UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Mecânica

LUÍS FELIPE DE OLIVEIRA LIMA

Applications of the Bidirectional Evolutionary Structural Optimization using the Element-free Galerkin

Aplicações da Otimização Estrutural Evolucional Bidirecional utilizando o método Galerkin sem malha

CAMPINAS
2021

LUÍS FELIPE DE OLIVEIRA LIMA

Applications of the Bidirectional Evolutionary Structural Optimization using the Element-free Galerkin

Aplicações da Otimização Estrutural Evolucional Bidirecional utilizando o método Galerkin sem malha

Master's thesis presented to the School of Mechanical Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Mechanical Engineering.

Dissertação de Mestrado à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Mecânica, na Área de Mecânica do Sólidos e Projeto Mecânico.

Orientador: Prof. Dr. Renato Pavanello

ESTE EXEMPLAR CORRESPONDE À VERSÃO DO TEXTO DA QUALIFICAÇÃO APRESENTADA PELO ALUNO LUÍS FELIPE DE OLIVEIRA LIMA E ORIENTADO PELO PROF. DR. RENATO PAVANELLO.

CAMPINAS
2021

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

Lima, Luís Felipe de Oliveira, 1993-
L628a Applications of the Bidirectional Evolutionary Structural Optimization using the Element-free Galerkin / Luís Felipe de Oliveira Lima. – Campinas, SP : [s.n.], 2021.

Orientador: Renato Pavanello.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Otimização topológica. 2. Método dos elementos finitos. 3. Métodos de Galerkin. I. Pavanello, Renato, 1959-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Aplicações da Otimização Estrutural Evolucional Bidirecional utilizando o método Galerkin sem malha
Palavras-chave em inglês:
Topology optimization
Finite element method
Galerkin methods
Área de concentração: Mecânica dos Sólidos e Projeto Mecânico
Titulação: Mestre em Engenharia Mecânica
Banca examinadora:
Renato Pavanello [Orientador]
Eduardo Lenz Cardoso
Euclides de Mesquita Neto
Data de defesa: 29-06-2021
Programa de Pós-Graduação: Engenharia Mecânica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0001-7544-0447
- Currículo Lattes do autor: http://lattes.cnpq.br/2011541692022012

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

DISSERTAÇÃO DE MESTRADO ACADÊMICO

Applications of the Bidirectional Evolutionary Structural Optimization using the Element-free Galerkin

Aplicações da Otimização Estrutural Evolucional Bidirecional utilizando o método Galerkin sem malha

Autor: Luís Felipe de Oliveira Lima
Orientador: Renato Pavanello

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:

Prof. Dr. Renato Pavanello, Presidente
DMC / FEM / UNICAMP

Prof. Dr. Eduardo Lenz Cardoso
CCT / UDESC

Prof. Dr. Euclides Mesquita Neto
DMC / FEM / UNICAMP

Campinas, 29 de Junho de 2021.

Dedicatória

Aos meus pais, que sempre acreditaram no estudo como forma de vencer na vida.

Acknowledgments

To my family, Márcia e Francisco for the opportunity of studying. I'm very thankful to my dear aunt Lindinalva de Lima Franchini for being to me an example of integrity and strength and for presenting me the world.

A very special thanks to professor Dr. Renato Pavanello for all the precious opportunities and the support along my under-graduation and for making possible this master's thesis.

To my friends from LTM. In special to Evandro Carobino Souto and Daniel Candeloro Cunha, for the endless conversations.

To my colleagues from the Brazilian National Synchrotron Laboratory: Débora, Gustavo, Marcos e Laís.

To Lucas Leite Vasconcellos, Luiz Paulo Rodrigues Filho and Vinícius Eduardo de Almeida Malta friends from the university to life.

To my friend Jérémie Bry for the corrections and for the incredible moments during my studies in France.

To my beloved Nathália.

O papel aceita tudo.

F. C. de Lima

Resumo

DE OLIVEIRA LIMA, Luís Felipe. Aplicações da Otimização Estrutural Evolucional Bidirecional utilizando o método Galerkin sem malha. 2021. 121p. Dissertação (Mestrado). Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

O objetivo principal desse trabalho de mestrado é implementar uma ferramenta computacional para otimização topológica estrutural utilizando métodos sem malha para resolver o problema físico. O método de otimização topológica empregado é o *Bi-directional Evolutionary Structural Optimization* – BESO –, enquanto o método sem malha utilizado é o *Element-free Galerkin* – EFG. Duas características particulares desse método são a vanidade de malhas estruturadas e sua capacidade de lidar com problemas não lineares. As funções de forma do método EFG foram implementadas utilizando dois métodos distintos: o *Moving Least-Squares* e o *Radial Point Interpolation*. No campo da otimização topológica, esse trabalho foca em minimizar a flexibilidade média em problemas elásticos bidimensionais. As formulações em condição linear e para o caso de não linearidades geométricas são desenvolvidas para o problema de elasticidade que é, em seguida, discretizado utilizando o Método dos Elementos Finitos – FEM – e o EFG, no intuito de compará-los. A formulação do BESO é então apresentada, levando em conta as especificidades do FEM e do EFG. Em um primeiro momento, as implementações dos métodos FEM e EFG são validadas utilizando soluções analíticas para problemas simples e pacotes numéricos comerciais para problemas mais complexos. O método BESO-EFG implementado é comparado ao consolidado BESO-FEM através diferentes problemas de referência em elasticidade linear: as estruturas *two-bar* e *Michel-type* e as vigas engastada e *MBB*. Por fim, buscando explorar a capacidade do EFG de lidar com não linearidades geométricas, o método BESO-EFG implementado é usado na otimização topológica de uma viga engastada não linear.

Palavras-chave: Otimização Topológica, BESO, Método dos Elementos Finitos, *Element-free Galerkin*, Não Linearidades Geométricas.

Abstract

DE OLIVEIRA LIMA, Luís Felipe. Applications of the Bidirectional Evolutionary Structural Optimization using the Element-free Galerkin. 2021. 121p. Thesis (Master). School of Mechanical Engineering, University of Campinas, Campinas, Brazil.

The main objective of this master thesis is to implement a computational tool for structural topology optimization that uses meshless methods to solve the physical problem. In this work, the chosen topology optimization method is the Bi-directional Evolutionary Structural Optimization – BESO –, while the meshless method utilized is the Element-free Galerkin – EFG. Two characteristics of the EFG method are that no structured meshes are needed and its capacity to deal with nonlinear problems. The EFG shape functions are constructed using two different methods: the Moving Least-Squares and the Radial Point Interpolation Method. In the topology optimization field, this work is concerned with minimizing the mean compliance in two-dimensional elastic structures. Both linear and geometrically nonlinear elasticity formulations are developed and, subsequently, discretized using the Finite Element Method – FEM – and the EFG, in the interest of comparing these methods. Then, the BESO formulation is presented considering both FEM and EFG solvers. In a first moment, the EFG and FEM implementation are validated through comparisons with analytical solutions and commercial solvers. The implemented BESO-EFG algorithm is compared with the well-established BESO-FEM method using several linear elastic benchmark problems: two-bar and Michel-type structures, cantilever beam, MBB beam. Finally, seeking to explore the capability of the EFG method to deal with nonlinear structures, the implemented BESO-EFG is used in the topology optimization of a geometrically nonlinear cantilever beam.

Keywords: Topology Optimization, BESO, Finite Element Method, Element-free Galerkin, Geometrically Nonlinear Structures.

List of Figures

List of Tables

Table of Contents

1 Introduction

1.1 Background

In a world with a growing trend to sustainability and to cost reduction, intelligent design of structures plays a key role to achieve eco-friendly economy. Intelligent products or materials can be obtained, for example, through topology optimization methods, where material is added or removed following specifics performance criteria in a given project domain. For instance, consider a beam with length l and a rectangular cross-section with dimensions $b \times h$, projected to be a stiff structure, as illustrate in Figure 1.1a. The first design would be a completely filled beam, which meets the project requirements for stiffness. It is, however, interesting to use the least amount of material to manufacture this beam due to, for example, economics reasons or environmental impact reduction. A question which naturally arises is: *where is the most appropriate location to remove material, so that the stiffness properties reduces the least?* A key for this sort of question is topology optimization, which gives the material distribution over the design domain that minimizes or maximizes a performance criterion. To illustrate this idea, Figure 1.1 shows two final designs, among several possibilities, after material removal. Topology optimization methods help to decide which design 1.1b or 1.1c is the best, given a performance criterion.

In general, topology optimization involves two fields: the physical problem and the optimization problem. Typically, an optimization algorithm uses the results from equations that govern the physical behavior of a system to decide which is the best material distribution of this system, as illustrated in Figure 1.2. By solving the physical problem, one gets information on the system behavior and then evaluates the performance criterion or the objective function. The optimization strategies use the performance criteria to decide where to remove or to add material, providing an optimized system layout.

Often, the physical models are a set of partial differential equations – PDE – which, for the most part, can only be solved using numerical methods. In addition to consistent PDEs to model the physical problem, we also need precise and robust numerical methods for solving these equations, which is a motivation for the great effort in the scientific community to develop efficient numerical methods. Indeed, in order to assure appropriate topology results, it is important to have reliable numerical methods to approximate the physical system behavior. The widely known Finite

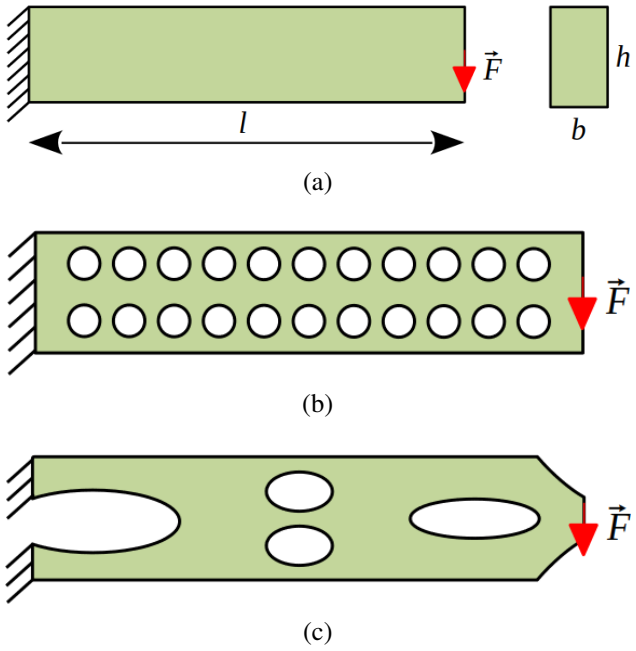


Figure 1.1: A completely filled cantilever beam with rectangular cross-section is presented in (a). Two alternatives for material removal are presented in (b) and (c), considering a same amount of removed matter. Given an objective function, optimization methods can predict whether (b) or (c) is the best design.

Element Method – FEM – have been employed with success to solve physical problems in topology optimization. However, there are other methods that could be used alternatively to FEM, which is the case of meshless methods, such as the Element-free Garlekin method – EFG – used in this work. To deal with the optimization problem, different methods can be used to achieve optimal designs, such as density-based methods, boundary variation methods or discrete methods, which are reviewed more in detail later in this chapter.

Because no structured mesh is required in meshless methods, they can present advantages over FEM when dealing with complex geometries, e.g., human bones or cartilages. Typically, meshing the complex domains like these can be a time expensive task. Indeed, correctly meshing the solution domain can be time consuming not only for FEM but also for the methods requiring structured or regular meshes, such as the Finite Difference Method – FDM – or the Finite Volume Method – FVM. Meshless methods can also be expedient when the structure being modeled needs remeshing during the solution process or when the mesh elements undergoes great distortions, leading to numerical precision loss.

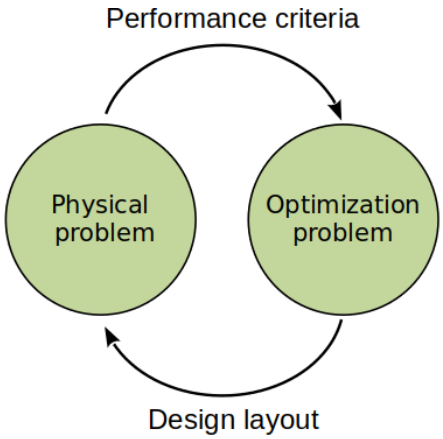


Figure 1.2: Relationship between physical and optimization problems, established through the structure design and its relative performance criterion.

Meshless methods like the Element-free Galerkin are based on high order shape functions, while several FEM codes uses the bilinear shape functions of the 4 nodes quadrilateral elements, called hereafter *quad4* elements. This can lead to smoother stress results in the case of EFG at the cost of higher computational time (MOLLON, 2016), (OVERVELDE, 2012), (BELYTSCHKO ET AL., 1994). Indeed, the EFG method provides continuous stress field, which is not the case of FEM with *quad4* elements. Finally, the EFG is presented as a more robust method under large deformation then the traditional FEM, being an interesting alternative for this class of problems. Indeed, the EFG uses no discretization mesh, therefore it avoids dealing with highly distorted elements; in addition, the EFG provides a smoother representation of the stress field compared to low order FEM (MOLLON, 2016) and (BELYTSCHKO ET AL., 1994).

The FEM has been historically employed to solve a large range of engineering problems such as solid and fluid mechanics, heat transfer, acoustics and electromagnetism. The EFG is a more recent and, so far, a less developed method then the FEM. Nonetheless, it has been successfully used to solve these engineering problems, being particularly promising in crack propagation (SALARI AND DIZADJI, 2012), contact and geometrically nonlinear problems (MOLLON, 2016). Of course, EFG and the whole class of mesh-free methods have their limitations and difficulties, for instance the imposition of essential boundary conditions and the high computational cost of to evaluate the shape functions. However, the EFG is still attractive, presenting interesting benefits over FEM which lead us to use it as a solver in topology optimization methods.

In the field of topology optimization, several methods have been proposed to solve a

wide range of optimization problems. Methods such as Level-Set, SIMP, Homogenization and BESO have been used to obtain optimized design of, for example, stiff structures (VICENTE ET AL., 2015), systems with fluid-structure interaction, piezoelectric harvesters (DE ALMEIDA ET AL., 2019) and fluid-cellular actuators (CUNHA D.C, 2018). Among the vast methods for topology optimization, the BESO, acronym for Bi-directional Evolutionary Structural Optimization, will be employed in this work. When compared to its concurrent methods, the BESO presents precise results and a simpler and more intuitive implementation. It has proven to be an efficient method when dealing with stiffness and frequency optimization, as well as solving problems with multiple material, periodic structures, material nonlinearities and large formations. A special strength of the BESO method is its capability to naturally deal with fluid-structure optimization problems in multiphysic systems.

1.2 Objectives

The main purpose of this work is to develop a topology optimization algorithm that uses meshless methods as numerical method for solving solid mechanics problems. The topology optimization strategy selected to this work is the Bi-directional Evolutionary Topology Optimization – BESO. The meshless method adopted is the Element-free Galerkin – EFG – due to its popularity and relatively easy implementation. The EFG will be implemented in two versions: in its classical form, which uses a Moving Least-Squares method to construct the shape functions, and in a slightly modified version, that uses a Radial Point Interpolation Method to construct the shape functions, incorporating the Kronecker delta property (LIU, 2003).

Although the main objective of this work is to perform topology optimization using the EFG method, a secondary objective is to also implement a FEM solver, for three main reasons. First, to gain a preliminary insight on the BESO method, which was historically implemented using the FEM solver for the physic problem. Second, it is intended to perform nonlinear structural analysis and, as a substantial portion of the scientific literature for this class of problems uses the FEM formulation, a clear understanding of this method is desirable. Third, comparisons between the FEM and EFG results are envisaged.

The results of BESO-FEM and BESO-EFG will be compared using topology optimization benchmark problems for 2D linear elasticity, such as two-bar, Michel-type structures, cantilever

beam and MBB beam. As one of the major advantages of the EFG is it capability to deal with large deformations, the last objective is to apply the implemented BESO-EFG method to the topology optimization of a nonlinear cantilever beam.

The main objectives of this project are given in Figure 1.3, organized in two main branches: objectives related to the FEM implementation and those concerning the EFG method. In a chronological order, the first objective is the FEM implementation once there is a vast literature about this method. Then, the shape functions to be used in the EFG method are implemented and studied before implementing the EFG itself. Actually, these shape functions have a crucial role in the EFG method and are studied in detail before tackling the implementation of this method. Once the discretization methods are well established, the topology problem can be tackled. We start with the implementation of the BESO-FEM followed by the BESO-EFG. These both methods are used in the topology optimization of linear problems. Next, we apply the BESO-EFG in the topology optimization of geometrically nonlinear structures.

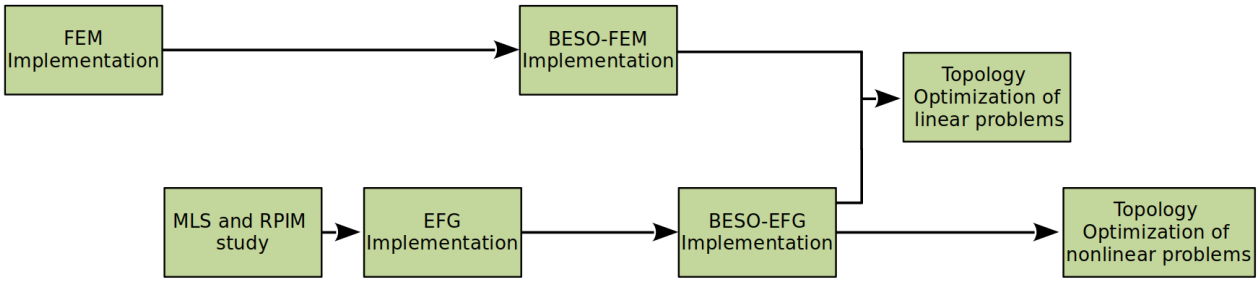


Figure 1.3: Major objectives of this project, organized using two branches: the first is relative to them FEM implementation and the other is related to the EFG. The black arrows indicate the chronological order of the objectives.

1.3 Scientific literature review

When working with topology optimization two different, yet related, problems show-up: the physical problem and the topology optimization problem, as illustrated in 1.4. To solve the physical problem governing equations, numerical methods are used to approximate the PDE by a set of algebraic equations. Traditionally, the Finite Element Method, the Finite Differences Method and the Finite Volume methods are used as numerical numerical methods to solve the governing PDE of the physical problem. Recently, meshless methods such as Diffuse Element Methods, Element-free Garlekin method, Smooth Particle Hydrodynamics Methods are also employed to solve the

physical problem equations.

It is hence useful to divide the literature review in three groups: first, on the numerical methods used to solve the physical problem, with an emphasis on EFG and FE methods; second, on topology optimization – material model and optimization solvers–; and finally, a specific review on topology optimization works using meshless methods as physical solvers.

1.3.1 Meshless methods and Finite Element Method

Meshless methods or, alternatively, mesh-free methods – MFM – were developed to prevent the difficulties associated with the meshing step in approximating PDEs (NGUYEN ET AL., 2008). The EFG method was first proposed by BELYTSCHKO et al. (1994) and an implementation using MATLAB is proposed in the works of (DOLBOW AND BELYTSCHKO, 1998) and (OVERVELDE, 2012).

Contemporary, due to the improvement in computational capacities, the Computer Aided Engineering – CAE – is a widely used tool in the design of industrial systems and in academic researches. Through this tool, complex multiphysic systems can be modeled and explored. These systems are governed by differential or partial differential equations which are, traditionally, solved using numerical methods such as the FEM, FDM or FVM (LIU, 2003).

The FEM, FDM and FVM fundamentally depend on structured meshes. In FEM the mesh constituted by elements, in FDM the mesh is called grid, and in FVM the mesh is composed by cells or volumes. For any case, the formulations of these methods are based on nodal discretization that uses a predefined relationship between the nodes. This relationship is what defines an element, a grid or a cell in the FEM, FDM or FVM, respectively.

More recently, different methods to solve the PDEs have been developed as an attempt to overcome the limitations of the mesh-based methods. Among these new methods, the so called MFM are a promising alternative. This class of method approximates the field function based on a nodal discretization that do not require any connections between the nodes.

A pioneering MFM is the Smoothed Particle Hydrodynamics – SPH – method, introduced

independently by LUCY (1977) and GINGOLD and MONAGHAN (1977), which was firstly proposed to solve astrophysics problems. It was then adapted to fluid mechanics (GINGOLD AND MONAGHAN, 1982) and to solid mechanics (LIBERSKY AND PETSCHEK, 1991) and (LIBERSKY AND PETSCHEK, 1993). The SPH uses a strong formulation of the PDEs, which includes second order derivatives. The first MFM based on a weak formulation and on the Galerkin technique is the Diffuse Element Method developed by NAYROLES et al. (1992), in which the main idea is to replace the FEM local interpolants by the Moving Least Squares local interpolation.

An extended version of the DEM was developed by BELYTSCHKO et al. (1994) named Element-free Galerkin – EFG –, a widely used mesh-free method. Despite more precise than the DEM, the EFG method is computationally more expensive. The EFG is applied to a vast range of engineering problems, such as cracks propagation SHOBEIRI (2015), granular materials modeling MOLLON (2016), rigid and deformable bodies dynamics MOLLON (2018), thermo-mechanical simulations IBANES et al. (2013) and tunnel design (HAJIAZIZI AND BASTAN, 2014).

Some advantages of MFM are pointed out by NGUYEN et al. (2008): h-adaptivity is simpler to incorporate in MFMs than in mesh-based methods; problems with moving discontinuities such as cracks propagation, shear bands and phase transformation can be treated with ease; large deformation can be handled more robustly; high-order continuous shape functions; non-local interpolation character; and no mesh alignment sensitivity.

The FEM is not the main object of this work, instead, it plays the role of a tool to discover the solid mechanics and topology optimization and to validate and compare the results. The FEM implementation is totally based on (KWON AND BANG, 2000) and (KIM, 2015).

1.3.2 Topology optimization

The domain of structural optimization can be divided in three main branches: sizing, shape and topology optimization. In sizing optimization, solely parameters as the thickness, lengths or diameters are set as design variables, which makes the final optimized structure to present a similar shape to the initial design. In shape optimization, the position of structural boundary can be altered during the optimization processes. The boundaries can be controlled, for instance, using limit points and the boundary tangent at these points as design variables. In topology optimization, material

can be added or removed within the design domain, which allows the creation of holes and the merging of these holes anywhere in the design domain, leading to results wholly different from the structure's initial design. Figure 1.4 illustrates these three categories.

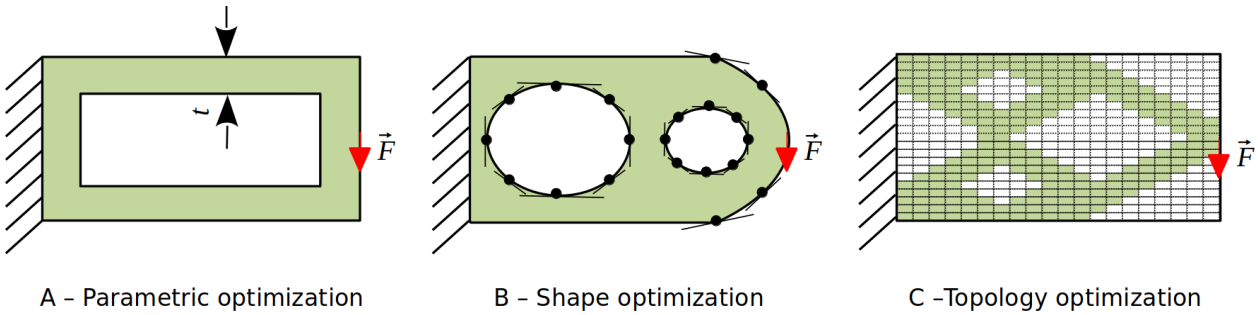


Figure 1.4: The three branches of structural optimization. In *A* the thickness t is the design variable; in *B* the position of black points and the tangent at these points are the design variables which will define the boundaries of the design domain; and *C* illustrates the void and solid elements using white and green rectangles, which defines the holes within the domain.

The topology is represented by a design variable vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ which assumes only values between 0 and 1, representing void or solid, respectively:

$$x_i = \begin{cases} 1 \; \textit{solid} \\ 0 \; \textit{void} \end{cases} \tag{1.1}$$

There are two types of topology optimization algorithms: the global methods, outside the scope of this work, and the gradient-based methods – object of this text – which relies on the objective function derivatives with relation to the design variables, often called sensitivity. Hereafter, we will always refer to topology optimization with gradients-base methods only as topology optimization, for simplicity sake.

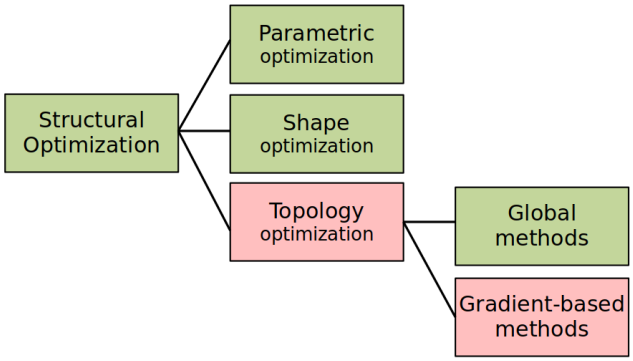


Figure 1.5: The three branches of structural optimization. In this work we are only concerned with topology optimization using gradient-based methods.

A topology optimization problem consists on finding the material disposition in a structure so that a specific performance criterion are maximized or minimized. The optimum structure is obtained by determining, for every point in the design domain, if there should be material or voids (HUANG AND XIE, 2010). In this sense, topology optimization is a more general class of structural optimization in which points within the design domain, representing the presence or absence of material, are used as design variables. To conveniently assign properties to these points and thus define solid or void, a material model needs to be conveniently defined. Once the material model is defined, an optimization strategy can be used to properly assign void or solid properties to a point in the domain aiming at the maximization or the minimization of a given performance criteria. The topology optimization field can roughly be categorized by material model and optimization scheme, leading up to three groups: density-based methods, boundary variation methods and discrete methods.

In density-based methods, the design variables are described in terms of material density over the design domain, imposing that the density is bounded between a minimum value ρ_{min} and 1. The material density at a point is given in terms of the design variables x_i. A crucial aspect of density-based methods is the choice of an interpolation function to conveniently express the density in a domain's point as a function of the design variables. For example, the Solid Isotropic Material with Penalization – SIMP – method proposed by BENDSOE (1989) uses a power law to describe the density in domain points. Alternatively to a power law, the Rational Approximation of Material Properties – RAMP – proposes the use of a rational function to describe the material behavior (STOLPE AND SVANBERG, 2001). In all cases, the density distribution over the domain can be treated as a function and an optimization method such as Method of Moving Asymptotes – MMA

– (SVANBERG, 1987). An extensive review on density method is presented by BENDSOE and SIGMUND (1999) discussing the advantage and drawbacks of each material model.

In level-set methods (WANG ET AL., 2003), a level-set function is used to describe the boundaries of solid or void material inside the design domain. The boundaries are controlled through the motion of the level-set function, which is subject to the physical problem and optimization condition. The main idea of using level-set methods is that topological merging or breaking are well defined and naturally performed (OSHER AND FEKIW, 2001). The level-set function is well defined, an optimizer such as MMA or OC can be used to find the optimal topology. Recently, the initial level-set method proposed by WANG et al. (2003) has been used with other techniques such topological derivatives (BURGER ET AL., 2004). A level-set method implementation is presented by ANDREASEN et al. (2020) where some overlaping similarities between level-set and density-based method are shown.

In discrete methods, the material model is usually taken from density-based methods, such as the SIMP, and the density of a point are allowed to assume rather 1 or ρ_{min}, no intermediary values are authorized. The design variable update scheme does not rely on optimization solvers; instead, the design variables are updated gradually along iterations until a prescribed volume is reached and a stop criteria is satisfied, that is why this type of method are also called evolutionary methods. The main representatives methods are the Evolutionary Structural Optimization – ESO (XIE AND STEVENS, 1993) and its more general version the Bidirectional ESO or BESO (HUANG AND XIE, 2010). These methods are also called heuristics because they depend on strong hypothesis on the behavior of the objective function (ZHOU AND ROZVANY, 2001).

Outside these three groups, the Homogenization method presented by BENDSOE and KIKUCHI (1988) which is one of the precursors of the modern topology optimization field. In this method, periodically small holes are inserted in a given homogeneous and isotropic material cells. Then, the properties of the resulting anisotropic material is obtained by means of the homogenization method. An optimization scheme is then used to determine the optimal distribution of small holes in each cell or element.

This work uses a BESO scheme for the design variables update where the main concept is to incrementally add or remove material along the iterations until the optimum design is obtained. For this class of topology optimization method, the material density assumes either zero or one values. The inefficient material to be eliminated is determined by a sensitivity analysis which uses the

derivative information of the objective function. The sensitivity of each point or material is often smoothed using a filter scheme, which helps to avoid undesirable numerical instabilities such as checkerboard patterns, mesh-dependency and convergence issues for more elaborated structures. The use of proper sensitivity filters and consistent SIMP model with the BESO procedure is a reliable optimization method, capable of dealing with complex structure with high computational efficiency HUANG and XIE (2010).

Although the density approaches – Homogenization, SIMP and Level-Set – have been privileged in topology optimization, the BESO is selected as the methods to be used in this work. The choice of BESO for this project is due to its simplicity in terms of formulation and implementation and due to its robust results in a wide range of problems, as showed in the works of LOPES et al. (2017) in natural frequency optimization, PICELLI (2015) in fluid-structure interaction and CUNHA (2019) in the conception of cellular fluid-actuators.

1.3.3 Topology optimization with meshless methods

The above presented topology optimization methods use the FEM to solve the physical problem equations. We will call BESO-FEM the algorithm that combines the BESO for topology optimization and the FEM for physical problems. Below, we present a review on the use of meshless methods employed for topology optimization problems.

The Reproducing Kernel Particles Method – RKPM – is employed with a SIMP method for topology optimization of geometrically nonlinear structures in CHO and KWAK (2006). The RKPM is also used with an implicit topology description approach in ZHOU and ZOU (2008). The EFG is used with a SIMP strategy for topology optimization of compliant actuators (DU ET AL., 2009), and with an Optimality Criteria – OC – strategy for topology optimization with displacement constraints in (YANG ET AL., 2017).

The EFG is adopted with the ESO method in (ZHENG ET AL., 2010) and with BESO in (ZHAO, 2014) and (SHOBEIRI, 2016), demonstrating that the use of meshless methods can effectively suppress the traditional numerical instabilities of BESO-FEM, such as mesh dependence, local minima and checkerboards. Indeed, the work presented by ZHAO (2014) is crucial for this master's thesis because it introduces the dual-level interpolation method, capable of penalizing

not only the discretization nodes but also the integration points in a BESO-EFG algorithm. In his master's thesis, OVERVELDE (2012) proposes a very interesting optimization method, called the Moving Nodes Approach – MNA – that uses the EFG.

Topology optimization methods have been used to obtain optimum design of structures undergoing large deflections, which is one of the objectives of this work. The topology optimization of geometrically nonlinear problems is approached by HUANG and XIE (2010) in their book, proposing a nonlinear FEM model with BESO strategy. Another study has been conducted by GOMES and SENNE (2014) using the Sequential Piecewise Linear Programming – SPLP – for topology optimization, providing several benchmark problems. In their work, BUHL et al. (2000) the MMA method is used to solve the topology optimization problem. Other relevant works on topology optimization of geometrically nonlinear structures are (KIKUCHI ET AL., 1998), (NISHIWAKI ET AL., 1998) and (ABDI ET AL., 2017). Concerning geometrically nonlinear structure and compliant mechanisms using meshless methods, there are the relevant works of (ZHENG ET AL., 2015) and (QIZHI ET AL., 2014) that use the EFG with the Optimality Criteria method, and (DU ET AL., 2009) that show interesting topology results for thermo-mechanical compliant mechanisms using the EFG with an MMA method.

2 Meshless Methods

We start this chapter by presenting a formulation for two-dimensional linear elasticity problems. Once the linear elasticity model is detailed, the FEM is presented in a straightforward manner. Next, the EFG method is presented along with its main characteristics: the absence of structured mesh and the shape functions. Due to the importance of shape functions to the EFG method, two subsections are reserved to study the two class of shape functions proposed in this work: the Moving Least-Squares and the Radial Point Interpolation Method. For both FEM and EFG it is presented how to obtain the nodal or local stiffness matrix and how to assembly the global matrix. Comparisons between FEM and EFG are point out whenever possible. Finally, the linear theory presented in the first section of this chapter is extended to account large deformations. The FEM and EFG discretization are also extended to incorporate the large deflections.

2.1 2D Linear elasticity

In continuum mechanics we are often interested in finding the displacement field of a body given the external load. As illustrated in Figure 2.1, the process to obtain the displacement field formulation as a function of the external load requires a few transformations. First, it is necessary a kinematic law that describes the body's motion to obtain the strain field. A relationship between strain and stress field is established through a material law obtained experimentally, called constitutive law. Finally, a force balance or energy minimization can be applied to obtain a relation between external loads and internal forces or the stress field. Below, there are four subsections dedicated to each of these steps, presenting in detail the assumptions and the procedures to derive the linear elastic model.

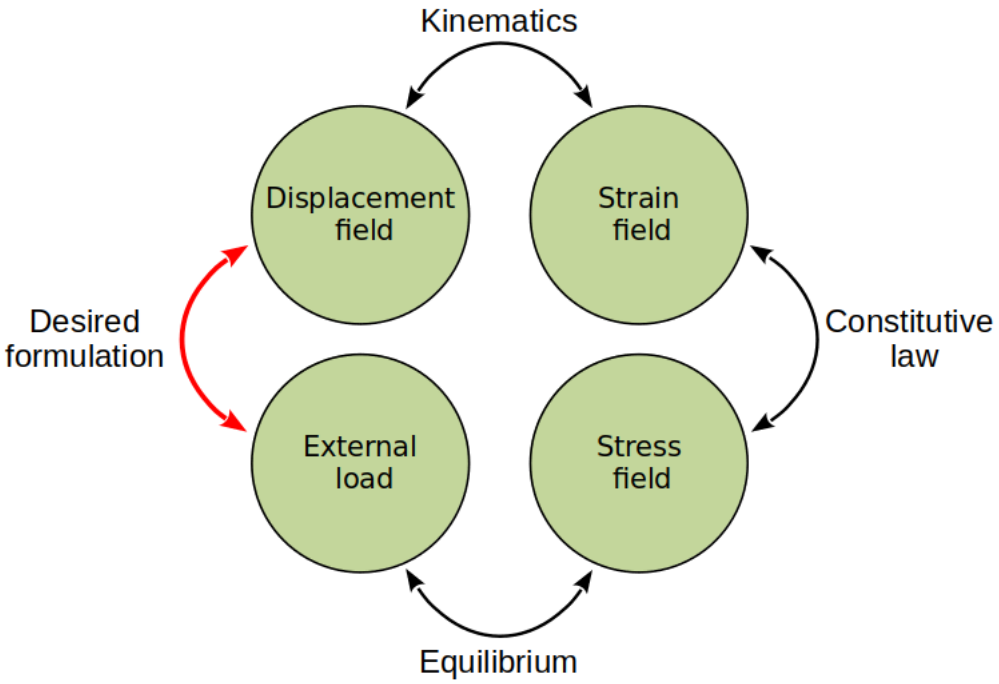


Figure 2.1: Steps for formulating a static structural problem, which relates the external loads with displacement field.

