reduction. International Journal of Material Forming **3**, 89–102 (2010)
2. Ladevèze, P., Chamoin, L.: On the verification of model reduction methods based on the proper generalized decomposition. Computer Methods in Applied Mechanics and Engineering **200**, 2032–2047 (2011)
3. Ammar, A., Chinesta, F., Diez, P., Huerta, A.: An error estimator for separated representations of highly multidimensional models. Computer Methods in Applied Mechanics and Engineering **199**, 1872–1880 (2010)
4. Chinesta, F., Keunings, R., Leygue, A.: The Proper Generalized Decomposition for Advanced Numerical Simulations: A Primer. Springer, Cham (2014)
5. Zlotnik, S., Díez, P., Gonzalez, D., Cueto, E., Huerta, A.: Effect of the separated approximation of input data in the accuracy of the resulting PGD solution. Advanced Modeling and Simulation in Engineering Sciences **2**, 28 (2015)
6. Ammar, A., Huerta, A., Chinesta, F., Cueto, E., Leygue, A.: Parametric solutions involving geometry: A step towards efficient shape optimization. Computer Methods in Applied Mechanics and Engineering **268**, 178–193 (2014)
7. Courard, A., Néron, D., Ladevèze, P., Ballere, L.: Integration of PGD-virtual charts into an engineering design process. Computational Mechanics **57**, 637–651 (2016)
8. de Almeida, J.P.M.: A basis for bounding the errors of proper generalised decomposition solutions in solid mechanics. International Journal for Numerical Methods in Engineering **94**, 961–984 (2013)
9. Almeida, J.P.M., Maunder, E.A.W.: Equilibrium Finite Element Formulations, pp. 1–274. John Wiley & Sons, Ltd, Chichester, UK (2017)
10. de Almeida, J.P.M.: A basis for bounding the errors of proper generalised decomposition solutions in solid mechanics. International Journal for Numerical Methods in Engineering **94**, 961–984 (2013)
11. Prager, W., Synge, J.L.: Approximations in elasticity based on the concept of function space. Quarterly of Applied Mathematics **5**, 241–269 (1947)
12. Modesto, D., Zlotnik, S., Huerta, A.: Proper generalized decomposition for parameterized Helmholtz problems in heterogeneous and unbounded domains: Application to harbor agitation. Computer Methods in Applied Mechanics and Engineering **295**, 127–149 (2015)
13. Chamoin, L., Pled, F., Allier, P.-E., Ladevèze, P.: A posteriori error estimation and adaptive strategy for PGD model reduction applied to parametrized linear parabolic problems. Computer Methods in Applied Mechanics and Engineering **327**, 118–146 (2017)
14. Reis, J., Moitinho de Almeida, J.P., Díez, P., Zlotnik, S.: Error estimation for proper generalized decomposition solutions: A dual approach. International Journal for Numerical Methods in Engineering **n/a**(n/a). doi:10.1002/nme.6452
15. Reis, J., Almeida, J.P.M., Díez, P., Zlotnick, S.: Error estimation for pgd solutions: Dual analysis and adaptivity for quantities of interest (2020). Submitted for publication