2.1.1 Notation

In this short subsection the notation adopted in this chapter is explained, in an attempt to make the its reading easier. We will denote the displacement field by $\mathbf{u}(\mathbf{x})$, where $\mathbf{x}$ are the coordinates of an arbitrary point. Using a vector notation, with subscripts 1 and 2 to indicate the 2D directions, we write:

$$\mathbf{u} = \begin{Bmatrix} u_1(\mathbf{x}) \\ u_2(\mathbf{x}) \end{Bmatrix},$$

and

$$\mathbf{x} = \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix}.$$

The partial derivatives of a scalar function f will be denoted by:

$$\frac{\partial f}{\partial x_j} = f_{,j}.$$

The gradient of a function will be widely used hereafter. We denote the gradient ∇ of a scalar function f that depends on the variables x_1,$x_2, \ldots, x_n$ as:

$$\boldsymbol{\nabla} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix}.$$

An important operator that is recurrent in elasticity problems is the linear differential operator, here denoted by $\mathbf{L}$:

$$\mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \end{bmatrix}.$$

The stress $\boldsymbol{\sigma}$ and strains $\boldsymbol{\epsilon}$ fields are written using the Voigt notation:

$$\boldsymbol{\epsilon} = \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ 2\epsilon_{12} \end{Bmatrix}, \quad \boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix}.$$

2.1.2 Kinematics

The kinematics of a deformable body is a mathematical description of the relative motion between two points within a body under external loads. The 2D linear case considers only small deformations, being a simplification of the more general nonlinear elasticity theory.

To obtain the kinematics of a deformable body with smooth boundaries, we use the diagram of Figure 2.2, where a body goes from an initial undeformed domain $\mathbb{S}_0$ to a final deformed domain $\mathbb{S}$. The deformation is denoted by a mapping function $\boldsymbol{\varphi}$, which transforms a coordinate $\mathbf{X}$ in the initial undeformed geometry to coordinates $\mathbf{x}$ in the final deformed shape. In the same manner, the inverse mapping function $\boldsymbol{\varphi}^{-1}$ transforms the deformed coordinates back to the initial undeformed geometry.

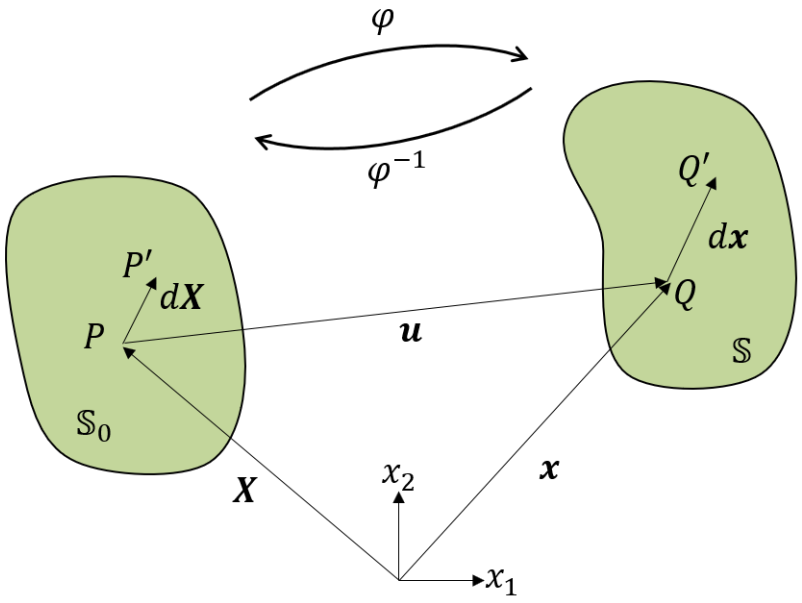


Figure 2.2: Deformation of a material fiber from an initial undeformed domain $\mathbb{S}_0$, with coordinates $\mathbf{X}$, to the final deformed domain $\mathbb{S}$, with coordinates $\mathbf{x}$. The deformation process is denoted by the mapping function $\boldsymbol{\varphi}$, the displacement field is $\mathbf{u}$ and $d\mathbf{X}$ and $d\mathbf{x}$ are infinitesimal vectors.

The coordinates of deformed geometry $\mathbf{x}$ can be obtained with the following relation, which states that the deformed shape is the initial geometry plus the displacement $\mathbf{u}$:

$$\mathbf{x} = \psi(\mathbf{X}) = \mathbf{X} + \mathbf{u}. \tag{2.1}$$

Still in Figure 2.2, an infinitesimal vector $d\mathbf{X}$ in the undeformed geometry, defined by the points P and P' in $\mathbb{S}_0$, is deformed to an infinitesimal vector $d\mathbf{x}$ – defined by the points Q and Q' in $\mathbb{S}$. The relationship between the differential vectors $d\mathbf{x}$ and $d\mathbf{X}$ can be expressed as:

$$d\mathbf{x} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}\, d\mathbf{X}. \tag{2.2}$$

In Equation 2.2, the derivative of each coordinate of $\mathbf{x}$ with respect to each coordinate of $\mathbf{X}$ is called the deformation gradient, denoted by $\mathbf{F}$, which is a recurrent quantity in continuum mechanics and represents the gradient of the mapping function. In matrix notation, we have:

$$\mathbf{F} = \frac{\partial x_i}{\partial X_j} = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} \end{bmatrix}. \tag{2.3}$$

Using the Equation 2.1, we can write the deformed geometry in terms of the displacement:

$$\begin{cases} x_1 = X_1 + u_1 \\ x_2 = X_2 + u_2 \\ x_3 = X_3 + u_3 \end{cases}. \tag{2.4}$$

And the deformation gradient $\mathbf{F}$ becomes:

$$\mathbf{F} = \begin{bmatrix} \frac{\partial u_1}{\partial X_1} + 1 & \frac{\partial u_1}{\partial X_2} \\ \frac{\partial u_2}{\partial X_1} & \frac{\partial u_2}{\partial X_2} + 1 \end{bmatrix}. \tag{2.5}$$

It is worth noting that to avoid singularities in deformations we must have $det\,(\mathbf{F}) > 0$.

We have so far defined the deformed and undeformed geometry and established a mapping function between these two configurations in Equation 2.1. The kinematics is defined by the deformation gradient $\mathbf{F}$, given in Equation 2.5. Once the kinematics is well posed, we can pass to the

next step, that is, to define a strain measure.

2.1.3 Strains

Considering a regime of small deformations, the difference between deformed and undeformed configuration is negligible. We can hence define a strain measure in the deformed shape referential.

A possible definition of strain is the difference between the squared lengths of the infinitesimal vectors $d\mathbf{x}$ and $d\mathbf{X}$ of Equation 2.6, given below. For illustration purposes, these infinitesimal vectors are depicted in Figure 2.2.

$$\begin{aligned} \|d\mathbf{x}\|^2 - \|d\mathbf{X}\|^2 &= d\mathbf{x}^T d\mathbf{x} - d\mathbf{X}^T d\mathbf{X} \\ &= (\mathbf{F}d\mathbf{X})^T \mathbf{F}d\mathbf{X} - d\mathbf{X}^T d\mathbf{X} \\ &= d\mathbf{X}^T \mathbf{F}^T \mathbf{F} d\mathbf{X} - d\mathbf{X}^T d\mathbf{X} \\ &= d\mathbf{X}^T \left(\mathbf{F}^T \mathbf{F} d\mathbf{X} - d\mathbf{X} \right) \\ &= d\mathbf{X}^T \left(\mathbf{F}^T \mathbf{F} - \mathbf{I}_2 \right) d\mathbf{X}. \end{aligned} \tag{2.6}$$

The term $\mathbf{F}^T\mathbf{F}$ in the last statement of Equation 2.6 is fundamental in the continuum mechanics and it is defined as right Cauchy-Green deformation tensor, denoted by $\mathbf{C}$. The term I_2 is the 2×2 identity matrix.

In order to obtain a strain measurement relative to the initial squared length, one can divide the last statement of Equation 2.6 by $d\mathbf{X}^T d\mathbf{X}$. This leads to:

$$\mathbf{E} = \frac{d\mathbf{x}^T d\mathbf{x} - d\mathbf{X}^T d\mathbf{X}}{d\mathbf{X}^T d\mathbf{X}} = \mathbf{F}^T \mathbf{F} - \mathbf{I}_2, \tag{2.7}$$

where $\mathbf{E}$ is the strain tensor. Using the relation for deformation gradient $\mathbf{F}$ presented in Equation 2.5, we can write the strain tensor in terms of the displacement:

$$\mathbf{E} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}^T\nabla\mathbf{u}\right), \tag{2.8}$$

where a factor of $1/2$ is incorporated in the strain measurements to match the infinitesimal strain (KIM, 2015). Furthermore, when only small deformations are considered, the quadratic term of Equation 2.8 – $\nabla\mathbf{u}^T\nabla\mathbf{u}$ – is much smaller than 1 and it can be hence neglected. This leads to the linear strain measurement:

$$\boldsymbol{\epsilon} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right). \tag{2.9}$$

It is important to note that the relation 2.9 is not a true strain measurement once it does not remain constant under rigid-body rotations. Thus, it should only be used in linear problems in small deformations regime, where it is known that the structure does not experiment rigid-body rotations. If it is necessary to deal with large displacements, the quadratic term of Equation 2.8 cannot be neglected and the problem will be nonlinear with the displacement gradient. The geometrically nonlinear cases are discussed in Subsection 2.5.

2.1.4 Constitutive Law

In this work we will only consider plane stress. To do so, we assume that both normal and shear stresses are negligible due to the small thicknesses of the structures studied in this work. The Figure 2.3 illustrates the plane stress approximation, where a 3D structure is loaded in a single plane.

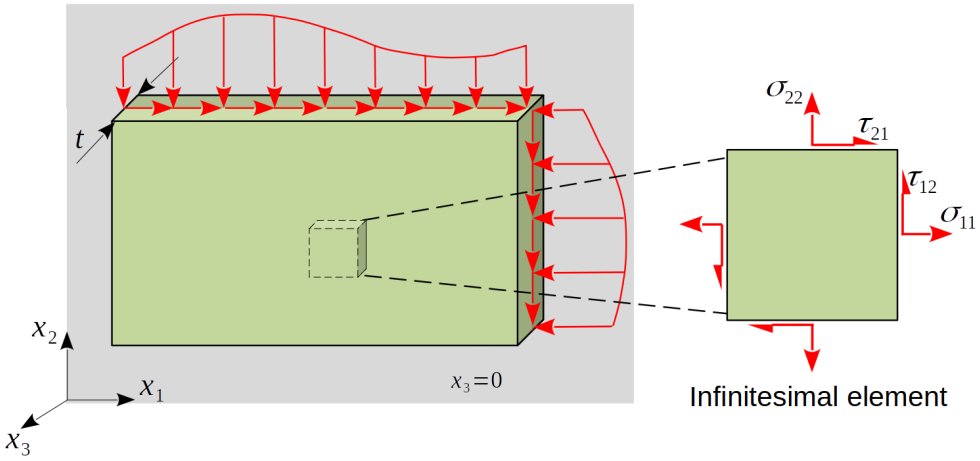


Figure 2.3: Illustration of the plane stress approximation. Three-dimensional structures loaded in a single plane can be approximated by a 2D structure if the thickness t is considerably small – here the load is represented in the plane $x_3 = 0$. In thick structures, the stress components in x_3 direction can be neglected.

In the plane stress, the stress tensor possess three components: σ_{11}, σ_{22} and σ_{12} as introduced the Subsection 2.1.1. Using the Voigt notation, we can establish a linear relation between $\boldsymbol{\sigma}$ and $\boldsymbol{\epsilon}$ as follows:

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{11}(\mathbf{x}) \\ \sigma_{22}(\mathbf{x}) \\ \tau_{12}(\mathbf{x}) \end{Bmatrix} = \mathbf{D} \begin{Bmatrix} \epsilon_{11}(\mathbf{x}) \\ \epsilon_{22}(\mathbf{x}) \\ 2\epsilon_{12}(\mathbf{x}) \end{Bmatrix}. \tag{2.10}$$

Using the strain definition of Equation 2.9, the strain tensor is written as:

$$\boldsymbol{\epsilon} = \begin{Bmatrix} \epsilon_{11}(\mathbf{x}) \\ \epsilon_{22}(\mathbf{x}) \\ 2\epsilon_{12}(\mathbf{x}) \end{Bmatrix} = \mathbf{L}\mathbf{u} = \begin{Bmatrix} \frac{\partial u_1(\mathbf{x})}{\partial x_1} \\ \frac{\partial u_2(\mathbf{x})}{\partial x_2} \\ \frac{\partial u_1(\mathbf{x})}{\partial x_2} + \frac{\partial u_2(\mathbf{x})}{\partial x_1} \end{Bmatrix}, \tag{2.11}$$

where $\mathbf{D}$ is a 3×3 matrix called stress-strain matrix or constitutive matrix. The constitutive law of Equation 2.10 is called St Venant-Kirchhoff model. For plane stress cases, $\mathbf{D}$ is given by:

$$\mathbf{D} = \frac{E_y}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}. \tag{2.12}$$

The material constants E_y and ν are the Young's modulus and the Poisson coefficient, respectively. This constitutive law, called Saint Venant-Kirchoff model.

2.1.5 Equilibrium Equations

In two-dimensional solid mechanic we work with systems similar to the one of Figure 2.4. It is a solid with an internal domain Ω and with a smooth boundary Γ. It can be subjected to point and distributed loads – called Neumann or natural boundary conditions – and to displacement supports – called Dirichlet or essential boundary conditions.

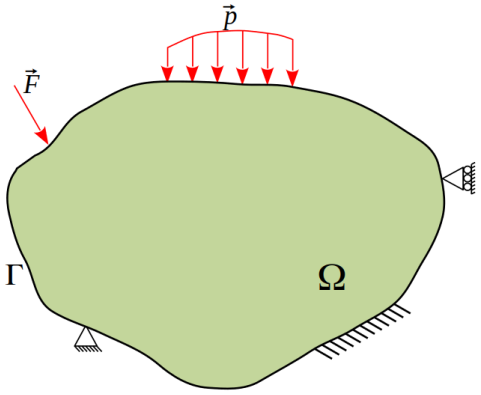


Figure 2.4: Deformable body subjected to external loads and displacement restrictions. The solution domain is represented by Ω and its boundary by Γ.

The strong formulation of equilibrium equations for the linear elasticity will be deduced using a balance of forces and linear momentum, in a newtonian approach. The principle of virtual work is employed to obtain the weak-formulation of the elasticity problem, which is used in the EFG method.

In a 2D approach and we assume that all variables and external loads depend only on x_1

and x_2 directions. To deduce the strong formulation of the two-dimension linear elasticity, one can apply the force equilibrium to the infinitesimal solid element of Figure 2.5. The force equilibrium in x_1 and x_2 directions is:

$$\begin{aligned} \frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \tau_{12}}{\partial x_2} + f_1 = 0 \\ \frac{\partial \sigma_{22}}{\partial x_s} + \frac{\partial \tau_{12}}{\partial x_1} + f_2 = 0. \end{aligned} \tag{2.13}$$

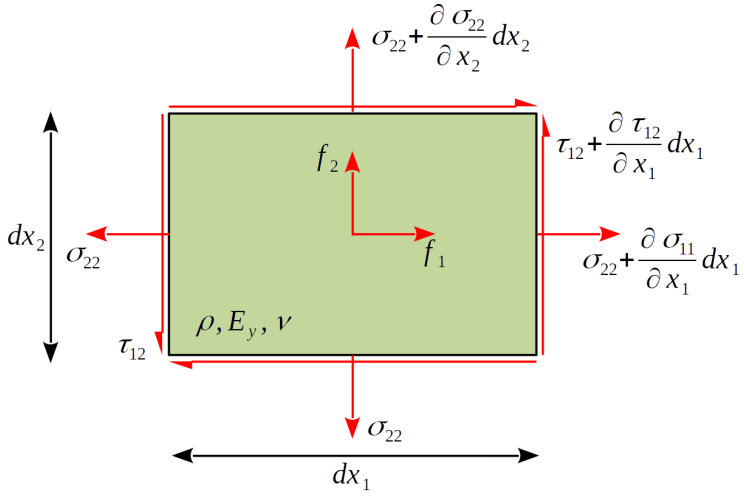


Figure 2.5: Infinitesimal element represented with stress and body forces.

Here, ρ is the material density, u_1 and u_2 are displacements in x_1 and x_2 direction respectively. The variables f_1 and f_2 are the body forces. Here we are concerned with static structural analysis, so the modeled structures does not experiment any variation with time thus the terms in the right-hand side of Equation 2.13 are set to zero. The forces f_1 and f_2 are also set to zero as gravitational or magnetic effects are not taken into account in this work. Thus, the Equation 2.13 can be simplified to:

$$\begin{aligned} \frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \tau_{12}}{\partial x_2} = 0 \\ \frac{\partial \sigma_{22}}{\partial x_2} + \frac{\partial \tau_{12}}{\partial x_1} = 0. \end{aligned} \tag{2.14}$$

The force balance of Equation 2.14 is valid for any elemental stress over the entire domain Ω. By introducing a matrix notation, the Equations 2.14 can be written in a more compact form:

$$\mathbf{L}^T \boldsymbol{\sigma} = 0, \tag{2.15}$$

where $\mathbf{L}$ is the linear differential operator, defined as:

$$\mathbf{L} = \begin{bmatrix} \partial/\partial x_1 & 0 \\ 0 & \partial/\partial x_2 \\ \partial/\partial x_2 & \partial/\partial x_1 \end{bmatrix}. \tag{2.16}$$

The boundary conditions – natural and essential – are added to the problem through two extra equations:

$$\begin{aligned} \mathbf{L}^T \boldsymbol{\sigma} &= 0 \\ \mathbf{u} &= \bar{\mathbf{u}} \\ \boldsymbol{\sigma}\mathbf{n} &= \bar{\mathbf{t}}. \end{aligned} \tag{2.17}$$

In Equation 2.17 the prescribed displacement is denoted by $\bar{\mathbf{u}}$ and $\mathbf{n}$ is a unitary vector normal, outward to the surface. The prescribed traction, which can account for both distributed and point loads, it is denoted by $\bar{\mathbf{t}} = \begin{bmatrix} \bar{t}_1(\mathbf{x}) & \bar{t}_2(\mathbf{x}) \end{bmatrix}^T$.

It is worth noting that, although the Equation 2.17 has only a linear differential operator applied to the stress $\boldsymbol{\sigma}$, the elasticity problem is a second order problem with relation to the displacement field. Indeed, the stress field is obtained from the strain field, which in turn, results from the derivative of the displacement field. In this way, the strong formulation requires the displacement field to be such that the second-order derivatives are continuous.

An alternative way of stating the equilibrium equations is using an energy balance instead

of linear momentum equilibrium. It considers that the mechanical structure will deform under the applied load and will resist to deformation by generating internal forces. When a structure deforms, the internal forces increase and the stored energy also increases. The stored deformation energy is called strain energy and can be written as:

$$U(\mathbf{u}) = \frac{1}{2}\int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) \colon \boldsymbol{\epsilon}(\mathbf{u})\, d\boldsymbol{\Omega}. \tag{2.18}$$

Here the double dots product $:$ of two matrices produces a scalar. Considering two 3×3 second order tensors $\mathbf{Q}$ and $\mathbf{S}$, the double dots product is defined as:

$$\begin{aligned} \mathbf{Q} \colon \mathbf{S} = & Q_{11}S_{11} + Q_{12}S_{12} + Q_{13}S_{13} + \\ & Q_{21}S_{21} + Q_{22}S_{22} + Q_{23}S_{23} + \\ & Q_{31}S_{31} + Q_{32}S_{32} + Q_{33}S_{33}. \end{aligned} \tag{2.19}$$

As the structure deforms, the external load – traction and body forces – produces work. The work done by external forces can be written as:

$$W(\mathbf{u}) = \iint_{\Omega} \mathbf{u} \cdot \mathbf{f} d\boldsymbol{\Omega} + \int_{\Gamma} \mathbf{u} \cdot \bar{\mathbf{t}}\, d\Gamma. \tag{2.20}$$

As previously stated, the body forces are not considered in this work hence the term of $\mathbf{f}$ becomes zero:

$$W(\mathbf{u}) = \int_{\Gamma} \mathbf{u} \cdot \bar{\mathbf{t}}\, d\Gamma. \tag{2.21}$$

The structure's total potential energy, here denoted by Π, is obtained from the subtraction of the work of external loads – $W(\mathbf{u})$ – from the internal energy – $U(\mathbf{u})$ – , as follows:

$$\begin{aligned}\Pi(\mathbf{u}) &= U(\mathbf{u}) - W(\mathbf{u}) \\ &= \frac{1}{2}\int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) \colon \boldsymbol{\epsilon}(\mathbf{u})\, d\boldsymbol{\Omega} - \int_{\Gamma} \mathbf{u}\cdot\bar{\mathbf{t}}\, d\Gamma.\end{aligned} \tag{2.22}$$

The principle of minimum total potential energy states that: *of all kinematically admissible configurations, the deformation producing the minimum total potential energy is the stable equilibrium conditions.* The space of kinematically admissible configurations are those that satisfy the boundary conditions. Taking the stationary value of Equation 2.22, we obtain

$$0 = \frac{\partial}{\partial \mathbf{u}}\left(\frac{1}{2}\int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) \colon \boldsymbol{\epsilon}(\mathbf{u}) d\boldsymbol{\Omega} - \int_{\Gamma} \mathbf{u}\cdot\bar{\mathbf{t}}\, d\Gamma\right). \tag{2.23}$$

2.2 FEM Discretization

The FEM discretization is obtained in a straight forward manner with a weighted residual method applied to the equilibrium Equations 2.13. Below the residual method used to obtain the FEM formulation is presented in a simplified way. For more details one can consult (KWON AND BANG, 2000), upon which is based this subsection.

In weighted residual methods, a residual of the approximated solution is evaluated over the entire domain Ω and is boundaries. So applying the weighted residual to Equation 2.14, we have:

$$\begin{aligned}\int_{\Omega} \omega_1 \left(\frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \tau_1}{\partial x_2}\right) d\Omega + \int_{\Gamma} \omega_1\, \bar{t}_1\, d\Gamma = 0 \\ \int_{\Omega} \omega_2 \left(\frac{\partial \sigma_{22}}{\partial x_2} + \frac{\partial \tau_{12}}{\partial x_1}\right) d\Omega + \int_{\Gamma} \omega_2\, \bar{t}_2\, d\Gamma = 0.\end{aligned} \tag{2.24}$$

The functions ω_1 and ω_2 are called weight functions. Their definition and how they can be constructed will be presented latter on this subsection. The boundary traction $\bar{\mathbf{t}} = [\bar{t}_1,\bar{t}_2]^T$ are given in terms of the prescribed traction values and the unitary vector normal to the boundary Γ. In this way, we can write the vector $\bar{\mathbf{t}}$ as follows:

$$\begin{aligned} \bar{t}_1 &= \sigma_{11} n_1 + \tau_{12} n_2 \\ \bar{t}_2 &= \tau_{12} n_1 + \sigma_{22} n_2. \end{aligned} \tag{2.25}$$

Applying integration by parts, the Equation 2.24 becomes:

$$-\int_\Omega \begin{Bmatrix} \frac{\partial \omega_1}{\partial x_1}\sigma_{11} + \frac{\partial \omega_1}{\partial x_2}\sigma_{12} \\ \frac{\partial \omega_2}{\partial x_1}\sigma_{12} + \frac{\partial \omega_2}{\partial x_2}\sigma_{22} \end{Bmatrix} d\Omega + \int_\Gamma \begin{Bmatrix} \omega_1 \ \bar{t}_1 \\ \omega_2 \ \bar{t}_2 \end{Bmatrix} d\Gamma = 0. \tag{2.26}$$

Which can be rewritten in matrix form for a more compact notation:

$$-\int_\Omega \begin{bmatrix} \frac{\partial \omega_1}{\partial x_1} & 0 & \frac{\partial \omega_1}{\partial x_2} \\ 0 & \frac{\partial \omega_2}{\partial x_2} & \frac{\partial \omega_2}{\partial x_1} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix} d\Omega + \int_\Gamma \begin{Bmatrix} \omega_1 \ \bar{t}_1 \\ \omega_2 \ \bar{t}_2 \end{Bmatrix} = 0, \tag{2.27}$$

or

$$\int_\Omega \begin{bmatrix} \frac{\partial \omega_1}{\partial x_1} & 0 & \frac{\partial \omega_1}{\partial x_2} \\ 0 & \frac{\partial \omega_2}{\partial x_2} & \frac{\partial \omega_2}{\partial x_1} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \end{Bmatrix} d\Omega = \int_\Gamma \begin{Bmatrix} \omega_1 \bar{t}_1 \\ \omega_2 \bar{t}_2 \end{Bmatrix} d\Gamma. \tag{2.28}$$

Applying the definition of strain and stress presented in Sections 2.1.3 and 2.1.4 we can relate the strain field with the external loads using the constitutive relation:

$$\int_\Omega \begin{bmatrix} \frac{\partial \omega_1}{\partial x_1} & 0 & \frac{\partial \omega_1}{\partial x_2} \\ 0 & \frac{\partial \omega_2}{\partial x_2} & \frac{\partial \omega_2}{\partial x_1} \end{bmatrix} [D] \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{12} \end{Bmatrix} d\Omega = \int_\Gamma \begin{Bmatrix} \omega_1 \bar{t}_1 \\ \omega_2 \bar{t}_2 \end{Bmatrix} d\Gamma. \tag{2.29}$$

The domain Ω will be discretized using rectangular elements with a four nodes rectangular element – hereafter *quad4* element –, shown in Figure 2.6. For this element, the nodal displacements are interpolated using the bi-linear shape functions N_i:

$$\mathbf{u} = \mathbf{N}\mathbf{u_e}, \tag{2.30}$$

where $\mathbf{N}$ is a matrix containing the shape functions of each node of the *quad4* element and $\mathbf{u_e}$ is an array that collects the nodal diplacement of the *quad4* element. In matrix form, we have:

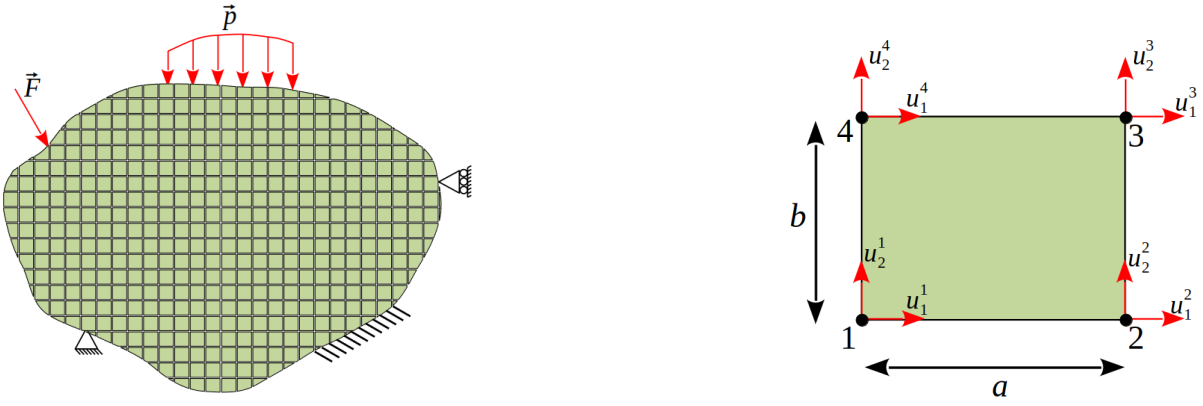
$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix}, \tag{2.31}$$

and

$$\mathbf{u_e} = \begin{Bmatrix} u_1^1 & u_2^1 & u_1^2 & u_2^2 & u_1^3 & u_2^3 & u_1^4 & u_2^4 \end{Bmatrix}^T. \tag{2.32}$$

The shape functions of each node i for a *quad4* element are given in Equations 2.33. A shape function N_i is written in terms of lengths a, b and the coordinates x_1 and x_2 of each point inside an element, as shown in Figure 2.6. These shape functions are called bi-linear shape functions and are illustrated in Figure 2.7. One may note that the shape function of a node i assumes value of 1 at this node coordinates and vanishes for all other nodes.

$$\begin{aligned} N_1(x_1,x_2) &= \frac{(a-x_1)(b-x_2)}{ab} \\ N_2(x_1,x_2) &= \frac{x_1(b-x_2)}{ab} \\ N_3(x_1,x_2) &= \frac{x_1 x_2}{ab} \\ N_4(x_1,x_2) &= \frac{x_2(a-x_1)}{ab} \end{aligned} \tag{2.33}$$



(a) (b)

Figure 2.6: Illustration (a) shows the domain discretized using the *quad4* element. Illustration (b) gives more detail on the DOF of *quad4* element: it is a rectangular element with 4 nodes and two degrees of freedom per node - in x_1 and x_2 directions.

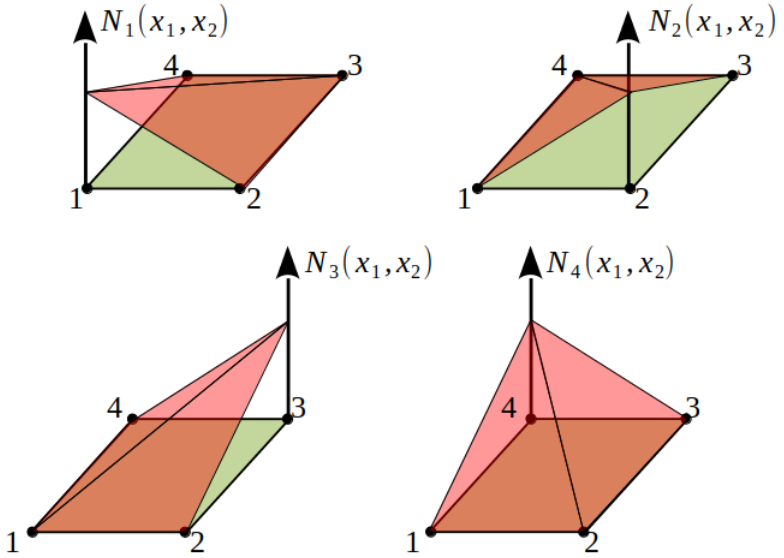


Figure 2.7: Bi-linear shape functions used in FEM. These functions are only defined inside an element and assume values 1 in their correspondent node and vanishes in all other nodes.

In a similar way to the discretization of the displacement field in Equation 2.30, the strain tensor can also be interpolated using the derivatives of the shape functions – the $\mathbf{B}$ matrix – and the elemental displacement vector $\mathbf{u_e}$:

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{u}_\mathbf{e}. \tag{2.34}$$

Here, $\mathbf{B}$ is a matrix containing the shape function derivatives and it is given by:

$$\mathbf{B} = \begin{bmatrix} N_{1,1} & 0 & N_{2,1} & 0 & N_{3,1} & 0 & N_{4,1} & 0 \\ 0 & N_{1,2} & 0 & N_{2,2} & 0 & N_{3,2} & 0 & N_{4,2} \\ N_{1,2} & N_{1,1} & N_{2,2} & N_{2,1} & N_{3,2} & N_{3,1} & N_{4,2} & N_{4,1} \end{bmatrix}. \tag{2.35}$$

This far, we know how to use the shape functions and their derivatives to discretize the displacement and strain field. However, it is still necessary to define the weight function ω_1 and ω_2 in Equation 2.29 to finish the FEM discretization process. In Galerkin methods, the weight functions ω_i are chosen to be equal to shape functions. In this way we have $\omega_1 = [N_1,N_2,N_3,N_4]^T$ and $\omega_2 = [N_1,N_2,N_3,N_4]^T$. This is to say that each variable ω_i in Equation 2.29 should be replaced by the 4×1 vector $\mathbf{N} = [N_1,N_2,N_3,N_4]^T$.

The Equation 2.29 then becomes:

$$\int_\Omega \begin{bmatrix} \frac{\partial \omega_1}{\partial x_1} & 0 & \frac{\partial \omega_1}{\partial x_2} \\ 0 & \frac{\partial \omega_2}{\partial x_2} & \frac{\partial \omega_2}{\partial x_1} \end{bmatrix} \mathbf{D} \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{12} \end{Bmatrix} d\Omega =$$

$$\mathop{\mathrm{A}}_{i=1}^{nel} \int_a \int_b \begin{bmatrix} \frac{\partial N_1}{\partial x_1} & 0 & \frac{\partial N_1}{\partial x_2} \\ 0 & \frac{\partial N_1}{\partial x_1} & \frac{\partial N_1}{\partial x_1} \\ \frac{\partial N_2}{\partial x_1} & 0 & \frac{\partial N_2}{\partial x_2} \\ 0 & \frac{\partial N_2}{\partial x_1} & \frac{\partial N_2}{\partial x_1} \\ \frac{\partial N_3}{\partial x_1} & 0 & \frac{\partial N_3}{\partial x_2} \\ 0 & \frac{\partial N_3}{\partial x_1} & \frac{\partial N_3}{\partial x_1} \\ \frac{\partial N_4}{\partial x_1} & 0 & \frac{\partial N_4}{\partial x_2} \\ 0 & \frac{\partial N_4}{\partial x_1} & \frac{\partial N_4}{\partial x_1} \end{bmatrix} \mathbf{D} \begin{bmatrix} \frac{\partial N_1}{\partial x_1} & 0 & \frac{\partial N_2}{\partial x_1} & 0 & \frac{\partial N_3}{\partial x_1} & 0 & \frac{\partial N_4}{\partial x_1} & 0 \\ 0 & \frac{\partial N_1}{\partial x_2} & 0 & \frac{\partial N_2}{\partial x_2} & 0 & \frac{\partial N_3}{\partial x_2} & 0 & \frac{\partial N_4}{\partial x_2} \\ \frac{\partial N_1}{\partial x_2} & \frac{\partial N_1}{\partial x_1} & \frac{\partial N_2}{\partial x_2} & \frac{\partial N_2}{\partial x_1} & \frac{\partial N_3}{\partial x_2} & \frac{\partial N_3}{\partial x_1} & \frac{\partial N_4}{\partial x_2} & \frac{\partial N_4}{\partial x_1} \end{bmatrix} \mathbf{u}_\mathbf{e}^\mathbf{i} \, dx_1 \, dx_2. \tag{2.36}$$

In Equation 2.36 two important modifications with relation to Equation 2.29 were done. First,

the elements of the matrix containing the weight function derivatives were reorganized to be in accordance with the elemental displacement vector $\mathbf{u_e}$. Second, the integral over the domain Ω were replaced by a summation – denoted by A – over all the *quad4* elements, where nel represents the total number of element. The summation A is an special operator: it is called the *assembly* operator, which performs the summation of all DOFs – u_1^i and u_2^i – present in the discretization.

Indeed, during the assembly of the global system, the contribution of each element i needs to be taken into account. The contribution of each element i is called elemental or local stiffness, while the matrix resulting from the assembly operation is called global stiffness matrix.

One may also note that in Equation 2.36 the matrix containing shape function derivatives are the same $\mathbf{B}$ matrix defined in Equation 2.34. In this way, the elemental stiffness can be written in a compact form:

$$\mathbf{K_e} = \mathop{\mathrm{A}}_{a}^{b} \int\int \mathbf{B}^T \mathbf{D}\ \mathbf{B}\ dx_1 dx_2. \tag{2.37}$$

The global force is calculated in a similar way. The imposed traction forces are discretized using the shape functions and integration along the boundary, as shown in Equation 2.38. When point forces are considered, the integral solution is straightforward and for these case the nodal forces can be imposed directly in the global force vector. The same assembly operator A can be used to correctly assign the nodal forces to each DOF in the global force vector.

$$\mathbf{f_g} = \int_\Gamma \mathbf{N}^T \begin{Bmatrix} \bar{t}_1 \\ \bar{t}_2 \end{Bmatrix}\ d\Gamma. \tag{2.38}$$

It is still necessary to calculate the integrals of Equation 2.37 and 2.38. Due to the simplicity of bi-linear shape functions, the integrals are calculated analytically for the *quad4* rectangular element. The elemental matrix are assembled into the global stiffness matrix considering a nodal numbering scheme, as shown in Figure 2.6.

Considering that each node posses 2 DOF and the total number of nodes is $n = 4 \times nel$, the

global matrix $\mathbf{K_g}$ is a $2n \times 2n$ matrix, accounting 2 DOF for each node and it is the sum of all element stiffness matrix. Similarly, the global forces vector $\mathbf{f_g}$ is a $2n \times 1$ vector. As a result, we have the following linear system, obtained from the discretization of Equation 2.29:

$$\mathbf{K_g u} = \mathbf{f_g}, \tag{2.39}$$

where $\mathbf{K_g}$ is the global stiffness matrix, $\mathbf{u}$ is the displacement field and $\mathbf{f_g}$ is the force vector, accounting both point and distributed forces.

2.3 EFG Discretization

The basic idea of EFG is to approximate the displacement field $\mathbf{u}(\mathbf{x})$ of the weak formulation of Equation 2.23 using a finite number n of approximated nodal values $\hat{\mathbf{u}}^{hk}$ and the shape functions $\phi^k(\mathbf{x})$. Each node k has a shape function, and we can approximate the displacement field and its derivatives as follows:

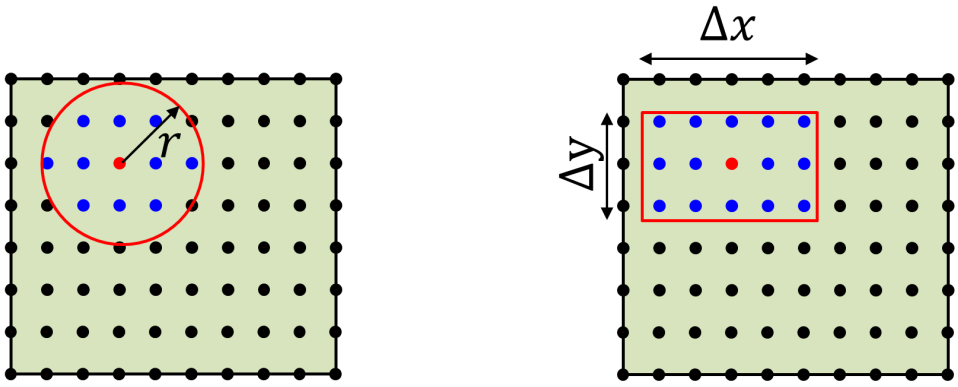
$$\mathbf{u}(\mathbf{x}) \approx \mathbf{u}^h(\mathbf{x}) = \sum_{k=1}^{n} \phi^k(\mathbf{x}) \hat{\mathbf{u}}^{hk}, \tag{2.40}$$

and

$$\mathbf{u}(\mathbf{x})_{,j} \approx \mathbf{u}^h_{,j}(\mathbf{x}) = \sum_{k=1}^{n} \phi^k_{,j}(\mathbf{x}) \hat{\mathbf{u}}^{hk}. \tag{2.41}$$

Here, $\mathbf{x}$ are the coordinates where we want the approximation, $\mathbf{x}^k$ are the coordinates of discretization nodes and $\hat{\mathbf{u}}^{hk} = \hat{\mathbf{u}}^h(\mathbf{x}^k)$ is the approximated nodal value of the displacement field. The function $\phi(\mathbf{x})^k$ is the shape function of node k evaluated at coordinates $\mathbf{x}$. This approximation is similar to the one used in FEM, where the nodal displacement of an element is interpolated using the bi-linear shape functions $\mathbf{N}$. Differently from FEM, the EFG uses approximated nodal values.

While in FEM the shape functions are defined inside an element of a structured mesh, in meshless methods they are defined in regions called influence domain. The influence domain is the neighborhood of a given node. This neighborhood can be defined as the region inside a circle of radius r centered in the interest node. Alternatively, a rectangle with sides Δx, Δy centered in the interest node can also be used. The influence domain concept is illustrated in Figure 2.8.



(a) Circular (b) Rectangular

Figure 2.8: Illustration of the influence domain of a node. In the EFG method, the shape functions of a node is active inside the influence domain of this node and vanishes for all positions outside.

For example, the nodes with coordinates $\mathbf{x}^{in}$ that lies inside the neighborhood of a node with coordinates $\mathbf{x}^k$ will be influenced by this node once the shape function $\phi^k(\mathbf{x})$ is active in this region. Equivalently, all nodes with coordinates $\mathbf{x}^{out}$ will not be influenced because the shape function vanishes for all nodes lying outside the influence domain of coordinates $\mathbf{x}^k$.

An important aspect of the EFG method is hence the construction of the shape functions, which will be discussed in the next subsection. With illustration purposes, Figure 2.9 shows the general aspects of a shape function: it is active only inside the nodal influence domain and it is decreasing with distance from its correspondent node. In the current section, however, we are concerned with the construction of EFG approximation and with the discretization of the linear elasticity problem. We will, for now, consider that $\phi(\mathbf{x})$ exists and it is at least C^1 continuous.

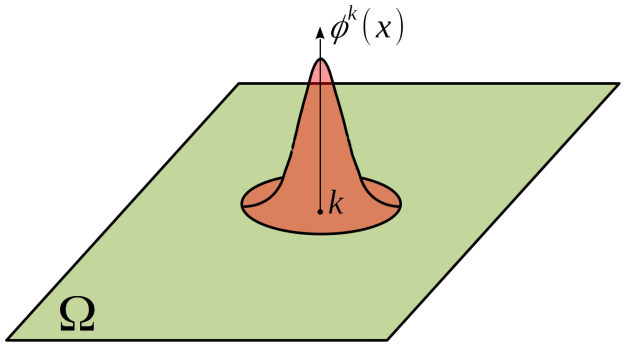


Figure 2.9: Illustration of the general aspects of a shape function. Here the shape function $\phi(\mathbf{x})^k$ of a given node k is drawn. The shape function assumes its maximum value at the coordinate $\mathbf{x}^k$ of its correspondent node and decreases as the distance from $\mathbf{x}^k$ increases.

The next step in the EFG construction is to replace the approximation of the displacement field and its derivatives in weak formulation of Equation 2.23 - (LIU, 2003), (OVERVELDE, 2012). The approximations of Equations 2.40 and 2.41 can be written in a more compact way using the matrix notation:

$$\mathbf{u}(\mathbf{x}) \approx \begin{Bmatrix} u_1^h(\mathbf{x}) \\ u_2^h(\mathbf{x}) \end{Bmatrix} = \begin{bmatrix} \sum_{k=1}^n \phi^k(\mathbf{x})\hat{u}_1^{hk} \\ \sum_{k=1}^n \phi^k(\mathbf{x})\hat{u}_2^{hk} \end{bmatrix} = \sum_{k=1}^n \boldsymbol{\phi}^k(\mathbf{x})\hat{\mathbf{u}}^{hk}. \tag{2.42}$$

With this relation we can write the displacement field $\mathbf{u}(\mathbf{x})$ in terms of approximated nodal values $\hat{\mathbf{u}}^{hk}$. In Equation 2.42 the final terms are:

$$\boldsymbol{\phi}^k(\mathbf{x}) = \begin{bmatrix} \phi^k(\mathbf{x}) & 0 \\ 0 & \phi^k(\mathbf{x}) \end{bmatrix}, \tag{2.43}$$

and

$$\hat{\mathbf{u}}^{hk} = \begin{Bmatrix} \hat{u}_1^{hk} & \hat{u}_2^{hk} \end{Bmatrix}^T. \tag{2.44}$$

So, in a compact form, the discretized displacement field approximation can be written as:

$$\mathbf{u}(\mathbf{x}) \approx \mathbf{\Phi U}, \tag{2.45}$$

where $\mathbf{\Phi}$ is a matrix containing the shape function of all n discretization nodes evaluated at the coordinates of interest $\mathbf{x}$, and the vector $\mathbf{U}$ is a vector collecting the approximated nodal values of displacement. They are written as:

$$\mathbf{\Phi} = \begin{bmatrix} \phi^1(\mathbf{x}) & 0 & \dots & \phi^n(\mathbf{x}) & 0 \\ 0 & \phi^1(\mathbf{x}) & \dots & 0 & \phi^n(\mathbf{x}) \end{bmatrix}_{2\times 2n}, \tag{2.46}$$

and

$$\mathbf{U} = \begin{Bmatrix} u_1^1 & u_2^1 & \dots & u_1^n & u_2^n \end{Bmatrix}^T_{1\times 2n}. \tag{2.47}$$

A shape function of an arbitrary node k evaluated in q points with coordinates $\mathbf{x} = \left[\left(x_1^1,x_2^1\right),\dots,\left(x_1^q,x_2^q\right)\right]$ is written as: $\phi^k(\mathbf{x}) = \boldsymbol{\phi}^k = \left[\phi^1\left(x_1^1,x_2^1\right),\dots,\phi^1\left(x_1^q,x_2^q\right)\right]$. The shape function vanishes for coordinates lying outside the influence domain of node k. Thus, $\boldsymbol{\phi}^k$ is evaluated only for coordinates inside the influence domain of node k. Moreover, as it explained later in the chapter, the coordinates $\mathbf{x}$ are thode of the Gauss points used in the integration of the weak-formulation.

One may note an important difference between the shape functions used in EFG and those used in FEM. Here, $\boldsymbol{\phi}^k$ is not a 3-by-8 matrix as in FEM, instead, it is assumes different sizes depending on the number of nodes inside the influence domain of each node. This number may vary, for example, if a node lies in the central portion of the solution domain Ω or in a corner.

The derivatives of the displacement field can also be discretized by simply applying the linear differential operator to the approximated displacement field:

$$\mathbf{Lu}(\mathbf{x}) \approx \sum_{k=1}^{n} \mathbf{L}\ \boldsymbol{\phi}^k\ \hat{\mathbf{u}}^{hk} = \mathbf{L\Phi U}, \tag{2.48}$$

where the matrix containing the shape function derivatives is denoted by $\mathbf{B}$, and has the following form:

$$\mathbf{L\Phi} = \mathbf{B} = \begin{bmatrix} \phi^1_{,1}(\mathbf{x}) & 0 & \dots & \phi^n_{,1}(\mathbf{x}) & 0 \\ 0 & \phi^1_{,2}(\mathbf{x}) & \dots & 0 & \phi^n_{,2}(\mathbf{x}) \\ \phi^1_{,2}(\mathbf{x}) & \phi^1_{,1}(\mathbf{x}) & \dots & \phi^n_{,1}(\mathbf{x}) & \phi^n_{,2}(\mathbf{x}) \end{bmatrix}. \tag{2.49}$$

The method to obtain the shape function and its derivatives will be discussed in the next subsection.

Finally, the strain field $\boldsymbol{\epsilon} = \mathbf{Lu}(\mathbf{x})$ is written in terms of $\mathbf{B}$ and the nodal values of displacement $\mathbf{U}$s:

$$\mathbf{Lu}(\mathbf{x}) \approx \mathbf{BU}. \tag{2.50}$$

Now that the displacement field and its derivatives are discretized, we can replace these approximations in the weak formulation of Equation 2.23. This equation is rewritten below in a variational form, where the test functions are chosen to be similar displacement field. The Lagrange multipliers are used to enforce essential boundary conditions. We have:

$$\int_\Omega \delta \left[\mathbf{Lu}\right]^T \mathbf{D} \left[\mathbf{Lu}\right] d\Omega - \int_\Omega \delta \mathbf{u}^T \mathbf{f} d\Omega - \int_\Gamma \delta \mathbf{u}^T \mathbf{t} d\Gamma - \int_\Gamma \delta \boldsymbol{\lambda}^T \left[\mathbf{u} - \bar{\mathbf{u}}\right] \, d\Gamma - \int_\Gamma \delta \mathbf{u} \boldsymbol{\lambda}^T \, d\Gamma = 0. \tag{2.51}$$

Applying the discretization to the first term of Equation 2.51:

$$\int_\Omega \delta \left[\mathbf{Lu}\right]^T \mathbf{D} \left[\mathbf{Lu}\right] d\Omega = \int_\Omega \delta \mathbf{U}^T \mathbf{D} \left[\mathbf{BU}\right] d\Omega = \int_\Omega \delta \mathbf{U}^T \mathbf{B}^T \mathbf{DBU} \, d\Omega. \tag{2.52}$$

As the terms $\hat{\mathbf{u}}^h$ and its variational $\delta\hat{\mathbf{u}}^h$ are constant nodal values, they can be removed from

the integral, leading to:

$$\int_{\Omega} \delta \left[\mathbf{Lu}\right]^T \mathbf{D} \left[\mathbf{Lu}\right] d\Omega = \delta \mathbf{U}^T \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \, d\Omega \, \mathbf{U}. \tag{2.53}$$

The integral in the above equation also appears in the FEM discretization and we can recognize it as the global stiffness matrix. We have then:

$$\mathbf{K_g} = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \, d\Omega. \tag{2.54}$$

The global stiffness matrix is composed by n sub-matrix with size 2×2, called nodal or local stiffness matrix, as shown below:

$$\mathbf{K_g} = \begin{bmatrix} \mathbf{K}_{11} & \dots & \mathbf{K}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{n1} & \dots & \mathbf{K}_{nn} \end{bmatrix} \;, \; \mathbf{K}_{ij} = \int_{\Omega} \mathbf{B}^{i^T} \mathbf{D} \mathbf{B}^j \, d\Omega. \tag{2.55}$$

The first term of Equation 2.51 is then written as:

$$\int_{\Omega} \delta \left[\mathbf{Lu}\right]^T \mathbf{D} \left[\mathbf{Lu}\right] d\Omega = \delta \mathbf{U}^T \mathbf{K_g} \mathbf{U}. \tag{2.56}$$

Here, $\mathbf{B}^i$ refers to a matrix containing only the derivatives of the shape functions relative to node i. Differently from FEM, the local stiffness matrix needs to be integrated over the entire domain or, more efficiently, over the influence domain of a node, once the shape function vanishes for all positions outside the influence domain of node. This integration is performed using the Gauss Quadrature method, explained later on this subsection. For now, we will firstly state the discretization for the other terms of weak formulation 2.51.

The second term in Equation 2.51, corresponding to body forces contributions, are discretized in a similar way:

$$\int_{\Gamma} \delta \mathbf{u}^T \mathbf{f} \, d\Omega = \int_{\Omega} \delta \left[\mathbf{\Phi U}\right]^T \mathbf{f} \, d\Omega = \int_{\Omega} \delta \mathbf{U}^T \mathbf{\Phi}^T \mathbf{f} \, d\Omega = \delta \mathbf{U}^T \int_{\Omega} \mathbf{\Phi}^T \mathbf{f} \, d\Omega, \tag{2.57}$$

where the integral over the domain is called global body force vector:

$$\mathbf{F_b} = \int_{\Omega} \mathbf{\Phi}^T \mathbf{f} \, d\Omega. \tag{2.58}$$

In this way, the body forces term is written as:

$$\int_{\Omega} \delta \mathbf{u}^T \mathbf{f} \, d\Omega = \delta \mathbf{U}^T \mathbf{F_b}. \tag{2.59}$$

The third term in Equation 2.51, relative to the distributed loads, becomes:

$$\int_{\Gamma} \delta \mathbf{u}^T \mathbf{t} d\Gamma = \int_{\Gamma} \delta \left[\mathbf{\Phi U}\right]^T \mathbf{p} \, d\Gamma = \int_{\Gamma} \delta \mathbf{U}^T \mathbf{\Phi}^T \mathbf{p} \, d\Gamma = \delta \mathbf{U}^T \int_{\Gamma} \mathbf{\Phi}^T \mathbf{p} \, d\Gamma. \tag{2.60}$$

The integral in the above equation is the global traction vector:

$$\mathbf{F_t} = \int_{\Gamma} \mathbf{\Phi}^T \mathbf{p} \, d\Gamma. \tag{2.61}$$

Thus, the traction loads term is:

$$\int_{\Gamma} \delta \mathbf{u}^T \mathbf{t} d\Gamma = \delta \mathbf{U}^T \mathbf{F_t}. \tag{2.62}$$

Now, for the Lagrange Multiplier terms, we can approximate the multipliers $\boldsymbol{\lambda}$ using the same strategy employed with the displacement field. We have:

$$\boldsymbol{\lambda}(\mathbf{x}) \approx \begin{Bmatrix} \hat{\lambda}_1(\mathbf{x}) \\ \hat{\lambda}_2(\mathbf{x}) \end{Bmatrix} = \begin{bmatrix} \sum_{i=1}^{n_b} \mathbf{N}^i(\mathbf{x})\lambda_1^i \\ \sum_{i=1}^{n_b} \mathbf{N}^i(\mathbf{x})\lambda_2^i \end{bmatrix} = \sum_{i=1}^{n_b} \mathbf{N}(\mathbf{x})\boldsymbol{\lambda}^i. \tag{2.63}$$

Here, the summation is over the boundary nodes n_b and the shape function matrix is denoted $\mathbf{N}$ and is written as:

$$\mathbf{N} = \begin{bmatrix} N^1(\mathbf{x}) & 0 & \dots & N^{n_b}(\mathbf{x}) & 0 \\ 0 & N^1(\mathbf{x}) & \dots & 0 & N^{n_b}(\mathbf{x}) \end{bmatrix}. \tag{2.64}$$

The notation for these shape functions changed because for essential boundary conditions imposition, the FEM shape functions can be employed (BELYTSCHKO ET AL., 1994), (DOLBOW AND BELYTSCHKO, 1998), (OVERVELDE, 2012) – of course, the shape functions ϕ could also be used. Actually, the FEM shape functions are more simple and computationally less expansive. Hence, in this work, the 1D linear FEM shape function will be used for imposing essential boundary conditions using the Lagrange Multipliers.

Now that we have a discretization for Lagrange Multipliers, we can apply it to the two last terms of Equation 2.51. Let us begin with the term containing the prescribed displacements:

$$\int_\Gamma \delta\boldsymbol{\lambda}\left[\mathbf{u} - \bar{\mathbf{u}}\right]\, d\Gamma = \int_\Gamma \left[\mathbf{N}\delta\boldsymbol{\Lambda}\right]^T\left[\boldsymbol{\Phi}\mathbf{U} - \bar{\mathbf{u}}\right]\, d\Gamma = \delta\boldsymbol{\Lambda}^T \int_\Gamma \mathbf{N}^T\boldsymbol{\Phi}\, d\Gamma - \delta\boldsymbol{\Lambda}^T \int_\Gamma \mathbf{N}^T\bar{\mathbf{u}}\, d\Gamma. \tag{2.65}$$

The integral in the first term of Equation 2.65 is denoted by $\mathbf{G}$ and the integral in the second term is denoted by $\mathbf{Q}$. Similarly to the global stiffness matrix, $\mathbf{G}$ is composed by n_b sub-matrix with size 2×2. The vector $\boldsymbol{\Lambda}$ and $\delta\boldsymbol{\Lambda}$ are constant nodal values relative to Lagrange Multipliers. We have:

$$\begin{aligned} \boldsymbol{\Lambda} &= \left\{\lambda_1^1, \lambda_2^1, \dots, \lambda_1^{n_b}, \lambda_2^{n_b}\right\}^T \\ \delta\boldsymbol{\Lambda} &= \left\{\delta\lambda_1^1, \delta\lambda_2^1, \dots, \delta\lambda_1^{n_b}, \delta\lambda_2^{n_b}\right\}^T, \end{aligned} \tag{2.66}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{11} & \dots & \mathbf{G}_{1n_b} \\ \vdots & \ddots & \vdots \\ \mathbf{G}_{n1} & \dots & \mathbf{G}_{nn_b} \end{bmatrix} \ , \ \mathbf{G_{ij}} = \int_\Gamma \mathbf{N}^{iT} \boldsymbol{\phi}^j \ d\Gamma, \tag{2.67}$$

$$\mathbf{Q} = \begin{Bmatrix} q_1^1 \\ q_2^1 \\ \vdots \\ q_1^{nb} \\ q_2^{nb} \end{Bmatrix} \ , \ \mathbf{q}^i = -\int_\Gamma \mathbf{N}^{iT} \bar{\mathbf{u}} \ d\Gamma. \tag{2.68}$$

The last term of Equation 2.51 is discretized in the same manner:

$$\int_\Gamma \delta \mathbf{u}^T \boldsymbol{\lambda} \ d\Gamma = \int_\Gamma \left[\boldsymbol{\Phi} \delta \mathbf{U}\right]^T \mathbf{N} \boldsymbol{\Lambda} \ d\Gamma = \delta \mathbf{U}^T \int_\Gamma \boldsymbol{\Phi}^T \mathbf{N} \ d\Gamma \ \boldsymbol{\Lambda}. \tag{2.69}$$

The integral of Equation 2.69 is the $\mathbf{G}$ matrix previously defined. With this, the last two terms of Equation 2.51 are written as:

$$\int_\Gamma \delta \boldsymbol{\lambda} \left[\mathbf{u} - \bar{\mathbf{u}}\right] \ d\Gamma = \delta \boldsymbol{\Lambda}^T \mathbf{G} \mathbf{U} + \delta \boldsymbol{\Lambda} \mathbf{Q}, \tag{2.70}$$

and

$$\int_\Gamma \delta \mathbf{u}^T \boldsymbol{\lambda} \ d\Gamma = \delta \mathbf{U}^T \mathbf{G} \boldsymbol{\Lambda}. \tag{2.71}$$

Finally, the discretized version of Equation 2.51 is:

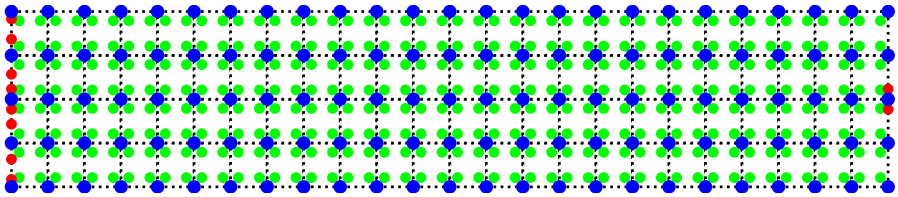
$$\begin{aligned} \delta \mathbf{U}^T \mathbf{K_g} \mathbf{U} + \delta \mathbf{U}^T \mathbf{F_b} + \delta \mathbf{U}^T \mathbf{F_t} + \delta \boldsymbol{\Lambda}^T \mathbf{G} \mathbf{U} + \delta \boldsymbol{\Lambda} \mathbf{Q} + \delta \mathbf{U}^T \mathbf{G} \boldsymbol{\Lambda} = 0 \\ \left[\delta \mathbf{U}\right]^T \left[\mathbf{K_g} \mathbf{U} + \mathbf{F} + \mathbf{G} \boldsymbol{\Lambda}\right] + \left[\delta \boldsymbol{\Lambda}\right]^T \left[\mathbf{G} \mathbf{U} + \boldsymbol{\Lambda} \mathbf{Q}\right] = 0, \end{aligned} \tag{2.72}$$

where the body and traction forces $\mathbf{F_b}$ and $\mathbf{F_t}$ are grouped in the global force vector $\mathbf{F}$. As the terms $\delta\mathbf{U}$ and $\delta\mathbf{\Lambda}$ can be chosen arbitrary (OVERVELDE, 2012), the above equations reduce to the following linear system:

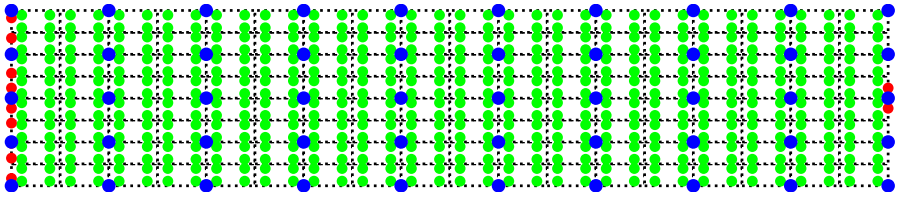
$$\begin{bmatrix} \mathbf{K_g} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{U} \\ \mathbf{\Lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F} \\ \mathbf{Q} \end{Bmatrix}. \tag{2.73}$$

The integration over the domain Ω and the boundary Γ are performed using the well-known Gaussian Quadrature. The Gauss points are distributed over the domain Ω and they are organized inside integration cells, as shown in Figure 2.10. Although the integration cells constitute a background mesh, its complexity is considerably inferior compared to a typicall FEM mesh (BELYTSCHKO ET AL., 1994), (OVERVELDE, 2012). In fact, the background mesh is not structured and it can be created independently from the discretization nodes, as illustrated in Figure 2.10b. Moreover, randomly spaced nodes can also be used instead of regularly distributed nodes.

It can also be seen in Figure 2.10 the boundary Gauss points for integration, represented in red. These points are used to perform the integration of Equations 2.70 and 2.71. Figure 2.11 shows in more detail the spatial distribution of Gauss points inside an integration cell. The number of Gauss points per cell is, of course, an arbitrary choice: too few points can lead to imprecision during the integration process and a large number of points may drastically increase the computational time to evaluate the integrals.



(a) Coincident



(b) Non-coincident

Figure 2.10: Typical EFG discretization. Nodes in blue are nodes and green dots are Gauss points organized inside the cells marked with dotted line. The red dots are boundary Gauss points. Images a and b show that discretization nodes and background mesh can be created independently.

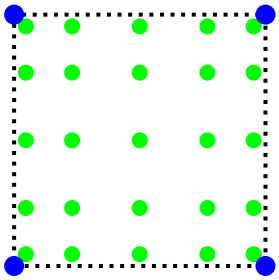


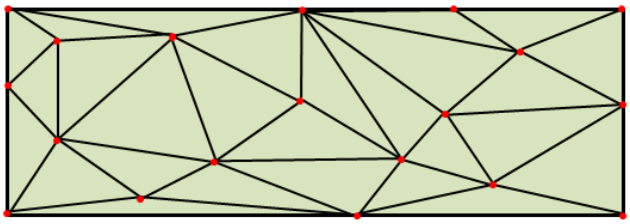
Figure 2.11: Integration cell with 5×5 Gauss points, represented in green. The blue points are the discretization nodes. The dotted lines illustrate borders of the integration cell.

A broad EFG algorithm is given in Figure 2.13. The assembly of the global vectors and matrix is a substantial portion of the EFG method. An algorithm for the calculation of the local stiffness matrix and for assembling the global matrix is given in Figure 2.14. The algorithm consists in a loop over all the integration point; this is done with a loop over the integration cell and, for each integration cell, a second loop over its Gauss points. As represented in Figure 2.14, n_c is the total number of cells and n_{GP} is the number of Gauss points per cell.

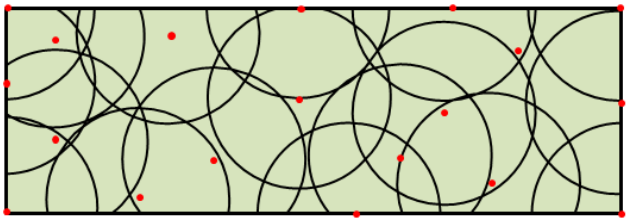
Still in the algorithm of Figure 2.13, the shape function of a Gauss point g is evaluated at the coordinates of each node i inside the influence domain of g. The matrix $\mathbf{B}$ can hence be obtained and the integration is done using the Gauss Quadrature rule. The resulting of the integration is than added to the global stiffness matrix using a localization vector, given in Equation 2.74, that states the position of each DOF in the global stiffness matrix. The order global vectors and matrix – $\mathbf{G}$, $\mathbf{Q}$ and $\mathbf{F}$ – are obtained using the same procedure.

$$\mathbf{loc} = [2n - 1 \ , \ 2n] \, . \tag{2.74}$$

A representation of the difference between FEM and EFG discretization is shown in Figure 2.12. The triangular elements of FEM mesh create a structured mesh through nodal connections. On the other hands, the EFG discretization does not possess an explicit connection between nodes, and some regions of the domain can belong to the influence domain of more than one node.



(a) Representation of a FEM discretization with triangular elements.



(b) Representation of a meshless discretization with circular influence domain.

Figure 2.12: A structured mesh of triangular elements typically used in FEM is shown in (a). A meshless discretization is represented in (b).

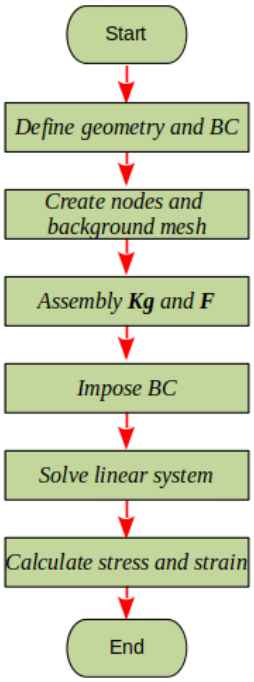


Figure 2.13: The general EFG algorithm.

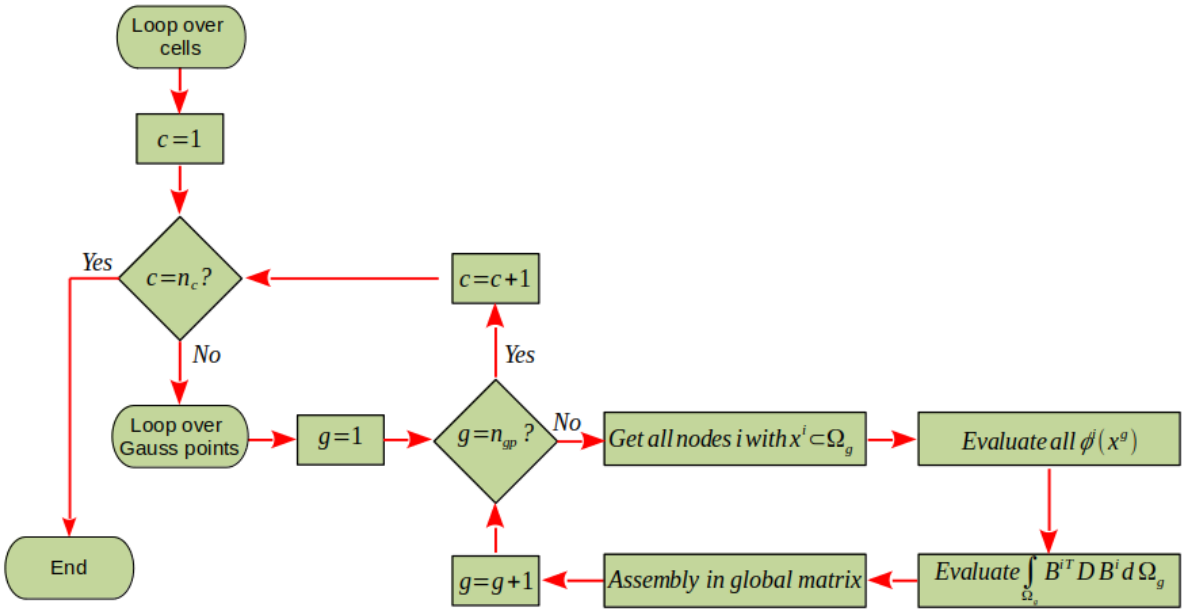


Figure 2.14: The algorithm to assembly the $\mathbf{K}_{\mathrm{g}}$ matrix. Through a loop over the n_c integration cells and over all the Gauss points n_{GP} of each cell, the shape functions of nodes i inside the influence domain of a quadrature point g are evaluated and the integrals are calculated using the Gauss quadrature method.

2.4 Shape Functions

Among several methods to obtain shape functions, in this work two are considered: the Moving Least-Squares – MLS – and the Radial Point Interpolation Method – RPIM – shape functions. In fact, in its first version, the EFG method were formulated with MLS shape functions (BELYTSCHKO ET AL., 1994). Although the promising results, one relevant drawback of using MLS is the difficulty to impose essential boundary conditions, as will be explained further in this subsection. To overcome this difficulty, new developments in the EFG method propose the use of RPIM shape functions, notably LIU (2003). With this class of shape functions, one can easily impose Dirichlet boundary conditions, in the same way of FEM.

The MLS, used in the first version of the EFG method (BELYTSCHKO ET AL., 1994), does not present the Kronecker delta property, that is, the approximated value of a function at node x^i is different from the nodal value used to construct the approximation, as stated in Equation 2.75:

$$\hat{u}(\mathbf{x^i}) \neq \mathbf{u}^i. \tag{2.75}$$

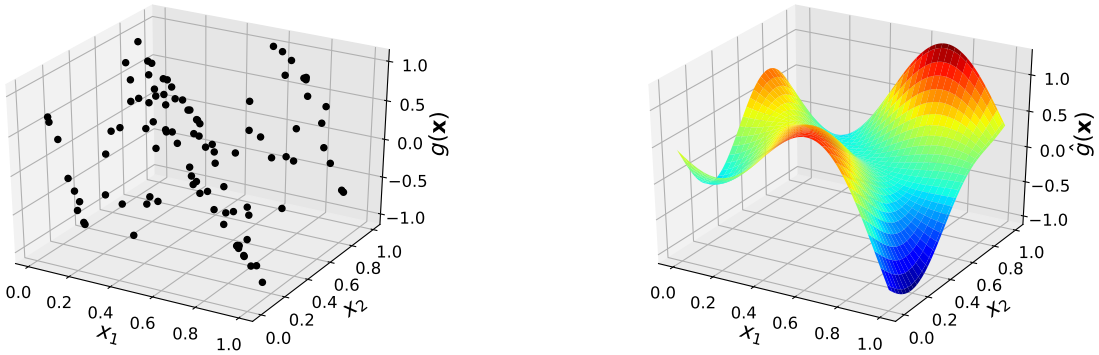
Because of the absence of Kronecker delta property, the essential boundary conditions need to be imposed using Lagrange Multipliers to ensure numerical precision. An alternative method to construct smooth shape functions is the RPIM, in which the Kronecker delta property does exist. Thus, the essential boundary conditions can be imposed directly in the global stiffness matrix, like in FEM, preventing the use of Lagrange Multipliers.

Further in this section, the procedure to obtain both MLS and RPIM shape function is detailed and some examples are given. The main differences between these methods are also highlighted.

2.4.1 Moving Least-Squares shape functions

The Moving Least-Squares were originally developed to fit a continuous smooth function through a set of scattered points. An interesting application is reconstructing surfaces from a set of points in computer graphics (AMIRFAKHRIAN, 2013), (NEALEN, 2004).

Given a set of n points in $\mathbb{R}^2$ with coordinates $\mathbf{x}^i$ and nodal values $\mathbf{G} = g^i$, $i = 1, \ldots, n$, we want a function $\hat{g}(\mathbf{x})$ that approximates the nodal values $\mathbf{G}$ in the sense of least-squares. This idea is illustrated in Figure 2.15, where a set of scattered points represented in black are approximated by a surface obtained by approximating the scattered points.



(a) Scattered points

(b) Approximant surface

Figure 2.15: Illustration of the basic idea of MLS in $\mathbb{R}^2$. The surface in (b) represents the function $\hat{g}(\mathbf{x})$ that approximates the scattered points shown in black in (a).

The objective here is to obtain the shape functions $\phi^k(\mathbf{x})$ to be used in Equations 2.40 and 2.41. To do so, we will consider that the approximation $\hat{g}(\mathbf{x})$ is a combination of a monomials basis $\mathbf{p}(\mathbf{x})$ and a vector of coefficient $\mathbf{a}(\mathbf{x})$ to be determined. In fact, the shape functions $\phi^k(\mathbf{x})$ are obtained when the coefficient vector $\mathbf{a}(\mathbf{x})$ is determined. We can write

$$\hat{g}(\mathbf{x}) = \sum_{l=1}^{m} p_l(\mathbf{x}) a_l(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}). \tag{2.76}$$

In the summation of Equation 2.76, m is the number of monomials present in the monomial basis $\mathbf{p}(\mathbf{x})$. A monomial basis is a combination of monomials, and the most common basis are:

$$\begin{aligned}
\mathbf{p}(\mathbf{x}) &= [1 \;\; x_1]^T && 1D\ ,\ m=2 \\
\mathbf{p}(\mathbf{x}) &= [1 \;\; x_1 \;\; x_1^2]^T && 1D\ ,\ m=3 \\
\mathbf{p}(\mathbf{x}) &= [1 \;\; x_1 \;\; x_2]^T && 2D\ ,\ m=3 \\
\mathbf{p}(\mathbf{x}) &= [1 \;\; x_1 \;\; x_2 \;\; x_1^2 \;\; x_2^2 \;\; x_1x_2]^T && 2D\ ,\ m=6.
\end{aligned} \tag{2.77}$$

The coefficients vector $\mathbf{a}(\mathbf{x})$ is written as:

$$\mathbf{a}(\mathbf{x}) = \left[a^1(\mathbf{x}),\ \ldots\ ,a^m(\mathbf{x})\right]^T. \tag{2.78}$$

In this work the 2D monomial basis with $m = 6$ is chosen, which is called quadratic monomial basis. The coefficients $\mathbf{a}(\mathbf{x})$ can be determined by minimizing the L_2 weighted norm given in Equation 2.79. The L_2 norm can be seen as an error between the true nodal value and the approximated values:

$$L_2 = \sum_{i=1}^{k} W\left(\left\|\mathbf{x}-\mathbf{x}^i\right\|,d\right)\left[\mathbf{p}^T(\mathbf{x}^i)\mathbf{a}(\mathbf{x}) - g^i\right]^2. \tag{2.79}$$

In Equation 2.79 the summation is over all nodes k with coordinates $\mathbf{x}^i$ and $\mathbf{x}$ are the coordinates of an arbitrary point where we want the approximation. The function $W(\mathbf{x}-\mathbf{x}^I,d)$ is called weight function and its arguments $\left\|\mathbf{x}-\mathbf{x}^i\right\|$ are the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}^i$ and d is a parameter relative to the influence domain size of a node, discussed in the beginning of Section 2.3.

In MLS approximations, the weight function must satisfy the following conditions:

i The unity condition. This property assures that the weight function integral is one, that is:

$$\int_\Omega W\left(\left\|\mathbf{x}-\mathbf{x}^i\right\|,d\right)d\mathbf{x} = 1. \tag{2.80}$$

ii The compact condition. This is to state that only for coordinates $\mathbf{x}$ inside the influence domain

of node I the value of scalar field u^I will influence the approximation of $u(\mathbf{x})$. In equation form:

$$W\left(\left\|\mathbf{x}-\mathbf{x}^i\right\|,d\right)=0 \quad outside \;\; \Omega^I. \tag{2.81}$$

iii The delta function condition. This property states that the closer a point of coordinates $\mathbf{x}$ is from the node of interest I, the higher will be the value of the weight function.

$$\lim_{d\to 0} W\left(\left\|\mathbf{x}-\mathbf{x}^i\right\|,d\right)=\delta(\mathbf{x}-\mathbf{x}^I). \tag{2.82}$$

A relatively simple function which satisfies these three conditions is the cubic spline shown in Equation 2.83. This type of weight function is widely used in MLS methods such as (NGUYEN ET AL., 2008) and (GINGOLD AND MONAGHAN, 1982).

$$W(\alpha)=\begin{cases}\frac{2}{3}-4\alpha^2+4\alpha^3 & \text{if } 0\leq\alpha\leq 1/2\\ \frac{4}{3}-4\alpha+4\alpha^2-\frac{4}{3}\alpha^3 & \text{if } 1/2\leq\alpha\leq 1\\ 0 & \text{otherwise,}\end{cases} \tag{2.83}$$

with derivatives

$$W_{,\,\alpha}(\alpha)=\begin{cases}-8\alpha+12\alpha^2 & \text{if } 0\leq\alpha\leq 1/2\\ -4+8\alpha-4\alpha^2 & \text{if } 1/2\leq\alpha\leq 1\\ 0 & \text{otherwise.}\end{cases} \tag{2.84}$$

Here α is calculated using the influence domain size and the distance between the points $\mathbf{x}$ and $\mathbf{x}^i$. Considering circular influence domain, the relation for α is:

$$\alpha=\frac{\left\|\mathbf{x}-\mathbf{x}^i\right\|}{d}. \tag{2.85}$$

Thus, the weight function and its derivatives can be written as:

$$W\left(\left\|\mathbf{x}-\mathbf{x}^{i}\right\|,d\right)=W\left(\alpha\right), \tag{2.86}$$

where the derivatives of α can be easily obtained by deriving the Euclidean norm that defines α:

$$W_{,\,j}\left(\left\|\mathbf{x}-\mathbf{x}^{i}\right\|,d\right)=W_{,\,\alpha}\left(\alpha\right)\alpha_{,\,j}. \tag{2.87}$$

As previously mentioned, rectangular influence domains can also be used. For this case, the weight function is written in terms of α_1 and α_2:

$$W\left(\alpha\right)=W\left(\alpha_2\right)W\left(\alpha_1\right). \tag{2.88}$$

The parameters α_1 and α_2 are also calculated using the distance between $\mathbf{x}$ and $\mathbf{x}^i$. We consider however the distance in x_1 and x_2 directions separately, as follows:

$$\alpha_1=\frac{\left\|\mathbf{x}_1-x_1^i\right\|}{d_1\ d}, \tag{2.89}$$

and

$$\alpha_2=\frac{\left\|\mathbf{x}_2-x_2^i\right\|}{d_2\ d}. \tag{2.90}$$

Thus, for rectangular influence domains, the weight function and its derivatives are written as

$$W\left(\left\|\mathbf{x}-\mathbf{x}^{i}\right\|,d\right)=W\left(\alpha_1,d_1\ d\right)W\left(\alpha_2,d_2\ d\right). \tag{2.91}$$

The parameters d, d_1 and d_2 are related to the size of the influence domain. For static analysis, the parameter d typically assumes values between 2.0 and 4.0 (BELYTSCHKO ET AL., 1994). Concerning the parameters d_1 and d_2, they are chosen to correspond to the distance between nodes in x_1 and x_2 directions, respectively. The impact of these parameters is studied later in this section.

Back to Equation 2.79, once the weight function is chosen, the minimization problem can be solved by making the first derivative of the L_2 norm equals to zero

$$\frac{\partial L_2}{\partial \mathbf{a}} = \mathbf{0}. \tag{2.92}$$

Which leads to the following set of linear equations:

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{u}. \tag{2.93}$$

The terms $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are:

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^{k} W(x - x^I)p(x^I)p^T(x^I), \tag{2.94}$$

and

$$\mathbf{B}(\mathbf{x}) = \sum_{I=1}^{k} W(x - x^I)p(x^I). \tag{2.95}$$

Hence, the coefficient vector $\mathbf{a}(\mathbf{x})$ can be written as

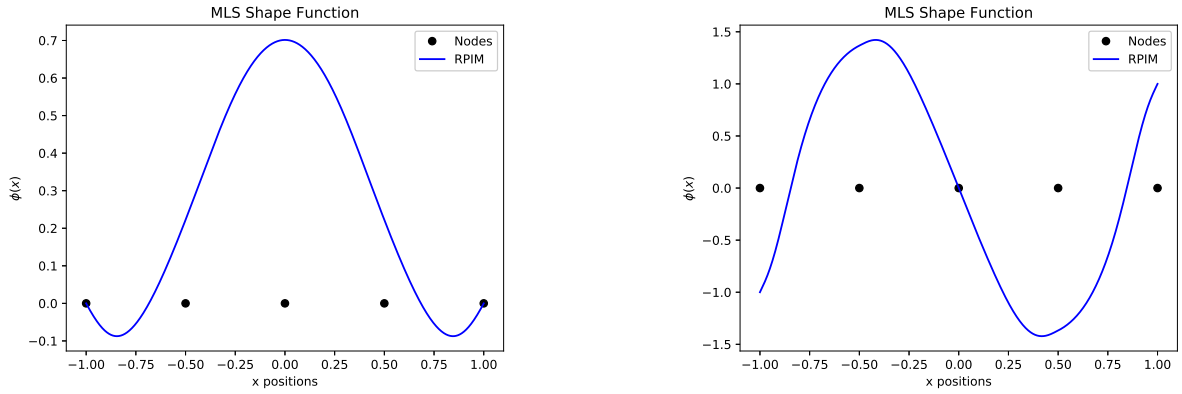
$$\mathbf{a}(\mathbf{x}) = \mathbf{A}(\mathbf{x})^{-1}\mathbf{B}(\mathbf{x})\mathbf{u}, \tag{2.96}$$

and the approximation of Equation 2.76 becomes:

$$u^h(\mathbf{x}) = \mathbf{p}(\mathbf{x})\mathbf{A}(\mathbf{x})^{-1}\mathbf{B}(\mathbf{x})\mathbf{u} = \sum_{I=1}^{k} \phi^I u^I. \tag{2.97}$$

The function ϕ^I is called the shape function at node I.

An example of one-dimensional MLS shape function for node $x_1 = 0$ is given in Figure 2.16. The domain is $[-1,1]$ and is discretized with 5 nodes, an influence domain with radius of 1 was used. One may note in Figure 2.16 that the Kronecker delta property is not satisfied at $x_1 = 0$ once $\phi\left(x_1 = 0\right) \neq 1$.

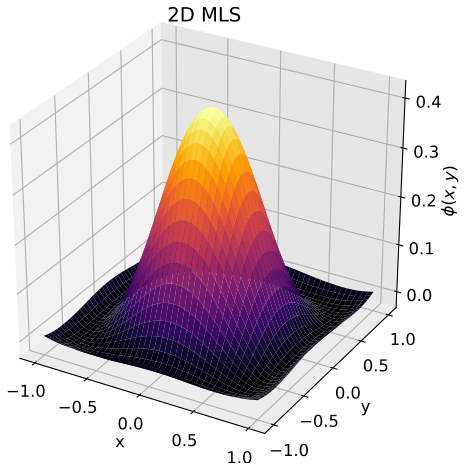


(a) Shape function

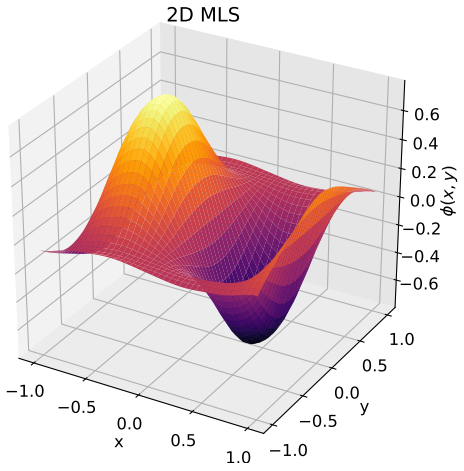
(b) Derivatives

Figure 2.16: Example of one-dimensional MLS shape function *(a)* and its derivative *b*. The shape function is relative to node $x_1 = 0$.

A two-dimensional MLS shape function is illustrated in Figure 2.17, for a domain with size $[-1,1] \times [-1,1]$ and discretized with 5 nodes equally spaced in both x_1 and x_2 directions.



(a) Shape functions



(b) Derivatives

Figure 2.17: Example of a two-dimensional MLS shape function *(a)* and its derivative in x_1 direction *(b)*. The shape function is relative to node $(x_1,x_2) = (0,0)$.

2.4.2 Radial Point Interpolation Method shape functions

As previously mentioned, the MLS shape function does not hold the Kronecker delta criterion, and for this reason the essential boundary condition cannot be easily imposed such as in FEM. In this way, strategies like the Penalty Method or Lagrange Multipliers need be used to impose the Dirichlet boundary conditions.

An alternative class of shape function that does hold the Kronecker criterion, and thus simplifies the imposition of boundary conditions, is the Radial Point Interpolation Method – RPIM. The use of RPIM as shape functions in meshless methods was proposed by (LIU, 2003) as a modification of the original EFG method. In this section we will discuss the basic theory of the RPIM used to construct the shape functions.

With the RPIM, the displacement field is approximated at a coordinate $\mathbf{x}$ as follows (MOLLON, 2016):

$$u(\mathbf{x},\mathbf{x}^I) = \sum_{i=1}^{n} R_i(\mathbf{x})a_i(\mathbf{x}^I) + \sum_{j=1}^{m} p_j(\mathbf{x})b_j(\mathbf{x}^I), \tag{2.98}$$

where $R_i(\mathbf{x})$ is a radial basis function, $p_j(\mathbf{x})$ is a monomial basis, the same one used in the MLS method, and m is the number of monomials of $p_j(\mathbf{x})$. The coefficients a_i and b_j are yet to be determined. The vector $\mathbf{x^I}$ contains the coordinates of nodes inside the influence domain of a point with coordinates $\mathbf{x}$.

There are several options for the radial functions R_i such as multi-quadratic, Gaussian, logarithmic, thin plate spline, among others. In this work we will use the following Gaussian radial function:

$$R_i(\mathbf{x}) = \begin{cases} \exp\left(-\frac{cd_i}{D_i}\right)^2 & \text{, if } d_i \leq D_i \\ 0 & \text{, otherwise.} \end{cases} \tag{2.99}$$

The c coefficient is called a shape parameter and its influence on the shape function construction will be investigated latter in this subsection. The value D_i is the influence domain radius of an node i and, finally, d_i is the Euclidean distance from the coordinates $\mathbf{x}$ to the coordinates $\mathbf{x}_i$ of an node i, as shown below:

$$d_i = \|\mathbf{x} - \mathbf{x}_i\| . \tag{2.100}$$

We can use matrix form for a more compact notation which is also more suited to computational implementation:

$$u(\mathbf{x}) = \mathbf{R}^T(\mathbf{x})\mathbf{a} + \mathbf{P}^T(\mathbf{x})\mathbf{b} = \begin{bmatrix} \mathbf{R}^T(\mathbf{x}) & \mathbf{P}^T(\mathbf{x}) \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \end{Bmatrix}, \tag{2.101}$$

where:

$$\mathbf{a} = [a_1, a_2, \ldots, a_n]^T, \tag{2.102}$$

$$\mathbf{b} = [b_1, b_2, \ldots, b_m]^T, \tag{2.103}$$

$$\mathbf{R}(\mathbf{x}) = [R_1(\mathbf{x}), R_2(\mathbf{x}), \ldots, R_n(\mathbf{x})]^T, \tag{2.104}$$

and

$$\mathbf{P}(\mathbf{x}) = [P_1(\mathbf{x}), P_2(\mathbf{x}), \ldots, P_m(\mathbf{x})]^T. \tag{2.105}$$

The vectors of coefficients a and b can be determined by stating that the Equation 2.101 to be true for all nodes n. This will produce the following set of equations:

$$\mathbf{u_s} = \mathbf{R_q}\mathbf{a} + \mathbf{P_m}\mathbf{b}. \tag{2.106}$$

Here $\mathbf{u_s}$ is a vector containing the displacement value for all the domain's nodes. The linear system of 2.106 cannot be solved because it is a set of n equations and $n + m$ variables. An additional constraint must be imposed in order to solve the equation system:

$$\mathbf{P}_m^T\mathbf{a} = \mathbf{0}. \tag{2.107}$$

Combining equations 2.106 and 2.107 leads to the following equation system:

$$\begin{bmatrix} \mathbf{R_q} & \mathbf{P_m} \\ \mathbf{P_m} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \end{Bmatrix} = \begin{Bmatrix} \mathbf{u}_s \\ \mathbf{0} \end{Bmatrix}. \tag{2.108}$$

Thus, the coefficients $\mathbf{a}$ and $\mathbf{b}$ can be determined:

$$\begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \end{Bmatrix} = \begin{bmatrix} \mathbf{R_q} & \mathbf{P_m} \\ \mathbf{P_m} & \mathbf{0} \end{bmatrix}^{-1} \begin{Bmatrix} \mathbf{u}_s \\ \mathbf{0} \end{Bmatrix}. \tag{2.109}$$

Using a more compact notation, where the coefficient matrix of Equation 2.108 is denoted $\mathbf{G}$:

$$\begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \end{Bmatrix} = \mathbf{G}^{-1} \begin{Bmatrix} \mathbf{u}_s \\ \mathbf{0} \end{Bmatrix}, \tag{2.110}$$

where $\mathbf{G}$ is the matrix containing $\mathbf{R_q}$ and $\mathbf{P_m}$. Back to the equation 2.101, with the coefficient $\mathbf{a}$ and $\mathbf{b}$ now determined, the displacement field is interpolated as:

$$\mathbf{u}(\mathbf{x}) = \begin{bmatrix} \mathbf{R}^T(\mathbf{x}) & \mathbf{P}^T(\mathbf{x}) \end{bmatrix} \mathbf{G}^{-1} \begin{Bmatrix} \mathbf{u}_s \\ \mathbf{0} \end{Bmatrix}. \tag{2.111}$$

By denoting the shape functions by ϕ as in the MLS subsection, we have:

$$u(\mathbf{x}) = \bar{\boldsymbol{\phi}}(\mathbf{x}) \begin{Bmatrix} \mathbf{u}_s \\ \mathbf{0} \end{Bmatrix}, \tag{2.112}$$

and the shape functions can then be obtained with the following relation:

$$\bar{\boldsymbol{\phi}}(\mathbf{x}) = \begin{bmatrix} \mathbf{R}^T(\mathbf{x}) & \mathbf{P}^T(\mathbf{x}) \end{bmatrix} \mathbf{G}^{-1}, \tag{2.113}$$

with:

$$\bar{\boldsymbol{\phi}}(\mathbf{x}) = \begin{bmatrix} \phi^1(\mathbf{x}) & \phi^2(\mathbf{x}) & \dots & \phi^n(\mathbf{x}) & \dots & \phi^{n+m}(\mathbf{x}) \end{bmatrix}. \tag{2.114}$$

The vector $\bar{\boldsymbol{\phi}}(\mathbf{x})$ contains m extra terms that are relative to the monomial basis terms of the linear system of Equation 2.109. Hence, the shape functions corresponding to the nodal displacements are:

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} \phi^1(\mathbf{x}) & \phi^2(\mathbf{x}) & \dots & \phi^n(\mathbf{x}) \end{bmatrix}. \tag{2.115}$$

The shape functions derivatives along the directions $j = 1,2$ are obtained using both radial and monomial basis vector derivatives, as follows:

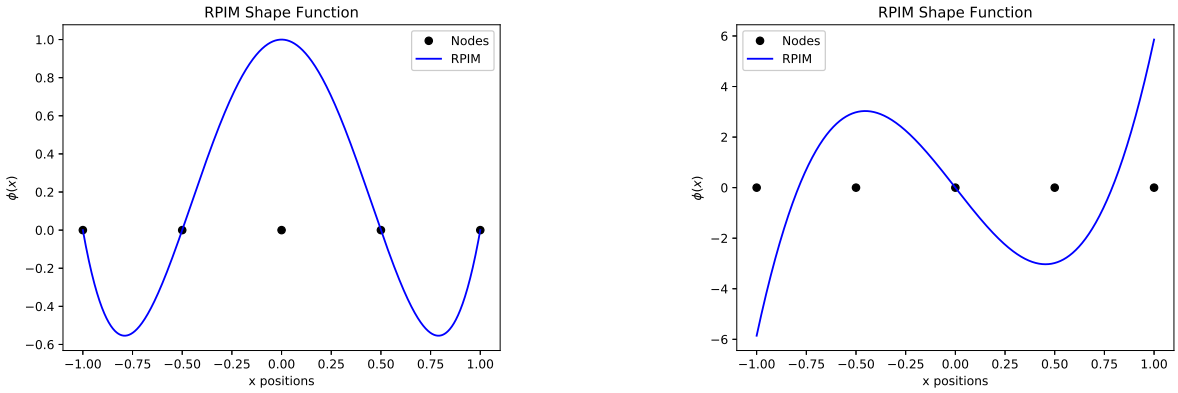
$$\bar{\boldsymbol{\phi}}_{,j}(\mathbf{x}) = \begin{bmatrix} \mathbf{R}^T_{,j}(\mathbf{x}) & \mathbf{P}^T_{,j}(\mathbf{x}) \end{bmatrix} \mathbf{G}^{-1}. \tag{2.116}$$

Similarly to the shape function, the derivatives corresponding to nodal displacement are:

$$\boldsymbol{\phi}_{,j}(\mathbf{x}) = \begin{bmatrix} \phi^1_{,j}(\mathbf{x}) & \phi^2_{,j}(\mathbf{x}) & \dots & \phi^n_{,j}(\mathbf{x}) \end{bmatrix}. \tag{2.117}$$

To differentiate the shape functions ϕ obtained with the MLS method from those constructed using RPIM, a subscript M for MLS and R for RPIM will be used. For example, ϕ^i_M and ϕ^i_R stands for the shape function of node i using MLS and RPIM methods, respectively.

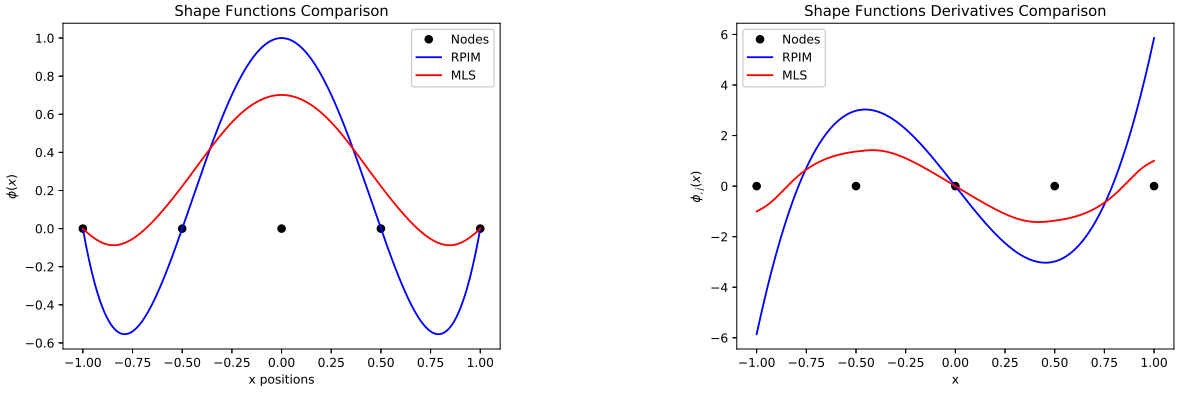
A typical one-dimensional RPIM constructed with the Gaussian radial function is illustrated in Figure 2.18. The domain $[-1,1]$ is discretized with five nodes equally spaced nodes and the shape function is relative to the node $x = 0$. The influence domain radius is set to 1 and the shape parameter of the radial function c is set to 0.3. The RPIM derivatives in x_1 direction is also shown in Figure 2.18b.



(a) Shape function (b) Derivatives

Figure 2.18: Example of RPIM shape function *(a)* and its derivatives *(b)*for node $x = 0$. In *(a)* we can note that the Kronecker delta property is respected.

A comparison between shape functions obtained with MLS and RPIM methods is given in Figure 2.19. One may note that RPIM shape function posses the Kronecker delta criterion while the MLS does not, that is, $\phi_M(x_1 = 0) \neq 1$ and $\phi_R(x_1 = 0) = 1$. The RPIM shape function is the same of Figure 2.18 and the MLS shape function is the same of Figure 2.16. One may also note in Figure 2.19a that RPIM shape function presents more variation over the domain than the MLS.



(a) Shape fucntions (b) Derivatives

Figure 2.19: Comparison between MLS and RPIM. One may note that the MLS does not possess the Kronecker delta criterion while the RPIM shape function does have this property.

In order to study the impact of the shape parameter c of Gaussian radial functions, the Figure 2.20 presents a comparison for different values of c. One may note that greater c values affects the decay rate of the oscillations behind the first dominant peak (LIU, 2003). In the EFG with RPIM presented by (MOLLON, 2016) the value c is usually set to be 3, and this choice is kept in this work.

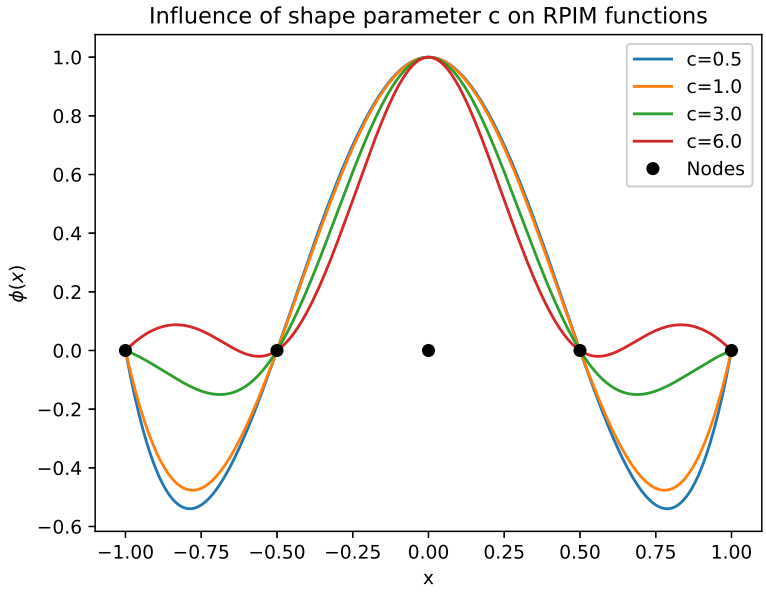
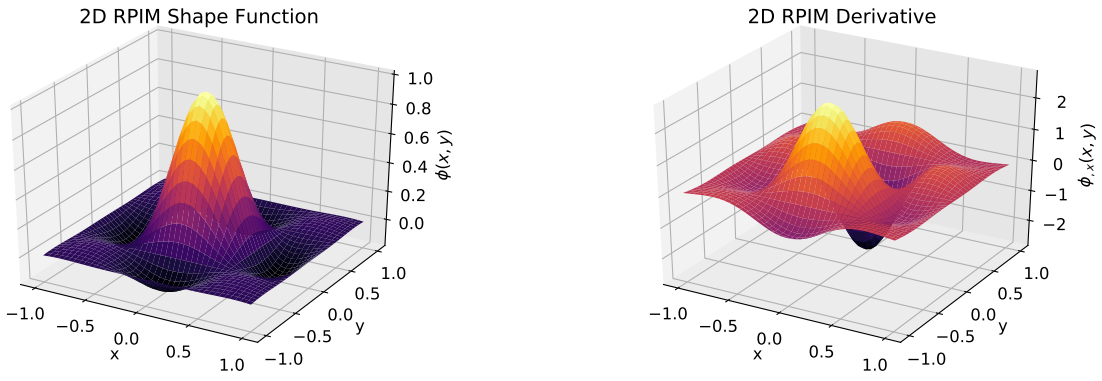


Figure 2.20: Influence of the c parameter on the RPIM shape function results.

To illustrate the RPIM for 2D cases, the Figure 2.21 shows RPIM shape function and its derivatives for a squared domain of $(x,y) \in [-1,1] \times [-1,1]$, discretized with 5×5 equally spaced nodes. A Gaussian radial function with $c = 3$ is used and the influence domain size is set to $d = 1$.



(a) Shape function

(b) Derivatives

Figure 2.21: Example of a 2D RPIM shape function and its derivative.

2.5 2D Nonlinear elasticity

As the geometrically nonlinear problems undergo large displacement, rotation and strains, the formulation used for linear cases, which assumes that there is no consequent difference between the deformed and undeformed shapes, cannot be used. Although the kinematics and constitutive law – presented in subsections 2.1.4 – remain the same, a new formulation for strains must be obtained in order to correctly define their dependency with large deformations.

2.5.1 Strain and Stress

Either deformed and undeformed geometries can be used to define a way of measuring the strain under large deformations. In this work, the undeformed initial domain $\mathbb{S}_0$ is used to define strain, which is called a Lagrangian approach. In this way, the Equation 2.8 – which relates the strain and displacement – can be rewritten using a subscript 0 to indicate that the displacement gradient is related to the undeformed referential:

$$\mathbf{E} = \frac{1}{2}\left(\nabla_0 \mathbf{u} + \nabla_0 \mathbf{u}^T + \nabla_0 \mathbf{u}^T \nabla_0 \mathbf{u}\right). \tag{2.118}$$

As large deflection is expected, the quadratic term of Equation 2.8 is not anymore negligible. Using the Voigt notation, the Lagrangian strain can be written in terms of displacements:

$$\mathbf{E} = \begin{Bmatrix} E_{11} \\ E_{22} \\ 2E_{12} \end{Bmatrix} = \begin{Bmatrix} u_{1,1} + \frac{1}{2}\left(u_{1,1}u_{1,1} + u_{2,1}u_{2,1}\right) \\ u_{2,2} + \frac{1}{2}\left(u_{1,2}u_{1,2} + u_{2,2}u_{2,2}\right) \\ u_{1,2} + u_{2,1} + u_{1,2}u_{1,1} + u_{2,1}u_{2,2} \end{Bmatrix}, \tag{2.119}$$

The conjugated stress pair of the Lagrangian strain is called Second Piola-Kirchhoff stress tensor and it is obtained through the constitutive matrix:

$$\mathbf{S} = \begin{Bmatrix} S_{11} \\ S_{22} \\ S_{12} \end{Bmatrix} = \mathbf{D} : \mathbf{E}. \tag{2.120}$$

As material nonlinearities are not considered in this work, the constitutive matrix $\mathbf{D}$ remains the same of Equation 2.12, which assumes a plane stress state. The Second Piola-Kirchhoff stress tensor both forces and area refer to undeformed configuration (KIM, 2015). The relation between the Second Piola-Kirchhoff and Cauchy – called *true* stress – is:

$$\mathbf{S} = J\mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T}, \tag{2.121}$$

where J is the determinant of $\mathbf{F}$, the gradient of the mapping function defined in 2.3, and $\boldsymbol{\sigma}$ is Cauchy stress, defined as:

$$\lim_{\Delta S \to 0} \frac{\Delta \mathbf{f}}{\Delta S} = \boldsymbol{\sigma} \cdot \mathbf{n}, \tag{2.122}$$

for an internal force vector $\mathbf{f}$ acting on a area ΔS, with unit normal vector $\mathbf{n}$, both referring to the final deformed domain $\mathbb{S}$. This is claimed to be a true measure of stress because it takes both force and area of the deformed domain. However, to have a conjugate pair in energy, the tensor $\mathbf{S}$ and $\mathbf{E}$

must be used together (KIM, 2015).

2.5.2 Equilibrium equations

Below, we present a simplified way of obtaining the nonlinear equilibrium equations by introducing the Green-Lagrange strain tensor. Next, a linearization technique is presented to solve the nonlinear equations with the Newton-Raphson method. The deductions here presented are extracted from KIM (2015), to which one may refer for more details.

We begin with the energy balance of a nonlinear elastic structure. We write the total potential energy of a system, as follows:

$$\begin{aligned}\Pi(\mathbf{u}) &= U(\mathbf{u}) - W(\mathbf{u}) \\ &= \frac{1}{2}\iint_{\mathbb{S}_0} \boldsymbol{\sigma}(\mathbf{u}) \colon \boldsymbol{\epsilon}(\mathbf{u})\, d\mathbb{S} - \iint_{\mathbb{S}_0} \mathbf{u}\cdot\mathbf{f}^b\, d\mathbb{S} - \int_{\Gamma_0} \mathbf{u}\cdot\mathbf{t} d\Gamma.\end{aligned} \tag{2.123}$$

We are now concerned with large displacements, thus the Lagrangian strain tensor given in Equation 2.8 is used in Equation 2.123, which becomes nonlinear with the displacement field $\mathbf{u}(\mathbf{x})$.

We know from the principle of minimum potential energy that the displacement field $\mathbf{u}(\mathbf{x})$ we are trying to determine is the one that minimizes the Equation 2.123. Knowing this, to find this displacement field, we can employ a perturbation method, as described below.

A perturbation in the displacement field in an arbitrary direction $\bar{\mathbf{u}}$ with size τ (OLIVER AND AGELET, 2017). The perturbation can be written as:

$$\mathbf{u}_\tau = \mathbf{u} + \tau\bar{\mathbf{u}}. \tag{2.124}$$

It is worth noting that the perturbation direction $\bar{\mathbf{u}}$ must not transgress the essential boundary conditions. In this way, the perturbed displacement $\mathbf{u}_\tau$ also belongs to the space of kinematically admissible configurations.

Here the concept of *functional first variation* needs to be introduced. We can understand the variational of a function as the *generalization of the concept of the differential of a function of one variable* (OF MATHEMATICS, 2011), which is also known as Gâteaux derivative (STOVER, 2021). In the context of this work, we will use the concept of functional first variational to obtain the potential energy variation in the direction of $\bar{\mathbf{u}}$. We can state that the first-variational of Π in the direction of $\bar{\mathbf{u}}$ is:

$$\bar{\Pi}\left(\mathbf{u},\bar{\mathbf{u}}\right) \equiv \frac{d}{d\tau}\Pi\left(\mathbf{u}+\tau\bar{\mathbf{u}}\right)\Big|_{\tau=0}. \tag{2.125}$$

Using the total potential energy of Equation 2.22, one can write:

$$\bar{\Pi}\left(\mathbf{u},\bar{\mathbf{u}}\right) = \iint_{\mathbb{S}^0} \frac{\partial W(\mathbf{E})}{\partial \mathbf{E}} : \bar{\mathbf{E}}\; d\mathbb{S} - \iint_{\mathbb{S}_0} \bar{\mathbf{u}}\cdot\mathbf{f}^b\; d\mathbb{S} - \int_{\Gamma_0} \bar{\mathbf{u}}\cdot\mathbf{t} d\Gamma, \tag{2.126}$$

where the chain rule is applied to differentiate the external work contribution. The $\bar{\mathbf{E}}$ portion is calculated using the definition of functional variation:

$$\begin{aligned}
\bar{\mathbf{E}}(\mathbf{u},\bar{\mathbf{u}}) &= \frac{d}{d\tau}\mathbf{E}\left(\mathbf{u}+\tau\bar{\mathbf{u}}\right)\Big|_{\tau=0} \\
&= \frac{1}{2}\left(\nabla_0\bar{\mathbf{u}} + \nabla_0\bar{\mathbf{u}}^T + \nabla_0\bar{\mathbf{u}}^T\nabla_0\mathbf{u} + \nabla_0\mathbf{u}^T\nabla_0\bar{\mathbf{u}}\right) \\
&= sym\left(\nabla_0\bar{\mathbf{u}}^T + \nabla_0\bar{\mathbf{u}}^T\nabla_0\mathbf{u}\right) \\
&= sym\left(\nabla_0\bar{\mathbf{u}}^T\mathbf{F}\right).
\end{aligned} \tag{2.127}$$

One can write the variational Equation 2.126 in a form similar to the linear problems:

$$a(\mathbf{u},\bar{\mathbf{u}}) = l(\bar{\mathbf{u}}), \tag{2.128}$$

where $a(\mathbf{u},\bar{\mathbf{u}})$ and $l(\bar{\mathbf{u}})$ are the strain energy and external load contribution to the structure's total potential energy, respectively. The strain energy term is:

$$a(\mathbf{u},\bar{\mathbf{u}}) = \iint_{\mathbb{S}^0} \mathbf{S}(\mathbf{u}) : \bar{\mathbf{E}} \, d\mathbb{S}^0, \tag{2.129}$$

where the term $\frac{\partial W(\mathbf{E})}{\partial \mathbf{E}}$ was replaced by $\mathbf{S}(\mathbf{u})$, which denotes the second Piola-Kirchhoff stress. The external load contribution is:

$$l(\bar{\mathbf{u}}) = \iint_{\mathbb{S}_0} \bar{\mathbf{u}} \cdot \mathbf{f}^b \, d\mathbb{S} - \int_{\Gamma_0} \bar{\mathbf{u}} \cdot \mathbf{t} \, d\Gamma. \tag{2.130}$$

As previously mentioned, body forces are not considered in this work thus the external load can be simplified to:

$$l(\bar{\mathbf{u}}) = \int_{\Gamma_0} \bar{\mathbf{u}} \cdot \mathbf{t} d\Gamma. \tag{2.131}$$

With the Equation 2.131 we have a formulation for the equilibrium equation which considers large displacements. In order to solve this equation, a linearization scheme is proposed so that a numerical method can be used to solve the nonlinear problem.

The numerical schemes used to solve nonlinear equations are generally iteratives methods that uses an initial guess to generate a sequence of improving approximate solutions (RUGGIERO AND LOPES, 1996). At each iteration a linearized equilibrium equation is solved and, until the convergence, a *residual* will exist once the minimum potential energy is not reached. At the correct solution, the residual will tend to zero. In this work the Newton-Raphson method will be used.

The above mentioned *residual*, is the difference between internal and external efforts, and it is defined as:

$$\mathbf{R} = a\left(\mathbf{u},\bar{\mathbf{u}}\right) - \ell\left(\bar{\mathbf{u}}\right). \tag{2.132}$$

The residual equation is nonlinear with respect to $\mathbf{u}$ and need hence to be linearized. The

linearization of a function $f(\mathbf{u})$ can be obtained with Taylor expansion, as follows:

$$f\left(\mathbf{u}\right) = f\left(\mathbf{u_0}\right) + \frac{\partial f\left(\mathbf{u_0}\right)}{\partial \mathbf{u}}\left(\mathbf{u} - \mathbf{u_0}\right) + O\left(\mathbf{u}^2\right). \tag{2.133}$$

Considering that the order two error $O\left(\mathbf{u}^2\right)$ can be neglected and knowing that we a looking for the zero of function f, we can write:

$$0 \approx f\left(\mathbf{u_0}\right) + \frac{\partial f\left(\mathbf{u_0}\right)}{\partial \mathbf{u}}\left(\mathbf{u} - \mathbf{u_0}\right). \tag{2.134}$$

The Equation 2.134 can be rewritten in a more convenient way using an iteration notation. Let k be used to denote the k-th iteration of the solution algorithm. Let also $\mathbf{u} - \mathbf{u_0}$ be defined as the change between two consecutive iterations, that is, $\mathbf{d}^k = \mathbf{u}^{k+1} - \mathbf{u}^k$. In this way, we can write a sequence of solutions, which is the widely known Newton-Raphson method:

$$\begin{aligned} \frac{\partial f\left(\mathbf{u}^k\right)}{\partial \mathbf{u}}\mathbf{d}^k &= -f(\mathbf{u}^k) \\ \mathbf{u}^{k+1} &= \mathbf{u}^k + \mathbf{d}^k. \end{aligned} \tag{2.135}$$

This linearization process can be applied to the residual $\mathbf{R}$ of Equation 2.132. One may note that the load portion $\ell\left(\bar{\mathbf{u}}\right)$ of Equation 2.132 does not depend on the displacement field and therefore there is no need to linearize this term. Thus, only the energy contribution $a(\mathbf{u},\bar{\mathbf{u}})$ needs linearization, which is given by:

$$L\left[a\left(\mathbf{u},\bar{\mathbf{u}}\right)\right] = \iint_{\mathbb{S}_0}\left[\boldsymbol{\Delta}\mathbf{S} : \bar{\mathbf{E}} + \mathbf{S} : \boldsymbol{\Delta}\bar{\mathbf{E}}\right] d\mathbb{S}, \tag{2.136}$$

where $\boldsymbol{\Delta}\mathbf{S}$ is the increment in stress, $\bar{\mathbf{E}}$ is the strain variation, $\mathbf{S}$ is the stress and $\boldsymbol{\Delta}\bar{\mathbf{E}}$ is the increment in the strain variation. The tensors $\boldsymbol{\Delta}\mathbf{S}$ and $\bar{\mathbf{E}}$ can be calculated with the following relations:

$$\boldsymbol{\Delta}\mathbf{S} = \mathbf{D} : \boldsymbol{\Delta}\mathbf{E}, \tag{2.137}$$

and

$$\boldsymbol{\Delta}\bar{\mathbf{E}} = sym\left(\nabla_0 \bar{\mathbf{u}}^T \nabla_0 \mathbf{u}\right). \tag{2.138}$$

Using the relations given by Equation 2.137, we can replace the terms $\boldsymbol{\Delta}\mathbf{S}$ and $\bar{\mathbf{E}}$ in Equation 2.136. This leads us to the following relation:

$$L\left[a\left(\mathbf{u},\bar{\mathbf{u}}\right)\right] = \iint_{\mathbb{S}_0} \left[\mathbf{D} : \boldsymbol{\Delta}\mathbf{E} : \bar{\mathbf{E}} + \mathbf{S} : \boldsymbol{\Delta}\bar{\mathbf{E}}\right] d\mathbb{S} \equiv a^*\left(\mathbf{u},\boldsymbol{\Delta}\mathbf{u},\bar{\mathbf{u}}\right), \tag{2.139}$$

which is the linearized form of $a(\mathbf{u},\bar{\mathbf{u}})$.

Using k to denote the iterations, we can write the residual Equation 2.132 in terms of incremental displacement and the linearized form of the energy:

$$\begin{gathered} a^*\left(\mathbf{u}^k,\boldsymbol{\Delta}\mathbf{u}^k,\bar{\mathbf{u}}\right) = \ell\left(\bar{\mathbf{u}}\right) - a\left(\mathbf{u}^k,\bar{\mathbf{u}}\right) \\ \mathbf{u}^{k+1} = \mathbf{u}^k + \boldsymbol{\Delta}\mathbf{u}^k. \end{gathered} \tag{2.140}$$

Is it possible to write the Equation 2.140 using matrix notation (LIU, 2003), (KIM, 2015), as follows:

$$\mathbf{K_t}\boldsymbol{\Delta}\mathbf{u} = \mathbf{R}, \tag{2.141}$$

where $\mathbf{K_t}$ is called tangent matrix, $\boldsymbol{\Delta}\mathbf{u}$ is an incremental in displacement and $\mathbf{R}$ is a residual array. The tangent matrix $\mathbf{K_t}$ and the residual vector $\mathbf{R}$ are obtained through the discretization of Equation 2.140, which is presented in the next two sections. With $\mathbf{K_t}$ and $\mathbf{R}$ we can obtain the $\boldsymbol{\Delta}\mathbf{u}$ for each

iteration until the correction solution for displacement field is reached.

As previously mentioned, the next two section are dedicated to present the FEM and EFG discretization of the nonlinear problem stated in Equation 2.140.

2.5.3 FEM Discretization

Here the FEM discretization is detailed. The discretization process is similar to the linear case, that is, the shape functions defined within an element and the nodal displacement are used to discretize the displacement field over the domain. This subsection is also based on (KIM, 2015).

In order to obtain the discretized form of the Equation 2.140 the Voigt notation presented in Subsection 2.1.4 is used. Using this notation, the strain and stress tensor are written as:

$$\mathbf{E} = \begin{Bmatrix} E_{11} \\ E_{22} \\ 2E_{12} \end{Bmatrix} = \begin{Bmatrix} u_{1,1} + \frac{1}{2}\left(u_{1,1}u_{1,1} + u_{2,1}u_{2,1}\right) \\ u_{2,2} + \frac{1}{2}\left(u_{1,2}u_{1,2} + u_{2,2}u_{2,2}\right) \\ u_{1,2} + u_{2,1} + u_{1,2}u_{1,1} + u_{2,1}u_{2,2} \end{Bmatrix}. \tag{2.142}$$

The displacement field and its derivatives are approximated using the conventional FEM interpolation scheme, based on shape functions and nodal displacement:

$$\mathbf{u} = \sum_{i=1}^{4} N_i \mathbf{u}^i, \tag{2.143}$$

and

$$\mathbf{u}_{,j} = \sum_{i=1}^{4} N_{i,j} \mathbf{u}^i. \tag{2.144}$$

Here N_i are the bi-linear shape function used for the *quad4* element and $\mathbf{u}^i$ is the displacement

vector relative to the nodes i of an element. It is worth noting that, for meshless formulations, the nodal displacement can be approximated in a very similar way. However, the bi-linear shape functions should be replaced by a MLS or RPIM shape function, and the summation would be over all the influence domain nodes of a given node and not just only over the 4 nodes of a *quad4* element.

Using the approximations of Equations 2.143 and 2.144, the variation of the Green-Lagrange strain tensor $\bar{\mathbf{E}}(\mathbf{u},\bar{\mathbf{u}})$ of Equation 2.127 can be written as:

$$\bar{\mathbf{E}} = \mathbf{B_N}\bar{\mathbf{u}}, \tag{2.145}$$

where $\mathbf{B_N}$ is called the nonlinear displacement-strain matrix, and is has the following form, depending on $\mathbf{u}$:

$$\mathbf{B_N} = \begin{bmatrix} F_{11}N_{1,1} & F_{21}N_{1,1} & \dots & F_{11}N_{4,1} & F_{21}N_{4,1} \\ F_{12}N_{1,2} & F_{22}N_{1,2} & \dots & F_{12}N_{4,2} & F_{22}N_{4,2} \\ F_{11}N_{1,2}+F_{12}N_{1,1} & F_{21}N1{,}2+F_{22}N_{1,1} & \dots & F_{11}N_{4,2}+F_{12}N_{4,1} & F_{21}N4{,}2+F_{22}N_{4,1} \end{bmatrix}. \tag{2.146}$$

The matrix $\mathbf{F}$ was defined in terms of displacement field in Equation 2.5 of Subsection 2.1.2 and it is repeated here for an ease reading:

$$\mathbf{F} = \begin{bmatrix} \frac{\partial u_1}{\partial X_1}+1 & \frac{\partial u_1}{\partial X_2} \\ \frac{\partial u_2}{\partial X_1} & \frac{\partial u_2}{\partial X_2}+1 \end{bmatrix}. \tag{2.147}$$

With the Lagrangian strain variation $\bar{\mathbf{E}}$ discretization is possible to also discretize the energy form $a(\mathbf{u},\bar{\mathbf{u}})$ defined in Equation 2.129 appearing also in Equations 2.140, as follows:

$$a(\mathbf{u},\bar{\mathbf{u}}) = \iint_{\mathbb{S}^0} \mathbf{S}(\mathbf{u}) \colon \bar{\mathbf{E}}\; d\mathbb{S}^0 \approx \bar{\mathbf{u}}^T \iint_{\mathbb{S}^0} \mathbf{B_N}^T \mathbf{S}(\mathbf{u})\; d\mathbb{S}^0 \equiv \bar{\mathbf{u}}^T \mathbf{f}^{int}. \tag{2.148}$$

Similarly, the load form:

$$l(\bar{\mathbf{u}}) = \int_{\Gamma_0} \bar{\mathbf{u}} \cdot \mathbf{t} d\Gamma \approx \sum_{i=1}^{4} \bar{\mathbf{u}}_\mathbf{i}^T \int_{\Gamma_0} N_i \, \bar{\mathbf{t}} \, d\Gamma \equiv \bar{\mathbf{u}}^T \mathbf{f}^{\mathbf{ext}}, \tag{2.149}$$

where $\mathbf{f}^{int}$ and $\mathbf{f}^{ext}$ stand for internal forces.

Similarly to the variation of Lagrangian strain $\bar{\mathbf{E}}(\mathbf{u},\bar{\mathbf{u}})$, the incremental Green-Lagrange strain tensor $\boldsymbol{\Delta}\mathbf{E}$ can be obtained with the following relation:

$$\boldsymbol{\Delta}\mathbf{E} = \mathbf{B_N} \boldsymbol{\Delta}\mathbf{u}. \tag{2.150}$$

Hence, the first integrand in the strain energy of Equation 2.139 becomes:

$$\iint_{\mathbb{S}^0} \bar{\mathbf{E}} : \mathbf{D} : \boldsymbol{\Delta}\mathbf{E} \, d\mathbb{S}^0 = \bar{\mathbf{u}}^T \left[\iint_{\mathbb{S}^0} \mathbf{B_N}^T \, \mathbf{D} \, \mathbf{B_N} \, d\mathbb{S}^0 \right] \boldsymbol{\Delta}\mathbf{u}. \tag{2.151}$$

The second integrand of Equation 2.139, frequently called the initial stiffness term, is given by the following formula:

$$\iint_{\mathbb{S}^0} \mathbf{D} : \boldsymbol{\Delta}\bar{\mathbf{E}} \, d\mathbb{S}^0 = \bar{\mathbf{u}}^T \left[\iint_{\mathbb{S}^0} \mathbf{B_G}^T \, \boldsymbol{\Sigma} \, \mathbf{B_G} d\mathbb{S} \right] \boldsymbol{\Delta}\mathbf{u}, \tag{2.152}$$

where:

$$\boldsymbol{\Sigma} = \begin{bmatrix} S_{11} & S_{12} & 0 & 0 \\ S_{12} & S_{22} & 0 & 0 \\ 0 & 0 & S_{11} & S_{12} \\ 0 & 0 & S_{11} & S_{12} \end{bmatrix}, \tag{2.153}$$

and

$$\mathbf{B_G} = \begin{bmatrix} N_{1,1} & 0 & N_{2,1} & 0 & N_{3,1} & 0 & N_{4,1} & 0 \\ N_{1,2} & 0 & N_{2,2} & 0 & N_{3,2} & 0 & N_{4,2} & 0 \\ 0 & N_{1,1} & 0 & N_{2,1} & 0 & N_{3,1} & 0 & N_{4,1} \\ 0 & N_{1,2} & 0 & N_{2,2} & 0 & N_{3,2} & 0 & N_{4,2} \end{bmatrix}. \tag{2.154}$$

Adding the expressions for first and second integrand of Equation 2.139 we have the tangent stiffness matrix:

$$\mathbf{K_T} = \iint_{\mathbb{S}^0} \left[\mathbf{B_N}^T \, \mathbf{D} \, \mathbf{B_N} + \mathbf{B_G}^T \, \mathbf{\Sigma} \, \mathbf{B_G} \right] d\mathbb{S}. \tag{2.155}$$

The discretized version of the linearized incremental residual - Equation 2.140 - is obtained by combining the expression 2.148, 2.149, 2.151, 2.152 and 2.155:

$$\bar{\mathbf{u}}^T \, \mathbf{K_T} \, \mathbf{\Delta u} = \bar{\mathbf{u}}^T \, \left(\mathbf{f}^{ext} - \mathbf{f}^{int} \right). \tag{2.156}$$

Removing the common term $\bar{\mathbf{u}}^T$:

$$\mathbf{K_T} \, \mathbf{\Delta u} = \mathbf{R}. \tag{2.157}$$

Which is the relation previously stated in Equation 2.141.

All the integrals necessary to evaluate the Equation 2.157 are calculated using the Gauss quadrature scheme for the *quad4* isoparametric element, with 4 integration points. Once the nodal tangent stiffness and external forces are calculated, they are assembled in the global matrix in the same manner shown in the section for linear FEM discretization. The Equation 2.157 is then iteratively solved using a Newton-Raphson scheme.

2.5.4 EFG Discretization

The EFG discretization of Equation 2.140 follows the same procedure used to FEM. The main difference is the type of shape function used in to approximate the displacement field. As explained in Subsection 2.3, the bi-linear shape functions N_i defined within an element i are replaced by MLS or RPIM shape functions ϕ_M^i and ϕ_R^i defined within the influence domain of node i.

As consequence, the matrix $\mathbf{B_N}$ and $\mathbf{B_G}$ have their sizes modified. Let n_e be the number of nodes inside the influence domain of a Gauss point, the matrix $\mathbf{B_N}$ and $\mathbf{B_G}$ will assume the following form:

$$\mathbf{B_N} = \begin{bmatrix} F_{11}\phi_{1,1} & F_{21}\phi_{1,1} & \dots & F_{11}\phi_{ne,1} & F_{21}\phi_{ne,1} \\ F_{12}\phi_{1,2} & F_{22}\phi_{1,2} & \dots & F_{12}\phi_{ne,2} & F_{22}\phi_{ne,2} \\ F_{11}\phi_{1,2} + F_{12}\phi_{1,1} & F_{21}\phi_{1,2} + F_{22}\phi_{1,1} & \dots & F_{11}\phi_{ne,2} + F_{12}\phi_{ne,1} & F_{21}\phi_{ne,2} + F_{22}\phi_{ne,1} \end{bmatrix}, \quad (2.158)$$

and

$$\mathbf{B_G} = \begin{bmatrix} \phi_{1,1} & 0 & \dots & \phi_{ne,1} & 0 \\ \phi_{1,2} & 0 & \dots & \phi_{ne,2} & 0 \\ 0 & \phi_{1,1} & \dots & 0 & \phi_{ne,1} \\ 0 & \phi_{1,2} & \dots & 0 & \phi_{ne,2} \end{bmatrix}. \quad (2.159)$$

The assembly of the tangent matrix $\mathbf{K_T}$ follows the same procedure described in Figure 2.13, that is, using a loop over Gauss integration points. The residual is calculated in that same way of FEM. After the global vectors are assembled, the Newton-Raphson method is carried out to obtain the displacement field.

3 Topology Optimization - BESO method

This chapter is dedicated to presenting the Bi-directional Evolutionary Structural Optimization - BESO - method. The general aspects of the method are presented considering the FEM approach, which is based on (HUANG AND XIE, 2010). The structural evolutionary optimization consists in gradually removing inefficient material from the structure, or design domain, looking at the maximization or minimization of a specified performance criterion, often called objective function. The BESO is hence an iterative method and an important characteristic is its ability to add or remove material along the iterations.

The material is removed from the structure by applying a penalization to the Young's modulus of an *element* if FEM is chosen as solver, or of a *node* if the EFG method is used. Of course, the final material distribution must be such that a predefined objective function is optimized. The strategy to penalize the Young's modulus and how to choose an objective function are presented in Section 3.1.

Basically, the BESO method removes inefficient material from the structure, and, in some cases, adds material to efficient regions. The material efficiency is determined through a sensitivity analysis, computing the effect of removing an element from the structure on the objective function. Elements with a low sensitivity number have less influence on the objective function than those with a high sensitivity number. Thus, a strategy to add and remove material can be implemented based on these sensitivity numbers. The sensitivity analysis is detailed in Section 3.2.

Problems like checker-board patterns and mesh-dependency are likely to happen when using BESO-FEM method with *quad4* elements. Fortunately, these problems are avoided by applying a filtering scheme to the sensitivity numbers, as explained in Section 3.5. Although no explicit filtering scheme is required in the BESO-EFG method, an extra technique is required to couple BESO with EFG: the dual-level interpolation method proposed by (ZHENG ET AL., 2015). This technique, which can work as a filter, is discussed in Section 3.6.

3.1 Problem formulation

In this work we are concerned with minimizing the mean compliance $\mathbf{C}$ of a structure, which is equivalent to maximizing its global stiffness, while removing material from the design domain until a target volume percentage W_f is reached. For now, a linear elastic behavior is considered and the problem can be written as:

$$\begin{aligned} &minimize:\ C = \tfrac{1}{2}\mathbf{f}^T\mathbf{u} \\ &subjected\ to:\ W_f - \textstyle\sum_{i=1}^{N} W_i\rho_i = 0 \\ &\mathbf{Ku} = \mathbf{f} \\ &\rho_i = \rho_{min}\ or\ 1 \end{aligned} \tag{3.1}$$

In this way, the element removal or addition consists on changing the relative density ρ_i of an element or a node, depending on rather FEM or EFG methods is used. More precisely, removing an element or a node consists in assigning them a low density value, often called ρ_{min}. Correspondingly, adding material is equivalent to restore the elemental or the nodal relative density to 1. In this work, as well as in many others (BENDSOE, 1989), (XIE AND STEVENS, 1993) and (HUANG AND XIE, 2010), ρ_{min} is set to 0.001, which is a common choice in elasticity problems to avoid singularities in the global stiffness matrix.

A pseudo-density is associated to the material properties using a penalization method, called *Solid Isotropic Material Penalization* or SIMP, where a power law is usually used to penalize the Young's modulus. In this way, the Young's modulus E_y^i of an element or node is written as:

$$E_y^i = \rho_i^p E_0, \tag{3.2}$$

where E_y^i is penalized Young modulus, ρ_i is the relative density of a given element or node i and E_y^0 is the initial Young's modulus of element or node i. Figure 3.1 shows the power law behavior for different values of p. Usually, for structural optimization problems, the penalization factor is set to $p = 3$.

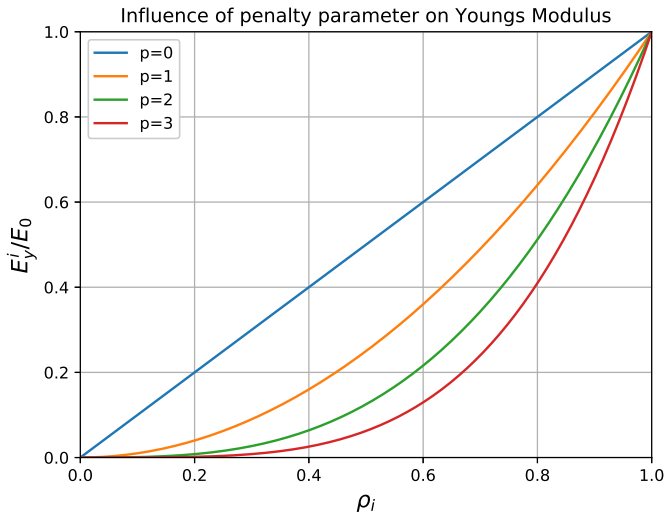


Figure 3.1: Influence of penalization factor p on Young's modulus. Typically, for structural optimization problems, $p = 3$.

3.2 Sensitivity analysis

Once the Young's modulus is written as a function of the design variables ρ_i, the sensitivity analysis can be tackled. In a nutshell, the sensitivity analysis is an indication of material efficiency. When added or removed, efficient material has a major impact on the objective function, while inefficient material causes less impact. In this sense, efficient material tends to be kept in the structure while the inefficient material tends to be removed.

In terms of discretization, a sensitivity number is assigned to each element or to each node, depending on rather FEM or EFG is used. Hence, the addition or removal of an element or node will depend on its sensitivity number, which is obtained with the derivative of mean compliance C with relation to each design variable ρ_i.

In order to do so, let us begin with the discretized elastic equation:

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{3.3}$$

where $\boldsymbol{f}$ the external force vector, assumed to be constant in direction, amplitude and application

point. Hence, the mean compliance C of the structure is:

$$C = \frac{1}{2}\mathbf{f}^T\mathbf{u}. \tag{3.4}$$

Using Equations 3.3 and 3.4, we can write the compliance as a function of the displacement field $\mathbf{u}$ and the global stiffness matrix $\mathbf{K}$:

$$C = \frac{1}{2}\left(\mathbf{K}\mathbf{u}\right)^T\mathbf{u}. \tag{3.5}$$

Using the transpose property of a matrix product, $(\mathbf{K}\mathbf{u})^T = \mathbf{u}^T\mathbf{K}^T$, and taking advantage from the fact that the stiffness matrix $\mathbf{K}$ is symmetric, which implies that $\mathbf{K}^T = \mathbf{K}$, the mean compliance can be written as:

$$C = \frac{1}{2}\left(\mathbf{u}^T\mathbf{K}\mathbf{u}\right). \tag{3.6}$$

Using the SIMP material model presented in the previous subsection, the mean compliance can be stated as a function of the relative density ρ_i. Using the vector $\boldsymbol{\rho}$ that collects all elemental or nodal relative densities, we can write:

$$C = \frac{1}{2}\mathbf{u}^T\mathbf{K}\left(\boldsymbol{\rho}\right)\mathbf{u}. \tag{3.7}$$

As previously stated, the sensitivity number is an indication of how the compliance varies when removing a node. In this sense, the sensitivity number is the compliance derivative with relation to each design variable ρ_i. We are hence looking for:

$$\frac{\partial C}{\partial \rho_i} = \frac{\partial}{\partial \rho_i}\left(\mathbf{u}\left(\boldsymbol{\rho}\right)^T\mathbf{K}\left(\boldsymbol{\rho}\right)\mathbf{u}\left(\boldsymbol{\rho}\right)\right). \tag{3.8}$$

Before calculating the derivatives of Equation 3.8, it worth noting that the term in parenthesis is in a quadratic form. A useful property is the fact that a generic quadratic form denoted θ

$$\theta = \mathbf{v^T}\ \mathbf{A}\ \mathbf{v} \tag{3.9}$$

has the following derivatives with relation to a scalar quantity z:

$$\frac{\partial \theta}{\partial z} = 2\mathbf{v}^T\mathbf{A}\frac{\partial \mathbf{v}}{\partial z} + \mathbf{v}^T\frac{\partial \mathbf{A}}{\partial z}\mathbf{v}. \tag{3.10}$$

Applying this property to the compliance derivative of Equation 3.8, we have:

$$\frac{\partial C}{\partial \rho_i} = \frac{1}{2}\left(2\mathbf{u}^T\mathbf{K}\frac{\partial \mathbf{u}}{\partial \rho_i} + \mathbf{u}^T\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u}\right). \tag{3.11}$$

It is now necessary to evaluate both derivatives of $\mathbf{u}$ and $\mathbf{K}$ in relation to ρ_i. To obtain $\partial\mathbf{u}/\partial\rho_i$ one can derive equilibrium equation:

$$\frac{\partial}{\partial \rho_i}(\mathbf{K}\mathbf{u}) = \frac{\partial \mathbf{f}}{\partial \rho_i}. \tag{3.12}$$

Assuming that the external load does not vary with material removal, Equation 3.12 becomes:

$$\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u} + \mathbf{K}\frac{\partial \mathbf{u}}{\partial \rho_i} = 0. \tag{3.13}$$

Arranging the terms:

$$\frac{\partial \mathbf{u}}{\partial \rho_i} = -\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u}, \tag{3.14}$$

and replacing Equation 3.14 into Equation 3.11:

$$\frac{\partial C}{\partial \rho_i} = \frac{1}{2}\left(-2\mathbf{u}^T\mathbf{K}\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u} + \mathbf{u}^T\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u}\right). \tag{3.15}$$

Observing that $\mathbf{K}\mathbf{K}^{-1} = \mathbf{I}$, it is possible to simplify the previous equation to:

$$\frac{\partial C}{\partial \rho_i} = \frac{1}{2}\left(-\mathbf{u}^T\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u}\right). \tag{3.16}$$

From the SIMP material definition, the term $\partial\mathbf{K}/\partial\rho_i$ is:

$$\frac{\partial \mathbf{K}}{\partial \rho_i} = p\ \rho_i^{(p-1)}\mathbf{K_0}, \tag{3.17}$$

where $\mathbf{K_0}$ is the initial element or nodal stiffness, that is, the stiffness matrix without penalizing Young's module. Finally, the sensitivity number can be written in terms of a given element or node i, as follows

$$\alpha_i = -\frac{1}{2}\left\{\mathbf{u}\right\}_i^T\left[\frac{\partial \mathbf{K}}{\partial \rho}\right]_i\left\{\mathbf{u}\right\}_i, \tag{3.18}$$

where the compliance derivative with respect to the design variable is denoted α for simplicity. When using FEM, $\{\mathbf{u}\}_i$ is a vector collecting the nodal displacement of all nodes of an element i. If the EFG is employed, $\{\mathbf{u}\}_i$ becomes a vector collecting the displacement of all nodes inside the influence domain of a node i. The term $\left[\frac{\partial \mathbf{K}}{\partial \rho}\right]_i$ represent the stiffness matrix of an element or a node i, depending on FEM or EFG is being used.

Now that sensitivity is calculated, it is possible to decide which elements to remove. From sensitivity definition, the nodes with the highest sensitivity numbers should be deleted. There is, however, a specific number of nodes or elements that can be removed per iteration, this number is called evolutionary ratio *ER* which is defined as the ratio between the number of nodes or elements

to be removed at each iteration and the total number of nodes or elements. This concept is presented in more detail in the next section.

3.3 Material removal and addition

Once the sensitivity number is properly calculated, it is necessary to determine how many nodes or elements will be removed or added. The first step is defining an evolutionary ratio ER, which represents how many nodes or elements will be removed from topology of current iteration k. With the evolutionary ratio one can calculate the structure weight at iteration $k+1$, which, consequently, allows us to verify when the target weight W^* is reached.

The target weight is normally expressed as percentage of the structures initial weight. In this way, the target weight represents how much material we want to remove from the structure. For example a target weight of 30% means that material will be removed until the resulting structure presents a weight equivalent to 30% of the initial structure's weight. Often, the amount of material to be removed can be equivalently expressed in terms of target volume.

In this way, there are three possibilities:

- If $W^{k+1} > W^*$ then it is still possible to remove material: $W^{k+1} = W^k(1 - ER)$
- If $W^{k+1} < W^*$ then it is necessary to add material: $W^{k+1} = W^k(1 + ER)$
- If $W^{k+1} = W^*$ then the target volume has been reached: $W^{k+1} = W^*$

To establish an addition or removal criteria, we begin with sorting the elements or nodes sensitivity in descendant order. Hence, a solid element or node i will be removed if:

$$\alpha_i \leq \alpha_{del}^{th}. \tag{3.19}$$

Which is to say that a material with a sensitivity number below a given threshold α_{del}^{th} should be removed. In turn, a void node will be added if:

$$\alpha_i > \alpha_{add}^{th}. \tag{3.20}$$

Here α_{del}^{th} and α_{add}^{th} are threshold sensitivity numbers which put levels for removing or adding nodes. These threshold numbers can be determined following three steps (HUANG AND XIE, 2010):

1. Let $\alpha_{del}^{th} = \alpha_{add}^{th} = \alpha^{th}$. For a design domain with 1000 elements, if W^{k+1} corresponds to 725 solid element, then $\alpha^{th} = \alpha_{725}$.

2. Calculate the addition ratio AR. This is the number of nodes which would be added based on the previously obtained α^{th} divided by the total number of nodes. If $AR \leq AR_{max}$ the nodes can be added or removed. However if $AR > AR_{max}$ it will be necessary to proceed as explained in step 3 to determine which nodes to add or remove.

3. Recalculate α_{add}^{th}, sorting the sensitivity number of only void elements in descendant order. The elements to be added are then the first e elements, where e is AR_{max} times the total number of nodes. The number of nodes to remove is calculated so that the removed weight is equal to $W^k - W^{k+1} + W_{added}$, where W_{added} is the weight of added nodes.

3.4 Convergence criteria

When the target weight W^* is reached, we can define a convergence criterion to stop de iterations or the BESO will continue to add and remove material indefinably at $W^k = W^*$. The convergence criterion states that the change ϵ in the objective function over N iterations should be considerably small. The expression for the converge criterion is:

$$\xi = \frac{\left|\sum_{i=1}^{N} C_{k-i+1} - \sum_{i=1}^{N} C_{k-N-i+1}\right|}{\sum_{i=1}^{N} C_{k-i+1}} \leq \tau, \tag{3.21}$$

where k is the current iteration. In his book, HUANG and XIE (2010) recommends N to be 5, which implies that the change in objective function over the last 10 iterations is acceptably small

and τ is the defined convergence tolerance. The number N is an arbitrary choice and can assume other values.

3.5 BESO-FEM filtering

When using FEM with the BESO algorithm, and more specifically, when a *quad4* element is employed. a filtering scheme is required to eliminate checkerboard patterns in the final topology. We will verify, along results chapter, that neither filtering nor stabilization are required when EFG methods are used with BESO strategy.

The objective of applying a filter over elements sensitivity is to avoid checkerboards patterns and mesh-dependency problems. As studied in previous project, the ESO method is susceptible to encounter both issues: checkerboards - due to *quad4* elements - and mesh-dependency. The filter is applied after calculating the sensitivity for all elements with Equation 3.18. The filtering process is divided in two steps as presented below.

3.5.1 Nodal sensitivity

The first step in BESO-FEM filtering process is to obtain the nodal sensitivity, which is a weighted average of the adjacent elemental sensitivity, as illustrated in Figure 3.2. In order words, considering that the j^{th} node is surrounded by M adjacent elements with sensitivity α_i, the nodal sensitivity α_j^n will be:

$$\alpha_j^n = \sum_{i=1}^{M} w_i \alpha_i. \tag{3.22}$$

Here a superscript n is used to differentiate nodal sensitivity from element sensitivity. The weight w_i is defined as an inverse distance law. Considering r_{ij} to be the distance between the centroid of element i and the node j, we have:

$$w_i = \frac{1}{M-1}\left(1 - \frac{r_{ij}}{\sum_{i=1}^{M} r_{ij}}\right). \tag{3.23}$$

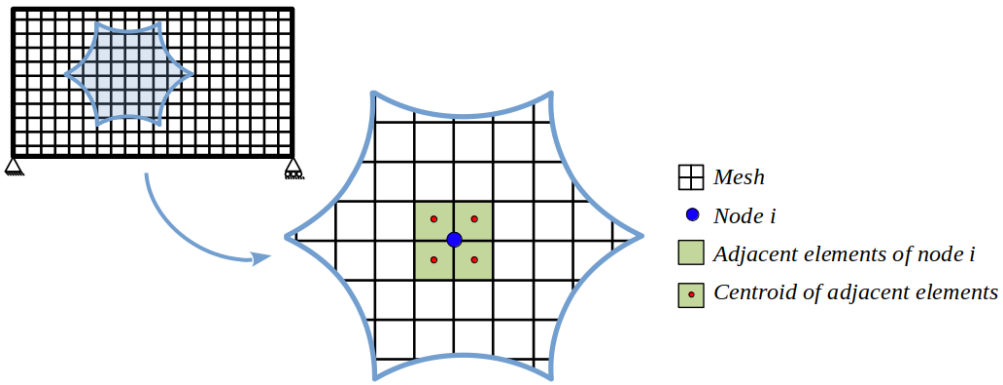


Figure 3.2: The nodal sensitivity is calculated based on the centroid sensitivity of adjacent elements and their distance to the current node.

3.5.2 Smoothed elemental sensitivity

With α^n calculated for all nodes, the element sensitivity can be recalculated using a weighted average of its neighbor nodes sensitivity. Typically, the neighborhood of an element is defined by a circle with radius r_{min} centered in the element centroid, as shown in Figure 3.3. Considering that K nodes lie in neighborhood of an element i, its sensitivity becomes:

$$\alpha_i = \frac{\sum_{j=1}^{N} w(r_{ij})\alpha_j^n}{\sum_{j=1}^{N} w(r_{ij})}. \tag{3.24}$$

Different from the weight used to obtain the nodal sensitivity, here $w(r_{ij})$ is defined as:

$$w(r_{ij}) = r_{min} - r_{ij}. \tag{3.25}$$

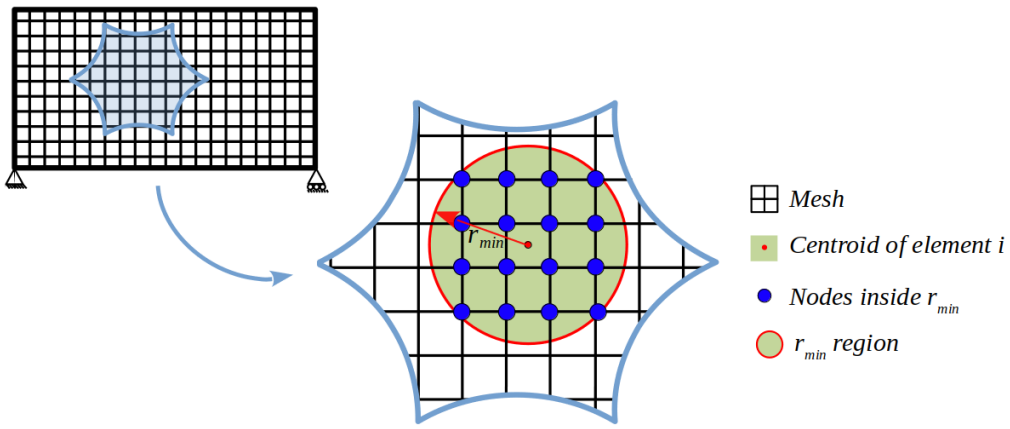


Figure 3.3: The smoothed elemental sensitivity is obtained through a weighted average of neighbors nodes sensitivity.

3.6 Dual-level interpolation

Some authors point that BESO-EFG does not require an additional filter scheme (SHOBEIRI, 2015), (SHOBEIRI, 2016) and (ZHAO, 2014). However, an interpolation is needed to couple the nodal densities with Gauss point densities, in order to correctly penalize the nodal stiffness. This interpolation can be though of as filtering scheme.

More precisely, as explained in Section 2.3, the global stiffness matrix in the EFG method, is assembled in a loop over the integration points, where, evidently, the integration is performed. As the material is removed or added by changing the Young's modulus of discretization nodes, this nodal density should be accounted at each Gauss point used for integration. This accountability is done through the interpolation of nodal density using a inverse distance law to obtain the equivalent nodal density at each Gauss point position.

In addition to the interpolation of nodes density to Gauss points position, the relative density of given node can also be smoothed using an interpolation of its neighbor nodes density through an inverse distance law, in a very similar way to the BESO-FEM filter.

In his work ZHENG et al. (2015) states the Dual-level Interpolation Method, in which the two interpolations mentioned above are performed: a first level interpolation to smooth nodal density and a second level interpolation to couple the discretization nodes density with the Gauss point

position. Both interpolations use the Shepard's functions – which use, essentially, inverse distance law. This methodology is employed in this work in order to implement the EFG-BESO.

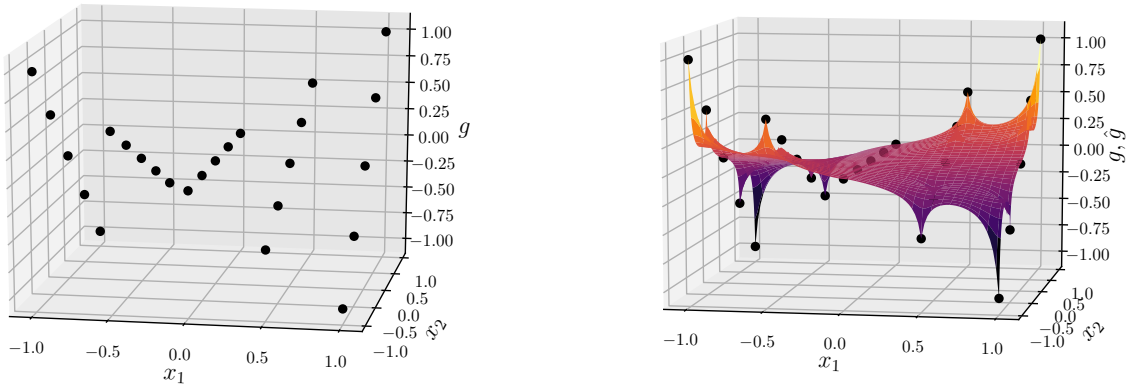
Before presenting the dual-level interpolation itself, it is important to present the Shepard function used in these interpolations. The Shepard method is a simple way of interpolating a set of N points, for example in $\mathbb{R}^2$, with nodal values $\mathbf{g} = g^i$ and coordinates $\mathbf{x} = \mathbf{x}^i = (x_1^i,x_2^i)$ for $i = 1,\ldots,N$. The interpolation function $\hat{\mathbf{g}} = \hat{g}^i$ at a given point $\mathbf{y} = (y_1,y_2)$ depends on the Euclidean distance d between the nodes $\mathbf{x}$ and the interpolation point $\mathbf{y}$. We can write:

$$\hat{g}(\mathbf{y}) = \begin{cases} \dfrac{\sum_{i=1}^{N} w^i(\mathbf{x}^i,\mathbf{y}) g^i}{\sum_{i=1}^{N} w^i(\mathbf{x}^i,\mathbf{y})} & \text{if } d(\mathbf{x}^i,\mathbf{y}) \neq 0 \\ \\ g^i & \text{if } d(\mathbf{x}^i,\mathbf{y}) = 0. \end{cases} \tag{3.26}$$

where:

$$\begin{aligned} w^i(\mathbf{x},\mathbf{y}) &= \frac{1}{d(\mathbf{x}^i,\mathbf{y})^q} \\ d(\mathbf{x}^i,\mathbf{y}) &= \sqrt{(x_1^i - y_1^i)^2 + (x_2^i - y_2^i)^2}. \end{aligned} \tag{3.27}$$

Here q is called power parameter. For illustration, Figure 3.4 shows the Shepard's interpolation of 5×5 nodal values generated with the function $g(\mathbf{x}) = x_1^i x_2^i$ over a squared $[-1,1] \times [-1,1$ domain.



(a) Scattered points

(b) Shepard interpolation

Figure 3.4: Interpolating a data set of 5×5 nodes in a squared domain $[-1,1] \times [-1,1]$ using the Shepard's method. The nodal values of (a) are $g(\mathbf{x}) = x_1^i x_2^i$ and the surface of (b) were generated with 50×50 equally space points.

3.6.1 First level

As previously mentioned, for smooth topology results, it is desired to approximate the relative density at any nodes i based on the relative density of nodes in the neighborhood of i, as illustrated in Figure 3.5. The neighborhood is a compact domain, similar to the influence domain of a node.

We state that the density of a node i is affected by the density of its neighbor nodes in the following manner:

$$\rho_i = \sum_{k=1}^{m} \phi(\mathbf{x}^i, \mathbf{x}^k)\rho_k, \tag{3.28}$$

where ϕ is the first Shepard function, m is the number of neighbor nodes of i and $\mathbf{x}$ are nodal coordinates vectors.

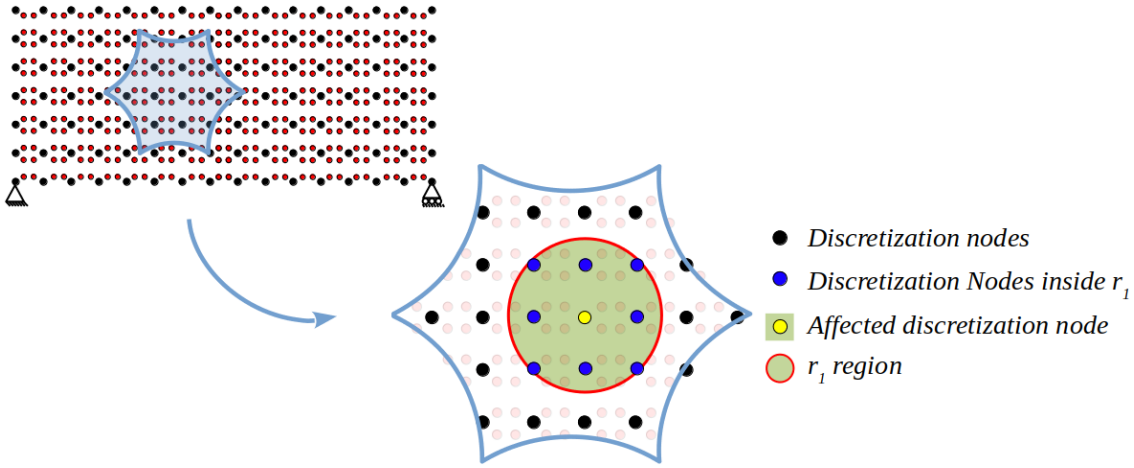


Figure 3.5: Smoothing nodal relative density using the Shepard interpolation. The compact domain which defines the neighborhood of the affected node is illustrated as the r_1 region.

3.6.2 Second level

Here, we want to assign a relative density for the Gauss points in order to penalize the global stiffness matrix in the EFG method. As illustrated in Figure 3.6, a compact domain is also attributed to a Gauss point j and we state that the discretization nodes lying inside this compact domain will affect the relative density of the Gauss point as follows:

$$\rho_j^{gp} = \sum_{k=1}^{n} \psi(\mathbf{x}_{gp}^j, \mathbf{x}^k)\rho_n. \tag{3.29}$$

Here the superscript gp is added to distinguish nodal from Gauss point density. Again, ψ is the second Shepard function, m is the number of nodes inside the compact domain of Gauss point j and $\mathbf{x}$ are nodal coordinates vectors and $\mathbf{x}_{gp}$ are the Gauss point coordinates vector.

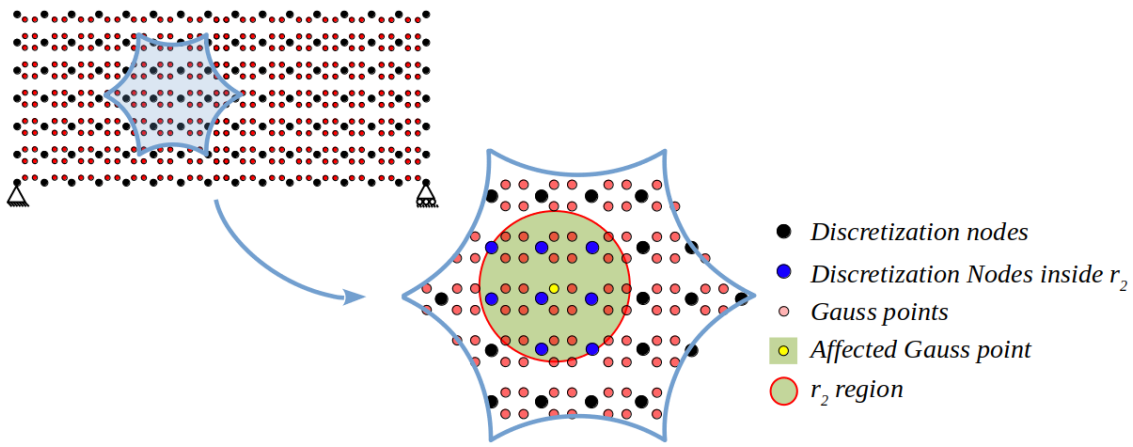


Figure 3.6: Shepard interpolation used to assign relative density to a Gauss point. The compact influence of the affected Gauss point is represented by r_2 region.

It is worth noting that compact domains used in the dual-level interpolation and the influence domain of a node or a Gauss point can assume different sizes.

Now that the Gauss point has densities based on the nodal relative density, one can penalize the contribution of a given Gauss point j when evaluating the global stiffness matrix through the penalization of Young's modulus at the position $\mathbf{x}_{gp}^{j}$:

$$E_y\left(\mathbf{x}_{gp}^{j}\right) = E_j^{gp} = \left(\rho_j^{gp}\right)^p E_0^{gp}, \tag{3.30}$$

where ρ_j^{gp} is given by Equation 3.29. Thus, when a node is removed, the Gauss points densities are affected and, consequently, the global stiffness matrix – which is obtained using the Gauss Quadrature integration – is impacted.

3.7 Stabilization of evolutionary process

This technique is used to avoid mesh-dependency problems and consists on averaging the sensitivity numbers of two consecutive iterations k and $k+1$:

$$\alpha_i^k = \frac{\alpha_i^k + \alpha_i^{k+1}}{2}. \tag{3.31}$$

Thus, the updated sensitivity number includes the whole history of the sensitivity information of previous iterations.

3.8 Algorithms

The FEM-BESO procedure is given in Figure 3.7. It is worth noting that the filtering process of BESO-FEM corresponds to steps *Calculate nodal sensitivity* and *Calculate smoothed element sensitivity*. The stop criteria are the target weight $W^k = w^*$ and the small changes in mean compliance of Equation 3.21. Evidently a maximum number of iterations should also be imposed.

The algorithm for EFG-BESO using the dual-level interpolation is shown in Figure 3.8, where the loop over Gauss points for integrate and assemble the global matrix is also depicted to illustrate the second interpolation level. The dual-level interpolation consists in steps *Smooth nodal density with Shepard's interpolation* and *Assign density to Gauss point with Shepard's interpolation*. One may note that the first interpolation level happens before carrying out the EFG analysis while the second level of interpolation is done during the loop to assemble the stiffness matrix. In Section 2.3 the loop over Gauss points and the EFG procedure are presented with more details.

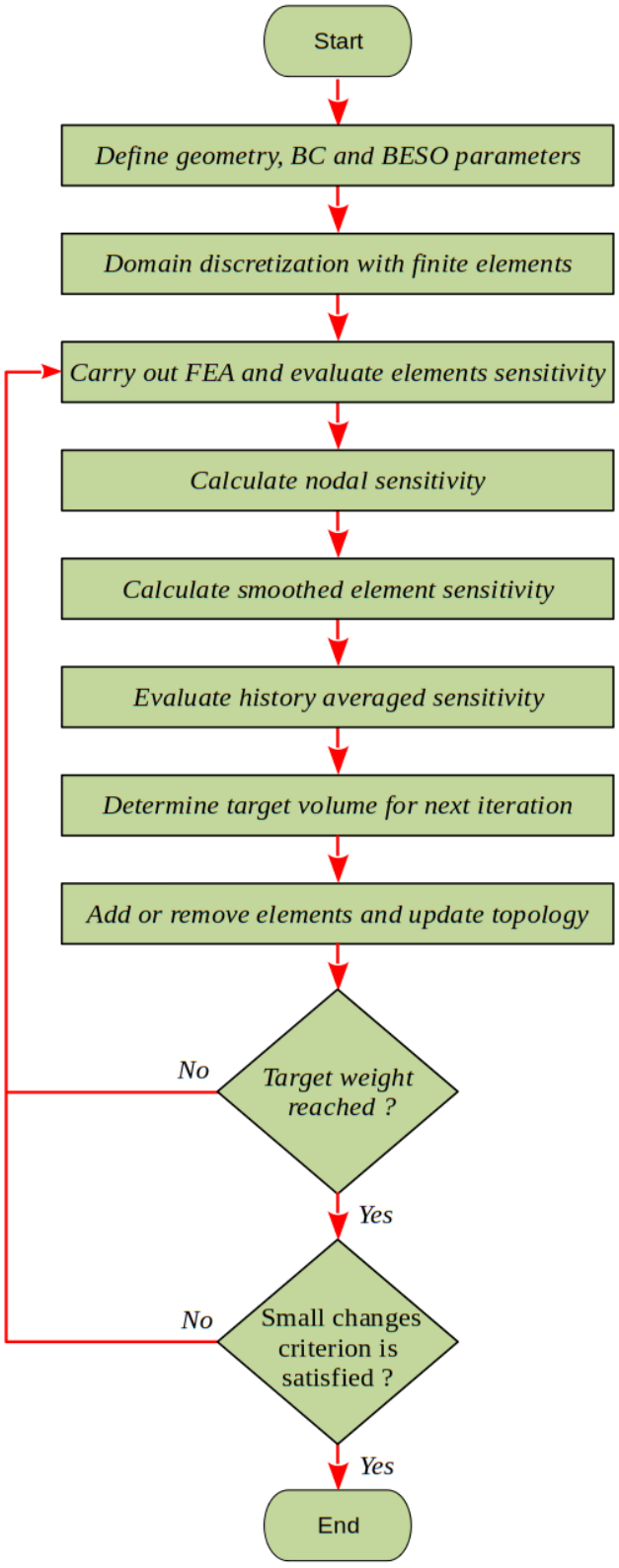


Figure 3.7: BESO-FEM Algorithm.

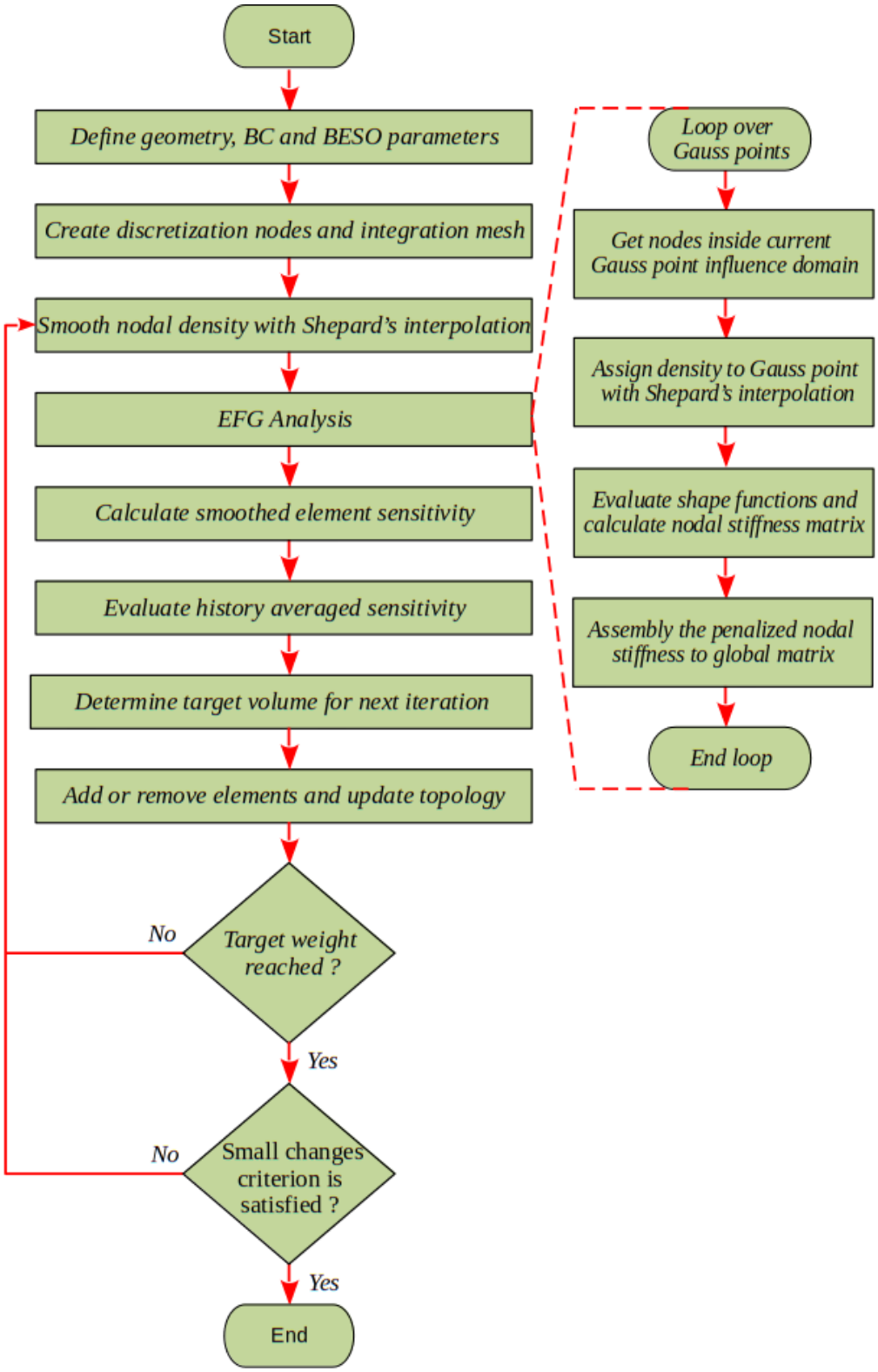


Figure 3.8: BESO-EFG algorithm.

3.9 Objective function and sensitivity number of nonlinear cases

The optimization problem of this work is the minimization the mean compliance C of a structure while removing material from the design domain. When it comes to optimize a nonlinear structure, different approaches can be used to solve this problem. In his work, BUHL et al. (2000) presents three options. The first is the minimization of the end-compliance, this is a very natural choice, where only the compliance at equilibrium is considered. The second options is a multiple loading case, where a weighted average of the end-compliance for different loads is used. The third option is the minimization of the complementary work of external forces for each load increment. In this work, as well as the reference textbook (HUANG AND XIE, 2010), we choose the minimization of external load approach.

To better illustrate this concept, consider the Figure 3.9, where the external force smoothly increases to a maximum value $\mathbf{F}^{max}$, for which the structure will present a total displacement of $\mathbf{u}^*$, and the complementary work is the shaded area. The trapezoidal rule can be used to calculate the shaded area as a function of the displacement at iteration k, denoted ${}^{k}\mathbf{u}$, and the load increment $\boldsymbol{\Delta}\mathbf{F}$, as stated in Equation 3.32.

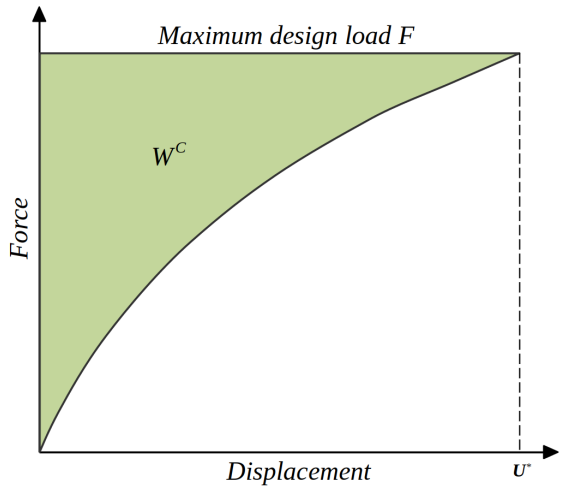


Figure 3.9: Illustration of the complementary work of external loads, represented by the shaded area. Adapted from (HUANG AND XIE, 2010).

$$W^C = \lim_{n\to\infty} \left[\frac{1}{2} \sum_{k=1}^{n} \mathbf{\Delta F}^T \left({}^k\mathbf{u} - {}^{k-1}\mathbf{u} \right) \right]. \tag{3.32}$$

As pointed by BUHL et al. (2000), the weak point of using the end-compliance is that the structure can fail for loads lower than the final design load. By minimizing the complementary work we can make sure that the structure will be stable for any load lower than the design load (HUANG AND XIE, 2010).

The optimization problem for nonlinear structures can be written as:

$$\begin{aligned} &minimize: \ W^C = \lim_{n\to\infty} \left[\tfrac{1}{2} \textstyle\sum_{k=1}^{n} \mathbf{\Delta F}^T \left({}^k\mathbf{u} - {}^{k-1}\mathbf{u} \right) \right] \\ &subjected\ to: \ W_f - \textstyle\sum_{i=1}^{N} W_i \rho_i = 0 \\ &\qquad\qquad \mathbf{Ku} = \mathbf{f} \\ &\qquad\qquad \rho_i = \rho_{min} \ or \ 1 \end{aligned} \tag{3.33}$$

The sensitivity number can be obtained from the complementary work of Equation 3.32, as follows:

$$\frac{\partial W^C}{\partial \rho_i} = \lim_{l\to\infty} \left[\frac{1}{2} \sum_{k=1}^{l} \left({}^k\mathbf{F}^T - {}^{k-1}\mathbf{F}^T \right) \left(\frac{\partial}{\partial \rho_i} ({}^k\mathbf{u}^T) + \frac{\partial}{\partial \rho_i} ({}^{k-1}\mathbf{u}^T) \right) \right]. \tag{3.34}$$

An adjoint method is used by BUHL et al. (2000) and HUANG and XIE (2010) to obtain the sensitivity of objective function. In both works, a design independent from load is assumed. It is shown by HUANG and XIE (2010) that the sensitivity when using the BESO method – which uses discrete design variables – can be written in a very similar way to the linear case:

$$\alpha_i = \begin{cases} U^i & \text{when } \rho_i = 1 \\ \rho_{min}^{p-1} U^i & \text{when } \rho_{min} = 1, \end{cases} \tag{3.35}$$

where U^i is the strain energy of an element or a node:

$$U^i = \frac{1}{2}\mathbf{S}^i \cdot \mathbf{E}^i. \tag{3.36}$$

As we are in nonlinear regime, the Green-Lagrange strain $\mathbf{E}$ of Equation 2.119 has to be used together with the Second Piola-Kirchhoff stress tensor $\mathbf{S}$.

4 Validations and Results

Along this chapter the procedures used to validate the implemented codes are presented, followed by the results achieved with the BESO-FEM and BESO-EFG. In a first moment, the codes are validated for both linear and nonlinear problems. Then, the topology optimization results of linear structures are detailed and, subsequently, the topology optimization results for nonlinear cases are presented.

The first validation problem is the linear cantilever beam. This is a relatively simple problem which has an analytic solution, being widely used for code validation. The FEM and EFG results are hence contrasted with the analytic solution provided by TIMOSHENKO and GOODIER (1970). Through this problem, a comparison between the two versions of the EFG method – one with MLS and other with RPIM shape functions – can also be carried out. The nonlinear cantilever beam is used as the second validation problem. As for this case no analytic solution is available, the validation of EFG and FEM codes is accomplished through a comparison with the solution obtained using the commercial software ANSYS 19.2.

The topology optimization results for linear structures are presented through four benchmarks problems: two-bar structure, Michell-type structure, cantilever beam, MBB beam. The BESO-FEM and BESO-EFG are compared in terms of final topology and convergence curve for each of these problems. Finally, the BESO-EFG method is used in the topology optimization of a nonlinear cantilever beam. In order to evaluate the quality of results, the final topologies obtained with the BESO-EFG are compared with literature solutions.

4.1 Validation

As previously mentioned, the cantilever beam will be used to validate the implemented codes. The validation procedure is divided in two parts: the first considers linear cantilever beam while the second one is dedicated to the validation of the codes for geometrically nonlinear structures.

4.1.1 Linear case

The cantilever beam used for validation is shown in Figure 4.1. The analytic solution for both displacement and stress fields are given in Equations 4.2 and 4.3 (TIMOSHENKO AND GOODIER, 1970), where the beam's width is considered to be unitary. The strain field is obtained by differentiating the displacement field, as stated in Equation 2.11. The exact solution requires a shearing traction $\mathbf{t}$ with parabolic distribution at the free extremity to be valid. Considering a parabola with maximum value P, the traction is written as:

$$\mathbf{t} = \begin{bmatrix} 0 \\ -\frac{P}{2I}\left(\frac{l_2^2}{4} - x_2\right) \end{bmatrix}. \tag{4.1}$$

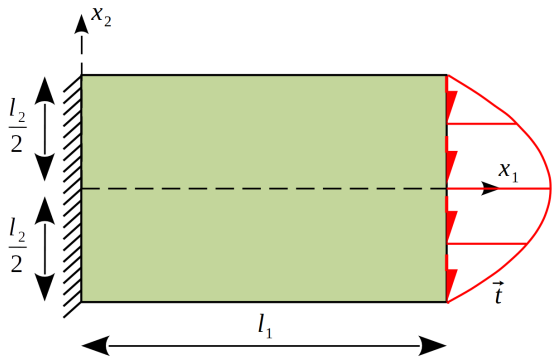


Figure 4.1: Cantilever beam illustration. The structure is subject to a parabolic shearing traction $\vec{\mathbf{t}}$ applied at $x_1 = l_1$ and it is fixed at $x_1 = 0$ portion.

$$\begin{aligned} u_1(\mathbf{x}) &= -\frac{Px_1}{6EI}\left(3(x_1^2 - l_1^2) - (2+\nu)x_2^2 + 6(1+\nu)\frac{l_2^2}{4}\right), \\ u_2(\mathbf{x}) &= \frac{P}{6EI}\left(3\nu\, x_1 x_2^2 + x_1^3 - 3l_1 x_1 + 2l_1^3\right), \end{aligned} \tag{4.2}$$

$$\begin{aligned}\sigma_{11} &= -\frac{Px_1x_2}{I},\\ \sigma_{22} &= 0,\\ \sigma_{12} &= -\frac{P}{2I}\left(\frac{l_2^2}{4} - x_2^2\right).\end{aligned} \tag{4.3}$$

Here I is the moment of inertia:

$$I = \frac{l_2^3}{12} \tag{4.4}$$

To solve this problem using the EFG method, the rectangular domain of $l_1 = 48mm$ by $l_2 = 12mm$ is discretized with 49×13 nodes. A background mesh containing 48×12 integration cells with 4 integration points is used. The FEM solver uses a mesh with 48×12 isoparametric elements with 4 integration points per element. A fictitious material with Young's module of $E_y^0 = 210GPa$ and Poisson's coefficient of $\nu = 0.3$ is used.

In a first time, a visual inspection was carried out and no significant differences between the methods could be found. In this way, for a better analysis, quantitative indicators must be used. We will use for this analysis the total displacement of the charged node at $x_1 = l_1$. The results for loads from $100N$ to $5000N$, obtained with EFG and FEM are compared to those coming by the analytic formula in the chart of Figure 4.2. As expected, the relative error between analytic and numerical solutions remain constant for all loads P. When compared with the analytic solution, the EFG-MLS presents a relative error of 2.3%, while the EFG-RPIM relative error is 2.5% and the FEM come up with 3.2%. As the relative errors are inferior to 5% the implemented codes are validated for linear structures.

A quantitative analysis of the results obtained with EFG-MLS and EFG-RPIM is carried out using the strain energy error norm. This scalar quantity is the relative error between the strain energies obtained with analytical and numerical solutions. It's given by:

$$\|E\| = \frac{\left\{\int_{\Omega} \frac{1}{2}\left[\boldsymbol{\epsilon}^h(\mathbf{x}) - \boldsymbol{\epsilon}^a(\mathbf{x})\right]^T \left[\boldsymbol{\sigma}^h(\mathbf{x}) - \boldsymbol{\sigma}^a(\mathbf{x})\right]\right\}^{1/2}}{\left\{\int_{\Omega} \frac{1}{2}\,\boldsymbol{\epsilon}^a(\mathbf{x})^T \boldsymbol{\sigma}^a(\mathbf{x})\right\}^{1/2}}. \tag{4.5}$$

In Figure 4.3 the norm errors are given as function of different discretization and Gauss points. These results are in accordance with (OVERVELDE, 2012) and (BELYTSCHKO ET AL., 1994). One may note in Figure 4.3 that the MLS results are more precise then RPIM's.

As the implementation of EFG-RPIM method is slightly simplified because of the imposition of boundary conditions and recognizing that the difference in results precision between MLS and RPIM is not big, the RPIM shape functions will be used hereafter. Indeed, when using RPIM shape functions the essential boundary conditions can be imposed alike FEM, while Lagrange multipliers or penalty methods must be used to impose Dirichlet conditions when using MLS shape functions.

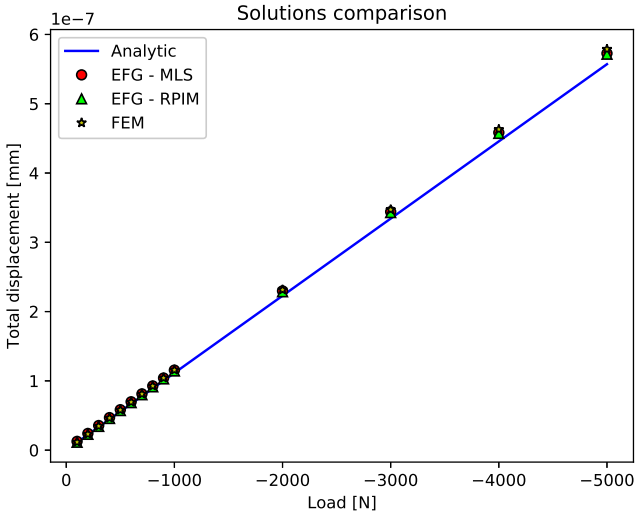
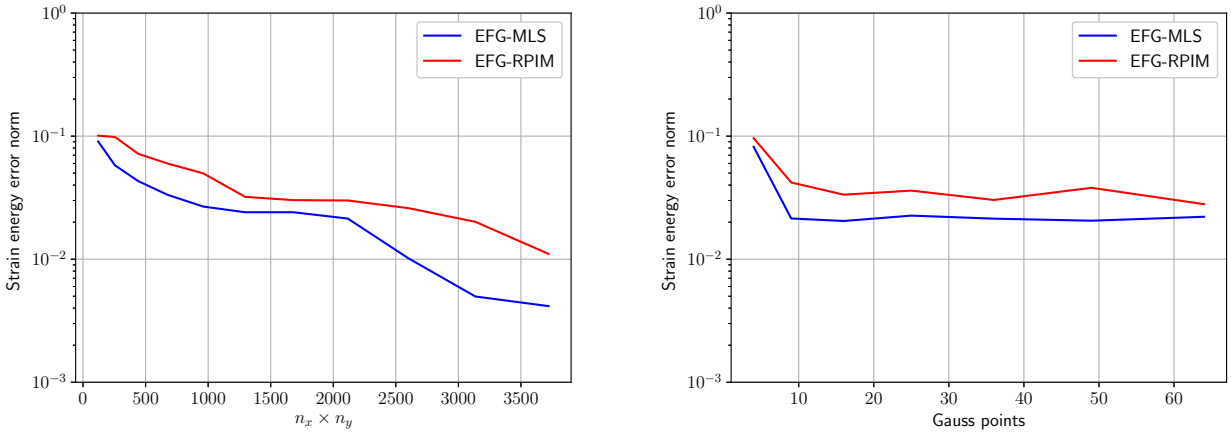


Figure 4.2: Comparison between numerical and analytical solutions for the linear elastic cantilever.

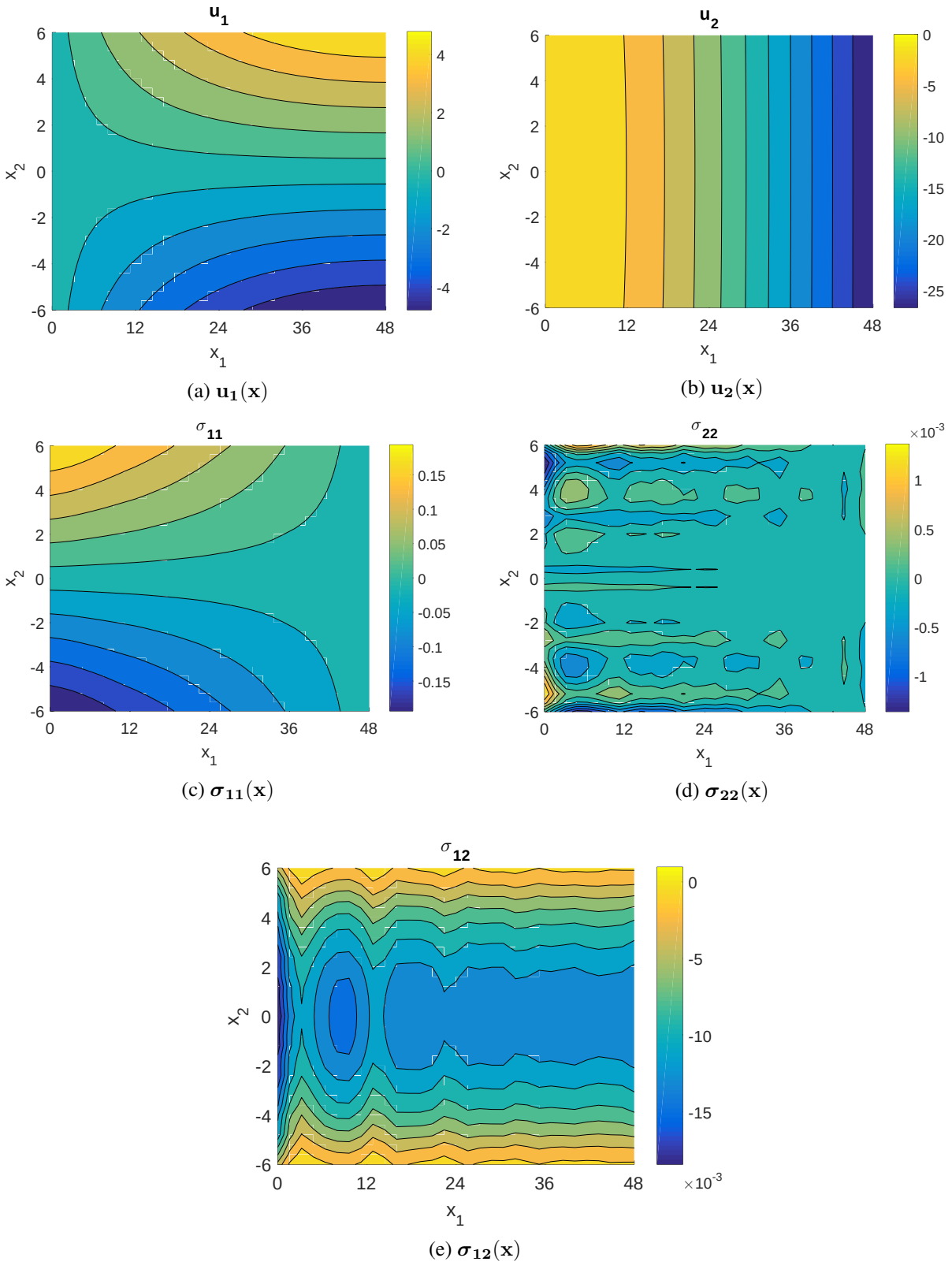


(a) Scattered points

(b) Shepard interpolation

Figure 4.3: Strain energy error analysis. The influence of discretization on the accuracy is given in a and the influence of integration points is given in b.

As an illustration, the results of the cantilever beam obtained using the EFG-MLS are given in Figures 4.4a to 4.4e. The problem configuration was $l_1 = 48$, $l_2 = 12$, $P = 0.1$, $E = 1$ and $\nu = 0.3$ (OVERVELDE, 2012).



(a) $\mathbf{u_1}(\mathbf{x})$

(b) $\mathbf{u_2}(\mathbf{x})$

(c) $\boldsymbol{\sigma_{11}}(\mathbf{x})$

(d) $\boldsymbol{\sigma_{22}}(\mathbf{x})$

(e) $\boldsymbol{\sigma_{12}}(\mathbf{x})$

Figure 4.4: Results for the cantilever beam: displacement $\mathbf{u}$ and stress $\boldsymbol{\sigma}$ fields in arbitrary units.

4.1.2 Nonlinear case

Here we will study the results for nonlinear structures obtained with EFG-RPIM and FEM. The geometrically nonlinear cantilever beam used for validation is presented in Figure 4.5. The beam dimensions are $l_1 = 100mm$ and $l_2 = 5mm$. Here we are going to use a fictitious material with Young's modulus of $E = 200MPa$ and a Poisson's coefficient of $\nu = 0.3$. Initially, the force at the right-hand extremity is set to $P = 100N$. For the EFG method, the discretization uses 151×21 nodes and 150×20 integration cells, with 4 integration points per cell. The FEM is applied using a mesh with 100×20 isoparametric elements with 4 Gauss points.

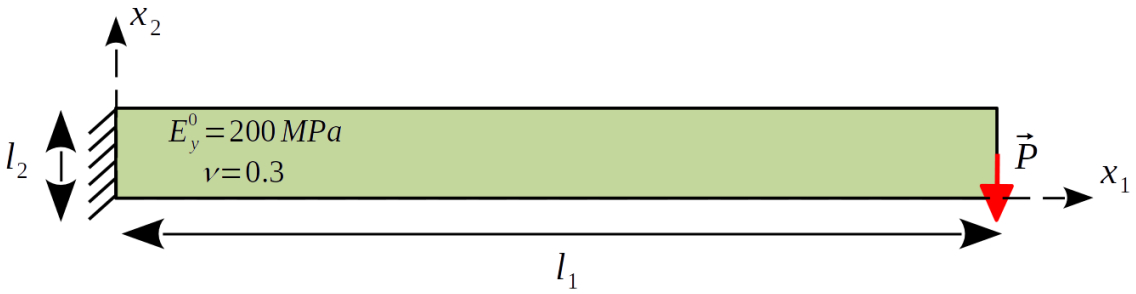
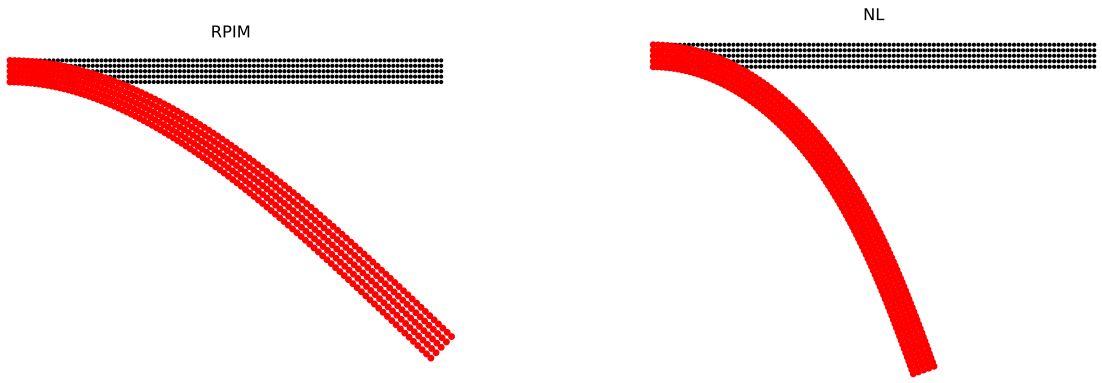


Figure 4.5: Proposed cantilever beam with nonlinear behavior.

The first step is to compare the results obtained when using linear and nonlinear models to solve the nonlinear cantilever. The deformed shape in true scale is given in Figure 4.6. These results are coherent: the linear model presents an amplification proportional do the kinematics of a beam in small deformation while the nonlinear model can predict the rotation along the beam, avoiding the undesired volume increase observed in the linear results. Clearly when using the linear model to solve the cantilever beam of Figure 4.5 we get only a roughly approximation of the beam behavior.



(a) Linear EFG model

(b) Nonlinear EFG model

Figure 4.6: Deformed shape of a nonlinear cantilever beam. While in a) a linear formulation is used, in b) a nonlinear model is employed.

The second step of the validation is to compare the results of EFG and FEM with ANSYS 19.2 solutions. For a quantitative analysis, we will monitor the total displacement of the charged node at $x_1 = l_1$. In Table 4.1, the total displacements are presented for different loads P along with the relative error between EFG, FEM and ANSYS. One may note that the maximum error is less then 2%, which shows that the codes were implemented correctly.

Load P	u_t EFG	u_t FEM	u_t ANSYS	Error EFG-ANSYS	Error FEM-ANSYS
10	-29.523 mm	-29.678 mm	-29.188 mm	1.13 $\%$	1.67 $\%$
20	-49.036 mm	-48.838 mm	-48.403 mm	1.29 $\%$	0.89 $\%$
30	-60.463 mm	-59.956 mm	-59.513 mm	1.57 $\%$	0.75 $\%$
40	-67.472 mm	-66.766 mm	-66.334 mm	1.68 $\%$	0.65 $\%$
50	-72.094 mm	-71.264 mm	-70.793 mm	1.80 $\%$	0.66 $\%$
60	-75.344 mm	-74.436 mm	-74.827 mm	0.68 $\%$	-0.52 $\%$
70	-77.752 mm	-76.795 mm	-77.133 mm	0.79 $\%$	-0.43 $\%$
80	-79.614 mm	-78.624 mm	-78.266 mm	1.70 $\%$	0.46$\%$
90	-81.101 mm	-80.092 mm	-79.765 mm	1.64 $\%$	0.41$\%$
100	-82.324 mm	-81.310 mm	-81.218 mm	1.34 $\%$	0.10$\%$

Table 4.1: Total displacement of the charged node for different loads P, in mm. The last two columns present the relative errors between EFG and ANSYS and FEM and ANSYS.

The proposed nonlinear code for EFG and FEM method presented coherent results, being in agreement with a widely used commercial software. It is inferred that the errors are due to the

difference between the material model adopted in this work – linear St Venant - Kirchhoff – and the one used by ANSYS – logarithmic strain tensor. Finally, the errors are considered small enough to validate the codes implementation.

4.2 Addressed problems

In this section we tackle typical topology optimization problems. Each of these problems is solved using the EFG-BESO and the FEM-BESO. We begin with four linear problems: the two-bar structure, the Michell-type structure, the cantilever beam and the MBB beam. Then, we will work on the nonlinear cantilever beam using the implemented EFG-BESO.

4.2.1 Linear cases

Two-bar

The first addressed problem is the two-bar structure which is maybe the simplest example in topology optimization. The initial domain is a rectangle with $l_1 > l_2$ having all its DOFs at the left-hand side fixed and with a force applied to the middle portion of the right-hand side, as shown in Figure 4.7. In this example, we consider a structure with $l_1 = 10mm$ and $l_2 = 24mm$, the material properties are $E_y^0 = 210MPa$ and $\nu = 0.3$ and the force is $P = 100N$. For lower final volume percentage – around 15% – the final topology will degenerate in a two-bar truss.

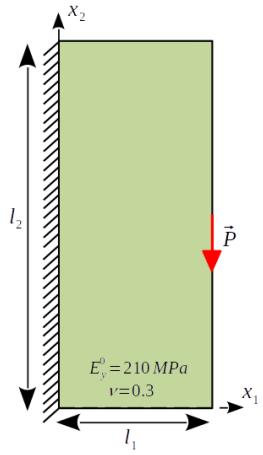


Figure 4.7: Two-bar geometry. All DOFs at left-hand side are fixed and a vertical force is applied to the middle portion of the right-hand side.

To begin, the BESO-FEM algorithm is used to solve the problem. The BESO parameters are: evolutionary ratio $ER = 5\%$, maximum addition ratio $AR_{max} = 2\%$, final weight $W^* = 15\%$, filter radius $r_{min} = 1mm$ and convergence criteria $\tau = 0.01$. The domain is discretized with a mesh of 50×120 *quad4* elements.

The topology results for iterations 2, 7, 26 and 88 are given in Figure 4.8. The material is initially removed from the right-hand corners and, in further iterations, in the middle portion of the left-hand side until, finally, the topology degenerates in a two-bar truss.

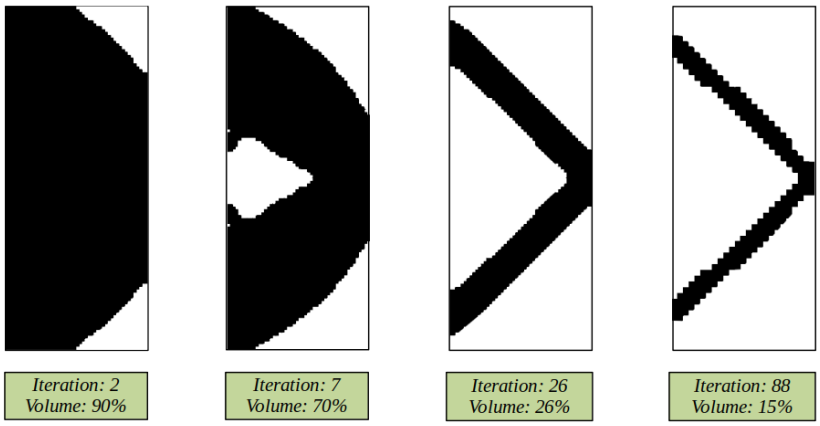


Figure 4.8: Topology results of BESO-FEM for the two-bar structure corresponding to the iterations 2, 7, 26 and 88. The external rectangle with black lines represents the initial design domain

The BESO-EFG uses the same BESO parameters and the domain is discretized with 21×49 nodes and we use a background mesh with 20×48 cells containing four Gauss points per cell. A rectangular influence domain is used for both discretization and dual-level interpolation, and the rectangle size is set to be 2.5 times the distance between nodes in each direction. The influence domain of the dual-level interpolation is identical to the influence domain used in the discretization.

The evolution of the topology is very similar to BESO-FEM's. The final topology obtained with the BESO-EFG is given in Figure 4.9, in which one may note the property $H = 2L$. The results obtained with BESO-FEM and BESO-EFG are in good agreement with (HUANG AND XIE, 2010) and (ZHENG ET AL., 2010), given in Appendix A.1.

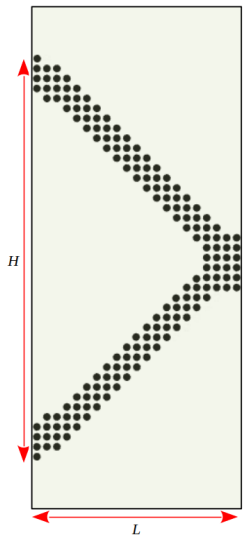


Figure 4.9: Topology results of BESO-EFG for the two-bar. The final volume of the structure is 15%. One may note the property $H = 2L$ of the final topology. The external rectangle represents the initial design domain.

Michell type structures

The second example is the Michell's structure presented in Figure 4.10. All the DOFs at the lower corners are fixed and a force is applied to the middle node at the lower portion. For this example we are going to use $l_1 = 16mm$ and $l_2 = 8mm$, the material properties are $E_y^0 = 210MPa$ and $\nu = 0.3$, the vertical force is $P = 100N$.

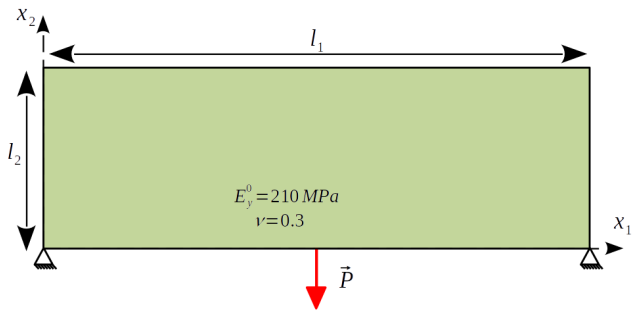
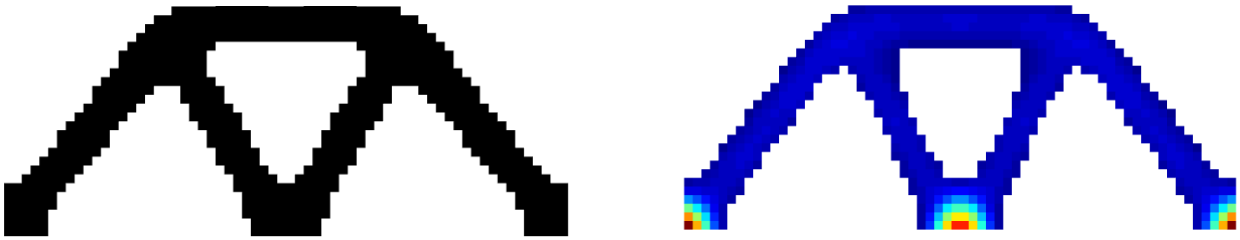


Figure 4.10: Michell type structure.

The BESO-FEM algorithm is applied with the following parameters: $ER = 5\%$, $AR_{max} = 5\%$, $r_{min} = 1mm$, $W^* = 30\%$ and $\tau = 0.01$. The domain is discretized with a mesh of 64×32 elements. The final topology and the final sensitivity are given in Figure 4.11 and a convergence curve, comparing the mean compliance and the structure's volume along the iterations, is presented in Figure 4.12.



(a) Final topology

(b) Final sensitivity

Figure 4.11: Results for the short cantilever beam using BESO-FEM. In Figure 4.17b the red color indicates regions with highest sensitivity numbers and the blue colored regions presents low sensitivity values.

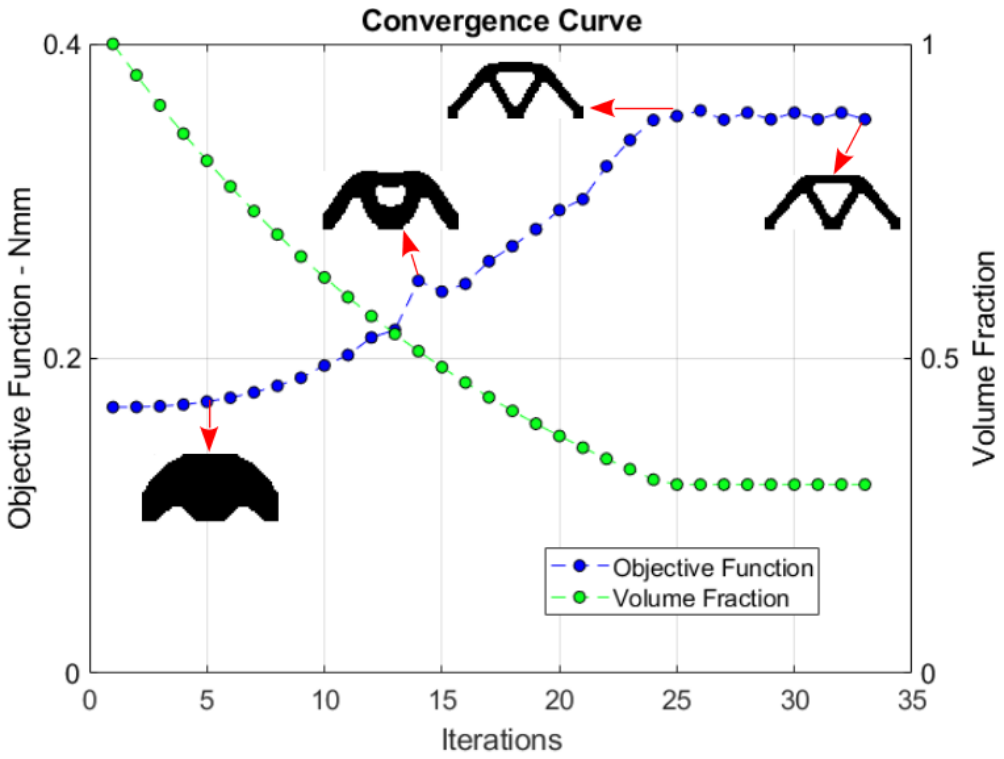


Figure 4.12: Evolutionary convergence curve of BESO-FEM for the Michell's problem. The topologies at iterations 5, 14, 25 and 33 are depicted.

The BESO-EFG discretization uses 64×32 nodes and a background mesh with 32×16 cells containing 4 integration points. The influence domain is rectangular with sizes chosen to be 2.5 times the nodal distance in each direction. The BESO parameters used in this analysis are the same used in BESO-FEM.

The final sensitivity is given in Figure 4.13, where we can observe that the elements with higher sensitivity are next to the boundary conditions. The final topology is given in Figure 4.14, represented in two different manners: the first figure presents the penalized Gauss points used for integration and the second figure shows the discretization nodes. In fact, in the BESO-EFG, the integration points are also penalized as part of the dual-level interpolation scheme when assembling the global stiffness matrix, as illustrated in Figure , of Chapter 3. The gray-scale indicates that the nodes density assumes values between ρ_{min} and 1, which is not true for the BESO-FEM algorithm. The convergence curve of BESO-EFG is given in Figure 4.15.

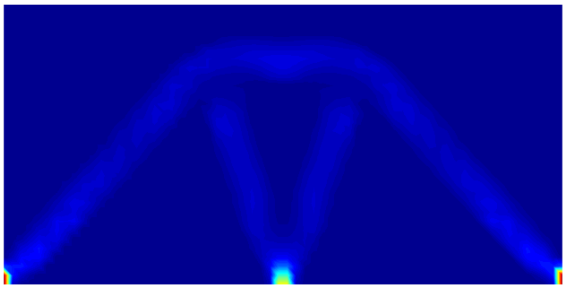
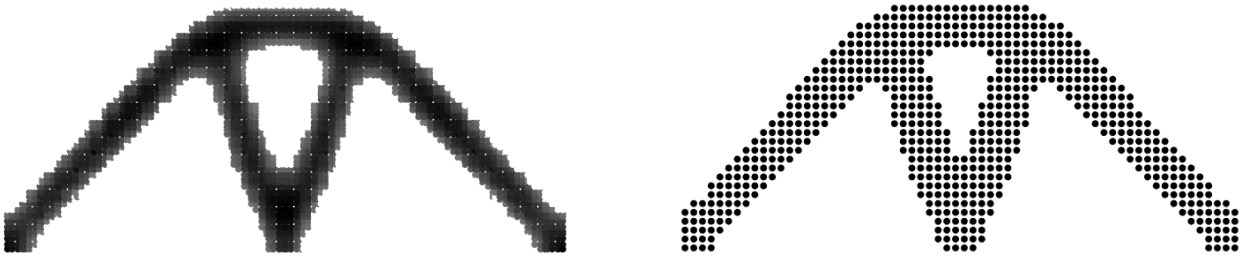


Figure 4.13: Final BESO-EFG sensitivity of the Michell structure.



(a) Gauss integration points

(b) Discretization nodes

Figure 4.14: Topology results of a Michell structure obtained with the BESO-EFG method. In (a) we have the topology of integration points and in (b) we have the topology of discretization nodes. The gray-scale indicates that the nodes density assume values between ρ_{min} and 1.

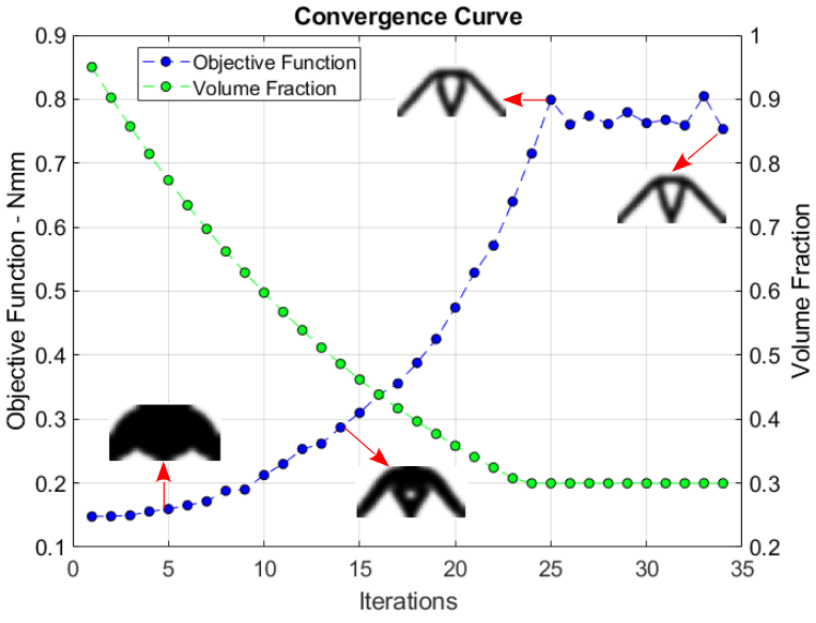


Figure 4.15: Convergence curve of BESO-EFG applied to the Michell structure. The topology at iterations 5, 14, 25 and 34 are detailed.

It can be seen from Figures 4.12 and 4.15 that both methods present a very smooth convergence. However, we can see a difference between the topology in the central triangular portion of the structure: the BESO-FEM provides a larger triangle than BESO-EFG. This tendency can also be observed in the literature results, provided in the Appendix A.2. A possible explanation for the smaller central triangle in BESO-EFG is the size of influence domain.

Cantilever beam

Here we study a classical problem in elastic mechanics: the cantilever beam. It is proposed by HUANG and XIE (2010) the compliance optimization of a short cantilever beam, as shown in Figure 4.16. The beam dimensions are $l_1 = 80mm$ and $l_2 = 50mm$, charged with a punctual force of $P = 100N$ at the right-hand extremity. A fictitious material with $E_y^0 = 100GPa$ and $\nu = 0.3$ is used. This problem is solved with BESO-FEM and BESO-EFG.

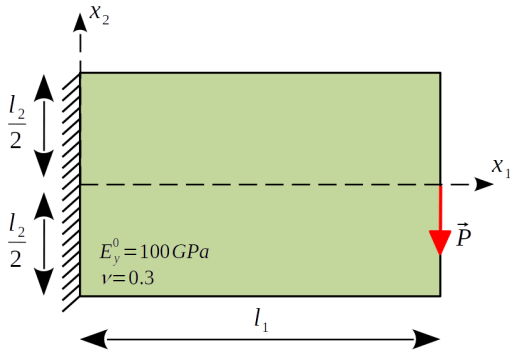
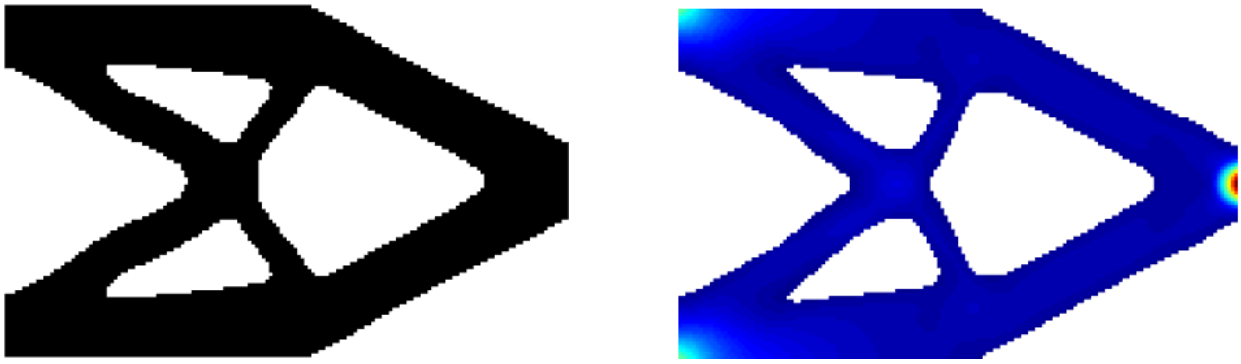


Figure 4.16: Short cantilever beam.

Let us begin with the BESO-FEM. The FEM model uses a mesh of 160×100 elements and the BESO-FEM parameters are: $ER = 2\%$, $AR_{max} = 2\%$, filter radius of $r_{min} = 4mm$, $W^* = 50\%$ and convergence criteria $\tau = 0.001$. The final topology and the final sensitivity are given in Figure 4.17 and the convergence curve is shown in Figure 4.18.



(a) Final topology (b) Sensitivity

Figure 4.17: Results for the short cantilever beam using BESO-FEM. In Figure 4.17b the red color indicates regions with highest sensitivity numbers and the blue colored regions presents low sensitivity values.

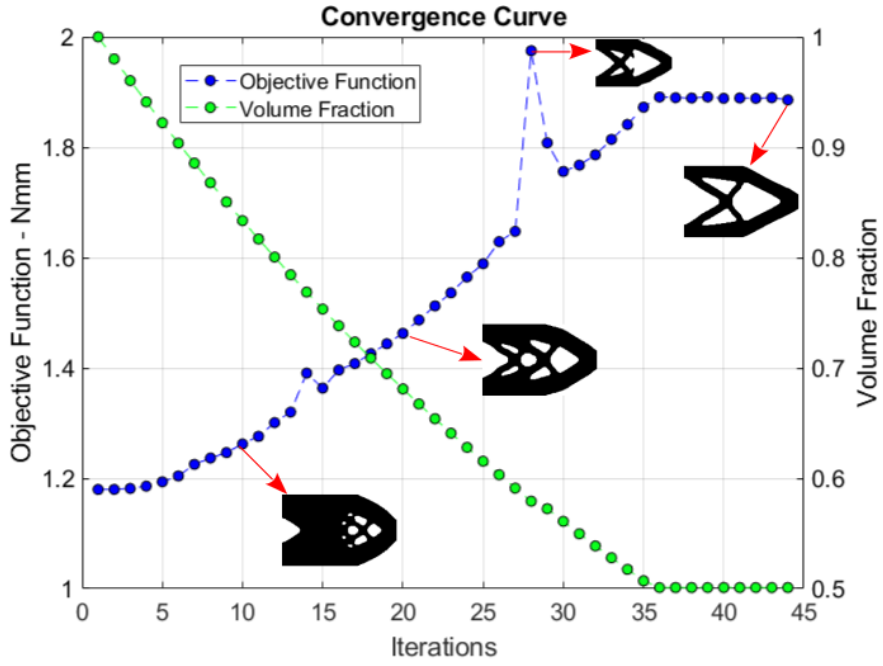


Figure 4.18: BESO-FEM convergence curve for the short cantilever beam. The topology at iterations 10, 20 and 38 are given.

The BESO-EFG algorithm uses a discretization of 96×60 nodes, with a background mesh with 88×55 cells, each cell containing 4×4 Gauss points. The BESO parameters are the same used in BESO-FEM optimization: $ER = 2\%$, $AR_{max} = 2\%$, $W^* = 50\%$, $\tau = 0.001$. It is worth noting

that no extra filtering method is used apart the implicit dual-level interpolation scheme. Again a rectangular influence domain is used with size $2.5dx_1 \times 2.5dx_2$ where dx_1 and dx_2 are the distance between nodes in x_1 and x_2 directions. The dual level interpolation uses the same influence domain size.

The final topology obtained with the BESO-EFG is shown in Figure 4.19 in two versions: (a) the final topology of integration points and (b) the final topology of discretization nodes. In fact, in the BESO-EFG, the integration points are also penalized as part of the dual-level interpolation scheme when assembling the global stiffness matrix, as illustrated in Figure 3.8. The final sensitivity is given in Figure 4.20, while the convergence curve for the BESO-EFG is shown in Figure 4.21.

(a) Gauss integration points

(b) Discretization nodes

Figure 4.19: Topology results of a short cantilever beam obtained with the BESO-EFG method. In $a)$ we have the topology of integration points and in $b)$ we have the topology of discretization nodes. The gray-scale indicates that the nodes density assume values between ρ_{min} and 1.

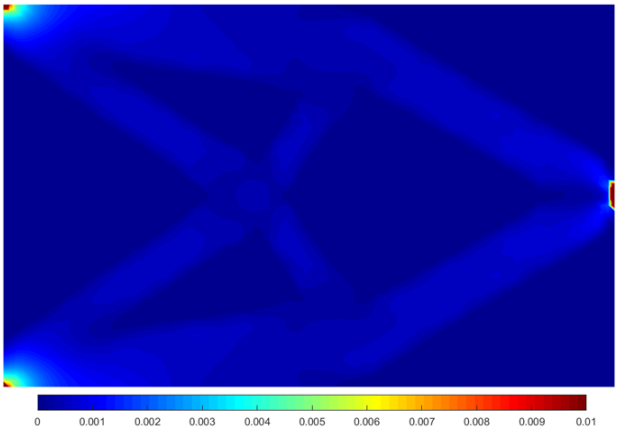


Figure 4.20: Final sensitivity results with BESO-EFG strategy, where the hottest is the color the higher is the sensitivity numbers. The dark blue region represents the void region.

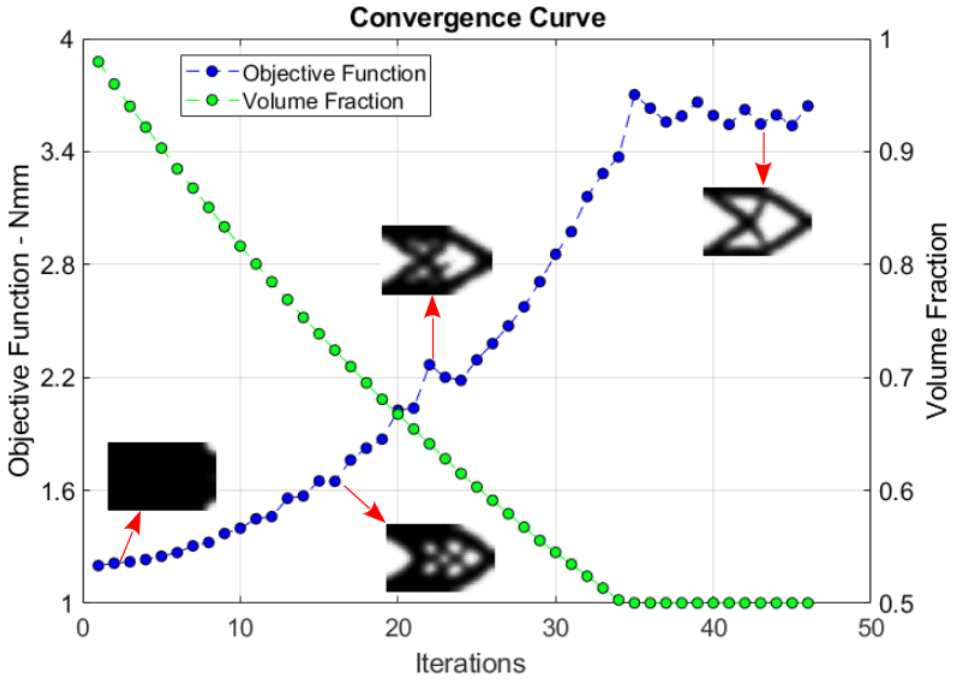


Figure 4.21: BESO-EFG convergence curve for the short cantilever beam. The topology at iterations 2, 16, 22 ans 43 are depicted.

When comparing the convergences curves 4.18 and 4.21, we can remark that the BESO-EFG presents smaller peaks then the BESO-FEM. This is can be attributed to the smooth shape functions of EFG and the continuity of the design variables $\boldsymbol{\rho}$ in the BESO-EFG. Indeed, for the BESO-FEM, we have employed $quad4$ element with bi-linear shape function and if high order elements were

used, the convergence smoothness of BESO-FEM would be improved. High order elements in FEM can also avoid the need of filtering schemes.

The results are in great agreement with the literature results presented in Appendix A.3, specially with those obtained by HUANG and XIE (2010). For this case, the final topology obtained using the BESO-EFG is very similar the topologies obtained using FEM. It can be seen from the convergence curves of BESO-FEM and BESO-EFG that the EFG can provide a smoother convergence.

MBB beam

Another benchmark problem, proposed by (HUANG AND XIE, 2010) and widely used in topology optimization studies is the MBB beam shown in Figure 4.22. The left-hand inferior node has all its DOFs fixed while the right-hand inferior node has only the DOF in x_1 direction fixed. A force is applied in the middle portion of the structure. For this problem we consider a material with Young's modulus $E_y^0 = 200GPa$ and Poisson's coefficient $\nu = 0.3$.

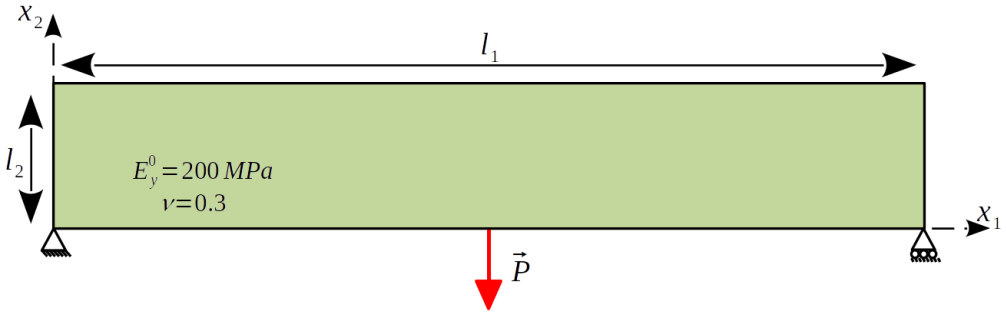


Figure 4.22: MBB beam problem.

Here, we take advantage from the problem's symmetry by modeling only the right-hand side of the structure, as shown in Figure 4.22. At the right-hand side only the DOF in x_1 direction are blocked, allowing a vertical movement of the structure. To apply the symmetry helps the problem convergence and allow finer discretizations.

We begin with the BESO-FEM, using a discretization of 120×40 elements. The BESO parameters are $ER = 2\%$, $AR_{max} = 2\%$, $r_{min} = 4mm$, $W^* = 0.5$ and $\tau = 0.001$. The final

topology is presented in Figure 4.24 and the convergence curve in Figure 4.25.

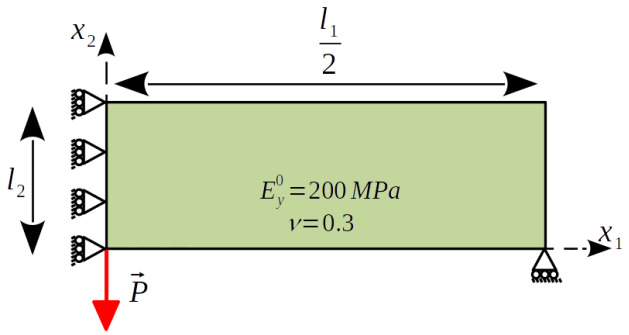


Figure 4.23: MBB beam problem.

Figure 4.24: Final topology of the MBB beam obtained with BESO-FEM.

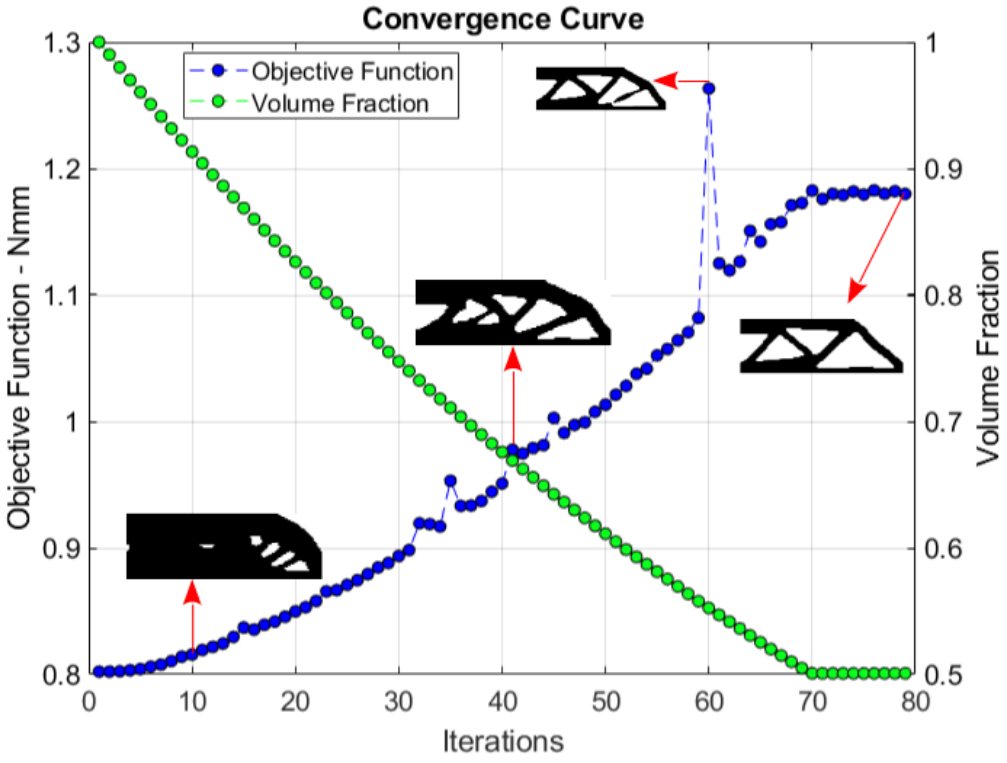


Figure 4.25: BESO-FEM convergence curve for the MBB beam. The topology at iterations 10, 41, 60 and 79 are detailed.

The BESO-EFG uses a discretization with 120×40 nodes and a background mesh with 108×36 integration cells containing 4×4 Gauss points per cell. The BESO parameters are: $ER = 2\%$,$AR_{max} = 2\%$, $W^* = 0.5$ and $\tau = 0.001$. As discussed in the previous problems, a rectangular influence domain with sizes $2.5dx_1 \times 2.5dx_2$ is used for both discretization and dual-level interpolation.

The final topology for Gauss points and discretization nodes is presented in Figure 4.26. As symmetry was applied to this problem, the final topology was duplicated in Figure 4.27 to represent the full structure. The convergence curve is shown in Figure 4.28.

(a) Gauss integration points (b) Discretization nodes

Figure 4.26: Short cantilever results with the BESO-EFG method.

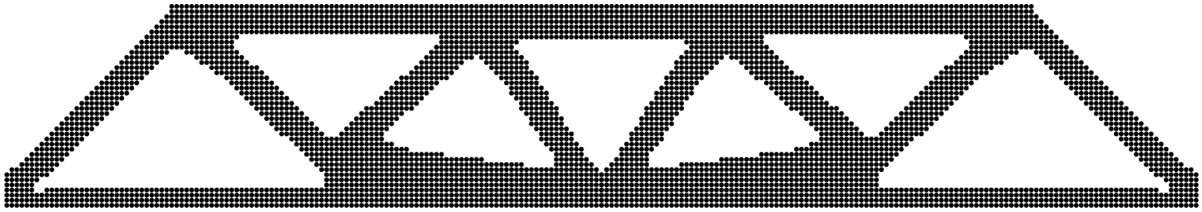


Figure 4.27: Final topology for full MBB beam.

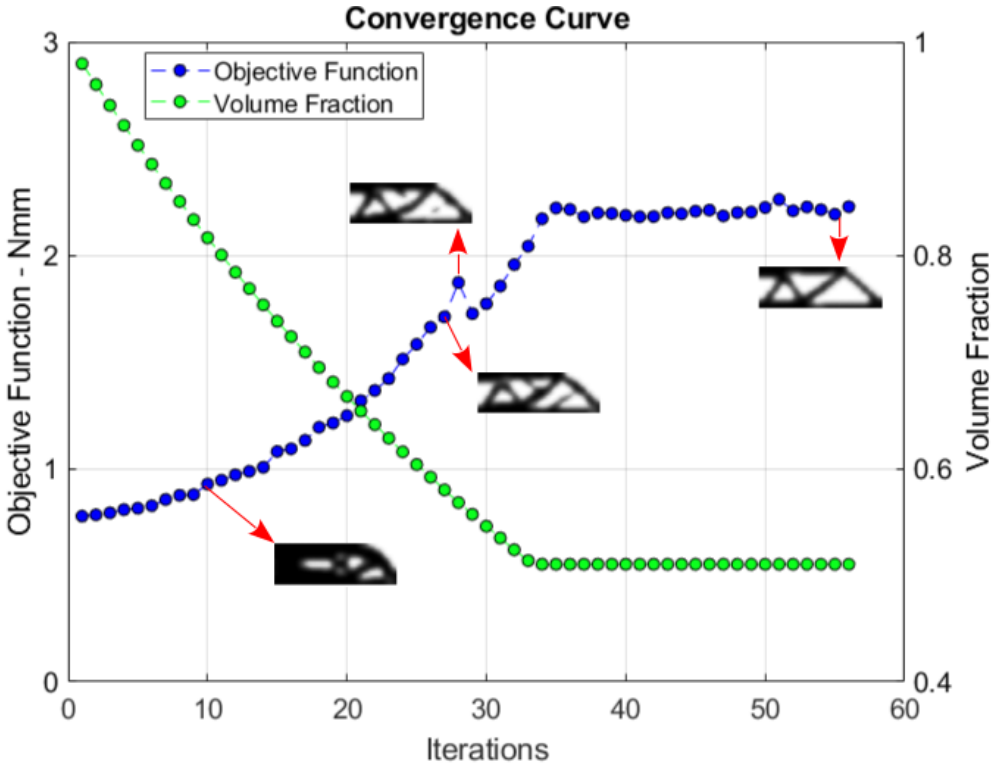


Figure 4.28: BESO-EFG convergence curve for MBB beam.

The final topology obtained with both BESO-EFG and BESO-FEM are in agreement with the results provided by HUANG and XIE (2010) and with other results of literature review, as presented in Appendix A.4. In the final topology we can observe that the material is removed in a triangular pattern that recalls a truss structure. We can also note the presence of a reinforcement at the bottom region of the structure, where the load is applied.

Indeed, the MBB beam presents a more unstable convergence curve than the three precedent examples for both BESO-FEM and BESO-EFG. When comparing the converges curves of Figures 4.25 and 4.28, we note less peaks in the BESO-EFG than in BESO-FEM; the peaks also have smaller amplitudes when using the EFG method. As discussed for the cantilever beam, this characteristic is attributed to the smoothed shape function of meshless methods and the use of continuous design variable in the dual-level interpolation scheme.

4.2.2 Nonlinear cases

Cantilever beam

In this section we apply the nonlinear model developed in the section 2.5 to topology optimization of a long cantilever beam, as shown in Figure 4.29. Here we only use the BESO-EFG algorithm, in an attempt to take advantage from the capacity of the EFG to deal with nonlinear problems (MOLLON, 2016).

The domain dimensions are $l_1 = 1m$ and $l_2 = 0.25m$. An initial discretization of 101×26 nodes and a background mesh with 100×25 integration cells, each containing 4 Gauss points, is used. The material properties are $E_y^0 = 3GPa$ and $\nu = 0.4$. The BESO parameters are: $ER = 1\%$,$AR_{max} = 1\%$, $W^* = 0.5$ and $\tau = 0.001$. Although in linear elastic problems the final topology does not depend on the load, the external forces do change the final topology in nonlinear problems. We will study this influence below.

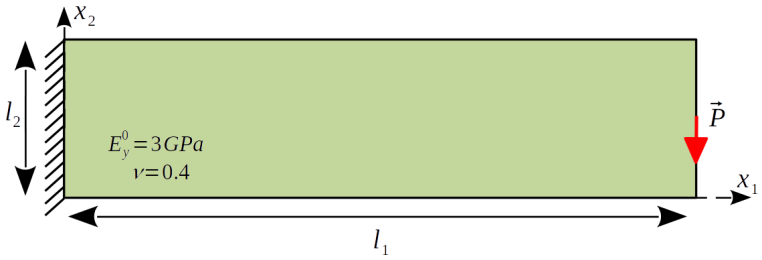
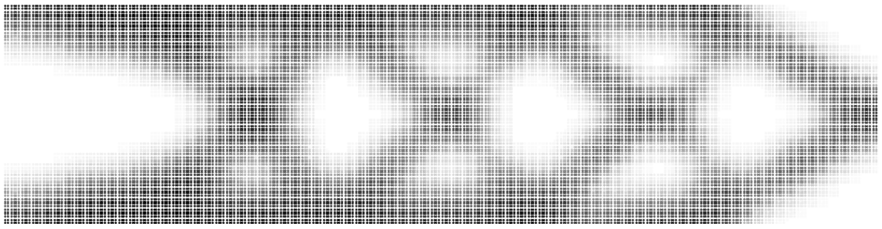


Figure 4.29: Proposed cantilever beam with nonlinear behavior.

It is expected for low forces a linear behavior of the structure (BUHL ET AL., 2000) and (GOMES AND SENNE, 2014). We present in the Figure 4.30 the resulting topology for a force of $P = 100N$, where one can observe the linear cantilever pattern – similar to the results presented in Figures 4.17 and 4.19. Some results of the literature review considering the lower forces scenario are given in Appendix A.5.



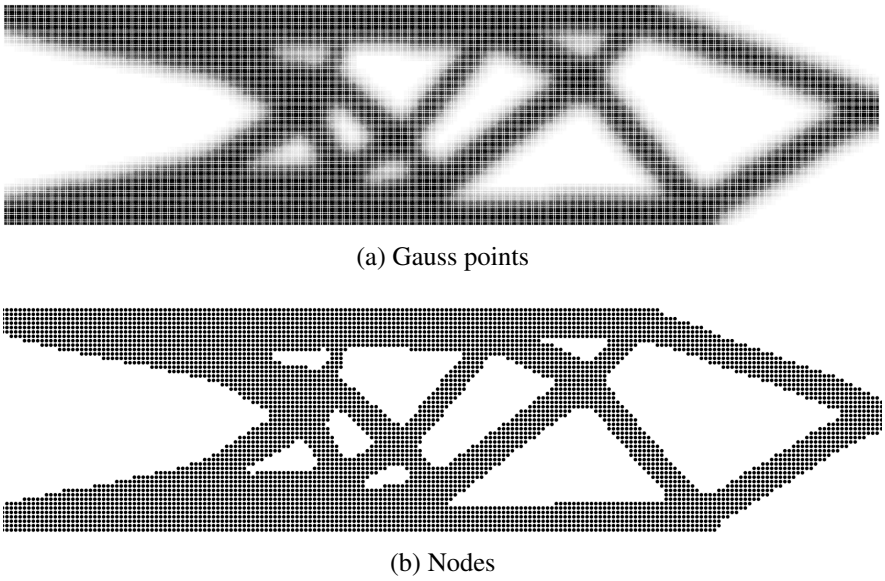
(a) Gauss points

(b) Nodes

Figure 4.30: Topology results for the nonlinear cantilever beam with a low force of $100N$. In this scenario the nonlinear effect is not present, resulting in the linear cantilever topology pattern.

A load of $P = 60kN$ is applied and to ensure the converge we use a finer discretization, containing 201×51 nodes and 120×30 integration cells with 4 Gauss points. The final topology is given in Figure 4.31, where both integration points and discretization nodes are plotted. It can be seen some similarities between the BESO-EFG results and the results provided by HUANG and

XIE (2010), given in Appendix A.5.



(a) Gauss points

(b) Nodes

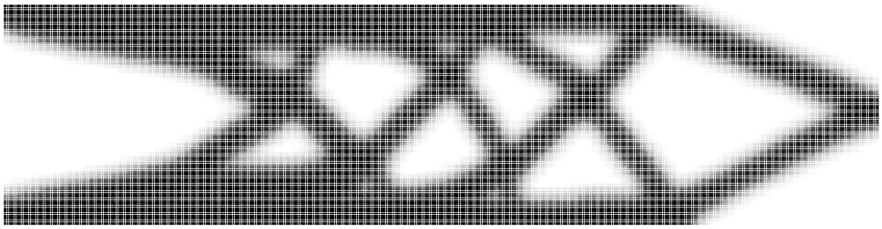
Figure 4.31: Final topology results for the Gauss integration points and discretization nodes for the nonlinear cantilever beam with $P = 60kN$ obtained with BESO-EFG.

To study the influence of different loads in the final topology, the external load is increased to $P = 144kN$ and the results are shown in Figure 4.32. As expected, the final design for $P = 60kN$ and $P = 144kN$ are different, and the effects of nonlinearities are strong in the second case, which is also observed in the works of HUANG and XIE (2010) and GOMES and SENNE (2014). It is worth noting that the discretization is the same for both loads.

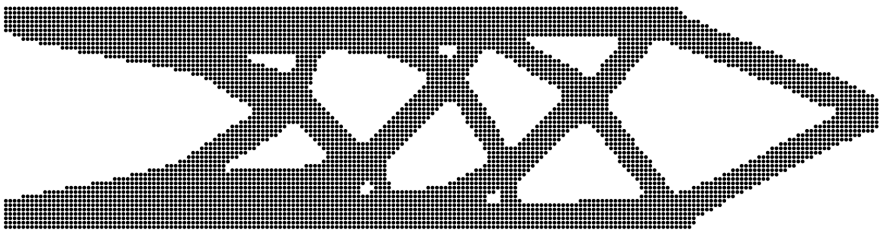
Although some similarities between the BESO-EFG results and those provided by HUANG and XIE (2010) are still present for $P = 144kN$, the results cannot be considered compatible. When comparing the BESO-EFG results with those of (BUHL ET AL., 2000) and (GOMES AND SENNE, 2014) the differences are even bigger, as illustrated in Appendix A.5 in Figures A.11 and A.12.

These results show that the implemented BESO-EFG is not yet capable of dealing with geometrically nonlinear structures with ease. It provides results incompatible with those found in the literature review. However, other authors have shown that the EFG can be successfully applied to the optimization of nonlinear structures (QIZHI ET AL., 2014) and (ZHENG ET AL., 2015). An improved version of the implemented code is envisaged in further works, such as a more effi-

cient implementation that allows finer discretizations for the nonlinear problem and a review on the sensitivity analysis provided in Chapter 3.



(a) Gauss points



(b) Nodes

Figure 4.32: Final topology results for the Gauss integration points and discretization nodes for the nonlinear cantilever beam with $P = 144kN$ obtained with BESO-EFG.

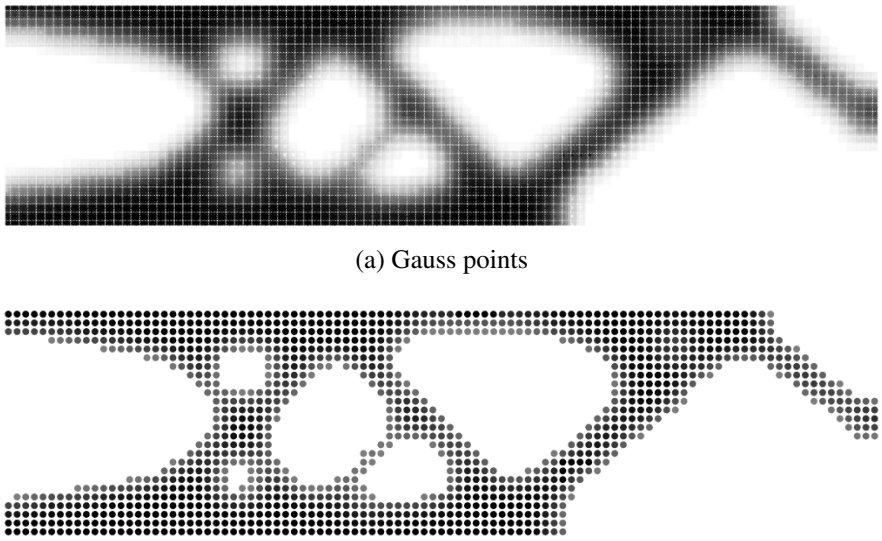
In an attempt to reproduce the results of BUHL et al. (2000), different loads are applied to the structure. The BESO-EFG presented a poor convergence due to the badly scaled tangent matrix, which is expected phenomena in both references (BUHL ET AL., 2000) and (HUANG AND XIE, 2010). For future works strategies to improve the convergence quality can be implemented, such as relaxation of the Newton-Raphson convergence criteria or totally removing low-density elements. For a load of $P = 1000kN$ and using a very low evolutionary ratio of $ER = 0.5\%$ and a damped Newton-Raphson scheme, the topology of Figure 4.33 can be obtained, which comes closer to the results of Appendix A.11 and A.12, which consider an end-compliance minimization. The final topology corresponds to a discretization of 101×26 nodes and a background mesh with 100×25 integration cells containing 4 Gauss points each.

The damped Newton-Raphson uses a factor ζ to reduce the Newton-Raphson step towards the next iteration. The factor is comprised in the interval $[0,1]$. For the topology of Figure 4.33 a factor of $\zeta = 0.5$ is used, which increase the number of iteration for each load step but provides a more stable convergence.

The damping of Newton-Raphson method can be stated as:

$$^{k+1}\mathbf{u} = \zeta \, \boldsymbol{\Delta}\mathbf{u} \, ^{k}\mathbf{u} \tag{4.6}$$

Where $^{k}\mathbf{u}$ is the current displacement, $^{k+1}\mathbf{u}$ is the next iteration initial guess and $\boldsymbol{\delta}$ is the displacement increment obtained with the tangent system of Equation 2.157.



(a) Gauss points

(b) Nodes

Figure 4.33: Final topology results for the Gauss integration points and discretization nodes for the nonlinear cantilever beam with $P = 144kN$ obtained with BESO-EFG.

5 Conclusion

This chapter is dedicated to some concluding remarks.

The main objective of this work is to couple the BESO with the EFG method, in an attempt to encounter alternative methods to the BESO-FEM. The results, detailed in Chapter 4, are in agreement with the results of literature review, showing that the BESO can effectively be used with a meshfree method like the EFG. It could be seen that the BESO-EFG can provide results equivalent to the BESO-FEM and for some cases with a better convergence rate – e.g. the MBB beam.

One of the interests of using the BESO-EFG is to take advantage of the EFG capacity to deal with nonlinear problems. The topology optimization of the nonlinear cantilever beam was not so successful as the optimization of linear problems. Indeed, some similarities could be found between the results of the implemented BESO-EFG with some results of literature review, notably (HUANG AND XIE, 2010). The implemented method still needs improvements to come more closer to the literature reference. Improvements like a more efficient computational implementation to allow finer discretizations, the use of a relaxation method on the Newton-Raphson method, and the total elimination of low-density nodes are feasible scenarios.

Concerning the implementation of the discretization methods for the linear elasticity, the FEM and EFG solvers were validated through a comparison with the analytic results for a cantilever beam. The errors were typically less than 5%. In addition, the nonlinear solvers were successfully validated trough a comparison with Ansys results for a geometrically nonlinear cantilever beam. As discussed in Chapter 4, the results provided by the implemented codes are considered acceptable.

The filtering scheme used in the BESO-FEM is very similar to the dual-level interpolation employed in the BESO-EFG. Although no explicit filtering scheme is used in BESO-EFG, the dual-level interpolation, for all effects, is a way of filtering. In this way, the no need of filtering scheme cannot be presented as an asset of the BESO-EFG method.

In general lines, the objectives established for this work were achieved. This work can be considered as a primary exploration of the BESO-EFG method and its main aspects. It was shown that the BESO-EFG can be an alternative to BESO-FEM, presenting consistent results for linear elastic cases.

References

ABDI, M., ASHCROFT, I., and WILDMAN, R. **Topology optimization of geometrically nonlinear structures using an evolutionary optimization method**. *Engineering Optimization*, pages 1850–1870, 11 2017. doi: 10.1080/0305215X.2017.1418864.

Amirfakhrian, M. **Image Reconstruction by Moving least Squares**. 01 2013.

ANDREASEN, C., ELINGAARD, M., and AAGE, N. **Level set topology and shape optimization by density methods using cut elements with length scale control**. *Structural and Multidisciplinary Optimization*, 62, 08 2020. doi: 10.1007/s00158-020-02527-1.

BELYTSCHKO, T., LU, Y. Y., and GU, L. **Element-free Garlekin methods**. *International Journal for Numeric Methods in Engineering*, 37:229–256, 1994.

BENDSOE, M. **Optimal shape design as a material distribution problem**. *Structural Optimization*, 1:193–202, 1989.

BENDSOE, M. and KIKUCHI, N. **Generating optimal topologies in structural design using a homogenization method**. *Computer methods in applied mechanics and engineering*, 71:197–224, 1988.

BENDSOE, M. and SIGMUND, O. **Material interpolation schemes in topology optimization**. *Archive of Applied Mechanics*, 69:635–654, 11 1999. doi: 10.1007/s004190050248.

BUHL, T., PEDERSEN, C., and SIGMUND, O. **Stiffness design of geometrically nonlinear structures using topology optimization**. *Structural and Multidisciplinary Optimization*, 19:93–104, 04 2000. doi: 10.1007/s001580050089.

BURGER, M., HACKL, B., and RING, W. **Incorporating topological derivatives into level set methods**. *Journal of Computational Physics*, 194:344–362, 2004.

CHO, S. and KWAK, J. **Topology design optimization of geometrically non-linear structures using meshfree methods**. *Computer Methods in Applied Mechanics and Engineering*, 195:5909–5925, 2006.

CUNHA, D. C. ***Otimização Topológica na Concepção de Fluido Atuadores Celulares***. PhD thesis, Universidade Estadual de Campinas, 2019.

CUNHA D.C, PAVANELLO, R. **Evolutionary Topology Optimization for Designing Cellular Fluid Actuators**. *Advances in Structural and Multidisciplinary Optimization.*, WCSMO 2017: 1484–1496, 2018.

DE ALMEIDA, B., CUNHA, D., and PAVANELLO, R. **Topology optimization of bimorph piezoelectric energy harvesters considering variable electrode location**. *Smart Materials and Structures*, 28, 2019. doi: 10.1088/1361-665X/ab2c3e.

DOLBOW, J. and BELYTSCHKO, T. **An Introduction to Programming the Meshless Element-free Garlekin Method**. *Archives of Computational Methods in Engineering*, 5,3:207–241, 1998.

DU, Y., LUO, Z., TIAN, Q., and CHEN, L. **Topology optimization for thermo-mechanical compliant actuators using mesh-free methods**. *Engineering Optimization*, 41:753–772, 2009.

GINGOLD, L. and MONAGHAN, J. J. **Smooth particle hydrodynamics: theory and application to non-spherical stars**. *Monthly Notices of the Royal Astronomical Society*, 181, 1977.

GINGOLD, L. and MONAGHAN, J. J. **Why particle methods work**. *SIAM Journal on Scientific Computing*, 3, 1982.

GOMES, F. and SENNE, T. **An algorithm for the topology optimization of geometrically non-linear structures**. *International Journal for Numerical Methods in Engineering*, 99:1–18, 08 2014.

doi: 10.1002/nme.4686.

HAJIAZIZI, M. and BASTAN, P. **The elastoplastic analysis of a tunnel using the EFG method: A comparison of the EFGM with FEM and FDM**. *Applied Mathematics and Computation*, 234: 82–113, 2014.

HUANG, X. and XIE, Y. ***Evolutionary Topology Optimization of Continuum Structure - methods and applications***. Wiley & Sons, 2010.

IBANES, D., ORDEN, J., BRANAS, B., CARMONA, J., and MOLLA, J. **Application of Galerkin meshfree methods to nonlinear thermo-mechanical simulation of solids under extremely high pulsed loading**. *Fusion Engineering and Design*, 88:2744–2747, 2013.

KIKUCHI, N., NISHIWAKI, S., FONSECA, J., and SILVA, E. **Design optimization method for compliant mechanisms and material microstructure**. *Computer Methods in Applied Mechanics and Engineering*, 151:401–417, 01 1998. doi: 10.1016/S0045-7825(97)00161-8.

KIM, N. ***Introduction to Nonlinear Finite Element Analysis***. Springer, 2015.

KWON, Y. and BANG, H. ***The Finite Element Method Using MATLAB***. CRC Press, 2000.

LIBERSKY, L. and PETSCHEK, A. G. **Smoothed Particle Hydrodynamics with strength of materials**. *Proceedings of the Next Free Lagrange Conference*, 395, 1991.

LIBERSKY, L. and PETSCHEK, A. G. **High Strain Lagrangian Hydrodynamics: A Three-Dimensional SPH Code for Dynamic Material Response**. *Journal of Computational Physics*, 109, 1993.

LIU, G. R. ***Mesh free methods: moving beyond the finite element method***. CRC Press, 2003.

LOPES, H., PAVANELLO, R., and MAHFOUD, J. **Topological Optimization for the Max-**

imization of Frequency Separation Margin. 04 2017. doi: 10.26678/ABCM.MecSol2017.MSL17-0145.

LUCY, L. **A numerical approach to testing the fission hypothesis**. *Astronomy Journal*, 82, 1977.

MOLLON, G. **A multibody meshfree strategy for the simulation of highly deformable granular materials**. *Int. J. Meth. Engng*, 108(12):1477–1497, 2016.

MOLLON, G. **A unified numerical framework for rigid and compliant granular materials**. *Computational Particle Mechanics*, 5:517–527, 2018.

NAYROLES, B., TOUZOT, G., and VILLON, P. **Generalizing the finite element method: diffuse approximation and diffuse elements**. *Comput. Mech.*, 10, 1992.

Nealen, A. **An As-Short-As-Possible Introduction to the Least Squares, Weighted Least Squares and Moving Least Squares Methods for Scattered Data Approximation and Interpolation**. 01 2004.

NGUYEN, V., RABCZUK, T., BORDAS, S., and DUFLOT, M. **Meshless methods: A review and computer implementation aspects**. *Mathematics and Computers in Simulation*, 79:763–813, 2008.

NISHIWAKI, S., FRECKER, M., MIN, S., and KIKUCHI, N. **Topology optimization of compliant mechanisms using the homogenization method**. *International Journal for Numerical Methods in Engineering*, 42:535–559, 06 1998.

of Mathematics, E. **Variation of a functional**, 2011. URL `https://www.encyclopediaofmath.org/index.php/Variation_of_a_functional`.

OLIVER, X. and AGELET, C. ***Continuum Mechanics for Engineers. Theory and Problems***. 2017.

OSHER, S. and FEKIW, R. **Level Set Methods: An Overview and Some Recent Results**. *Journal of Computational Physics*, 169(2):463–502, 2001. doi: https://doi.org/10.1006/jcph.2000.6636.

OVERVELDE, J. T. B. ***The Moving Nodes Approach in Topology Optimization***. PhD thesis, Delft University of Technology, 2012.

PICELLI, R. ***Evolutionary Topology Optimization of Fluid-structure Interaction Problems***. PhD thesis, Universidade Estadual de Campinas, 2015.

QIZHI, H., KANG, Z., and WANG, Y. **A topology optimization for geometrically nonlinear structures with meshless analysis and independent density field interpolation**. *Computational Mechanics*, 54:629–644, 2014.

RUGGIERO, M. and LOPES, V. ***Cálculo Numérico: Apectos Teóricos e Computacionais***. Pearson, 1996.

SALARI, H. and DIZADJI, M. **Meshless simulation of rock mechanics problems by Element Free Galerkin Method**. *Harmonising Rock Engineering and the Environment - Proceedings of the 12th ISRM International Congress on Rock Mechanics*, pages 517–521, 2012. doi: 10.1201/b11646-90.

SHOBEIRI, V. **The topology optimization design for cracked structures**. *Engineering Analysis with Boundary Elements*, 58:26–38, 2015.

SHOBEIRI, V. **Topology optimization using bi-directional evolutionary structural optimization based on the element-free Galerkin method**. *Engineering Optimization*, 48:380–396, 2016.

STOLPE, M. and SVANBERG, K. **An alternative interpolation scheme for minimum compliance topology optimization**. *Structural and Multidisciplinary Optimization*, 22:116–224, 2001.

STOVER, C. **Gâteaux Derivative - From MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein.**, 2021. URL `https://mathworld.wolfram.com/`

GateauxDerivative.html. Online; accessed 28-March-2021.

SVANBERG, K. **The method of moving asymptotes - a new method for structural optimization**. *International journal for numerical methods in engineering*, 24:359–373, 1987.

TIMOSHENKO, S. and GOODIER, J. ***Theory of Elasticity***. McGraw-Hill, 1970.

VICENTE, W., PICELLI, R., PAVANELLO, R., and XIE, Y. **Topology optimization of frequency responses of fluid–structure interaction systems**. *Finite Elements in Analysis and Design*, 98: 1–13, 02 2015. doi: 10.1016/j.finel.2015.01.009.

WANG, M. Y., WANG, X., and GUO, D. **A level set method for structural topology optimization**. *Computer methods in applied mechanics and engineering*, 192:227–246, 2003.

XIE, Y. and STEVENS, G. **A simple evolutionary procedure for structural optimization**. *Computer and Structures*, 49:885–896, 1993.

YANG, X., ZHENG, J., and LONG, S. **Topology optimization of continuum structures with displacement constraints based on meshless methods**. *International Journal of Mechanics and Materials in Design*, 13:311–320, 2017.

ZHAO, F. **Topology optimization with meshless density variable approximants and BESO method**. *Computer-Aided Design*, 56:1–10, 2014.

ZHENG, J., LONG, S., and LI, G. **The topology optimization design for continuum structures based on the element free Galerkin method**. *Engineering Analysis with Boundary Elements*, 34: 666–672, 2010.

ZHENG, J., YANG, X., and LONG, S. **Topology optimization with geometrically non-linear based on the element free Galerkin method**. *International Journal for Numerical Methods in Engineering*, 11:231–241, 2015.

ZHOU, J. and ZOU, W. **Meshless approximation combined with implicit topology description for optimization of continua**. *Structural and Multidisciplinary Optimization*, 36:347–353, 2008.

ZHOU, M. and ROZVANY, G. **On the validity of ESO type methods in topology optimization**. *Structural and Multidisciplinary Optimization*, 21:80–83, 03 2001. doi: 10.1007/s001580050170.

APPENDIX A – Topology results of literature review

This appendix gives the literature results for the addressed problems of Chapter 4.

A.1 Two-bar

The results proposed by HUANG and XIE (2010) using a ESO-FEM algorithm for the two-bar problem is given in Figure A.1. For this case, the two-bar structure was optimized for stress distribution over the domain. Although this is not a proper comparison, one may note a solid similarities to the results of Figure 4.9.

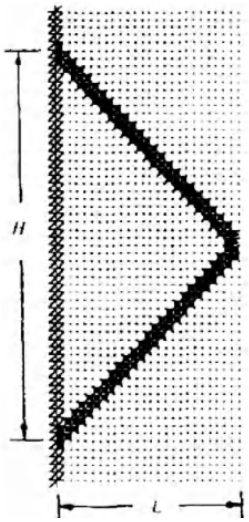


Figure A.1: Two-bar results proposed by (XIE AND STEVENS, 1993) and (HUANG AND XIE, 2010).

A second example is the two-bar results of (ZHENG ET AL., 2010) which also uses the dual-level interpolation method with the BESO-EFG method.

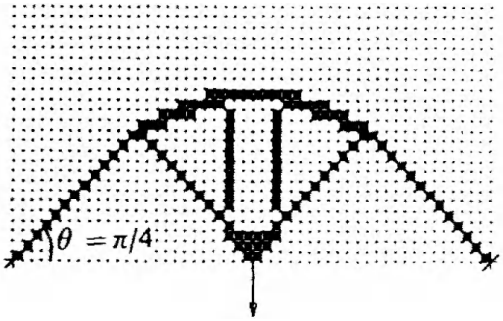
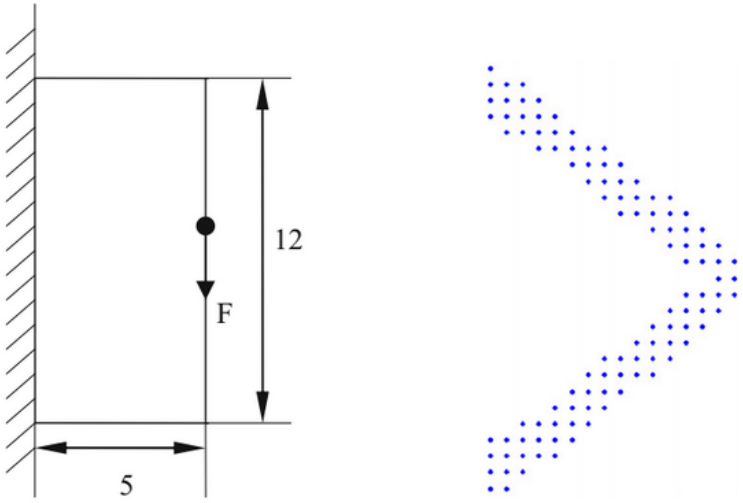


Figure A.3: Results proposed by (XIE AND STEVENS, 1993) for the Michell type structure using the ESO-FEM method. The structure's final volume is $W^* = 15\%$.



(a) Initial geometry (b) Final topolgy

Figure A.2: Results for the two-bar proposed by (ZHENG ET AL., 2010), which also uses the BESO-EFG with the dual-level interpolation method. In (a) we have the initial geometry and the final topology is given in (b).

A.2 Michell type structure

The topology considering a final volume of $W^* = 15\%$ in given in Figure A.3. This result was obtained using a ESO-FEM method for stress optimization.

Another example is the results obtained by (SHOBEIRI, 2015), presented in Figure A.4, considering a final weight of $W* = 30\%$. In his work a BESO-EFG method is employed with-

out any interpolation method between nodes and integration points. The final topology remains however similar the results presented in Figure 4.14.

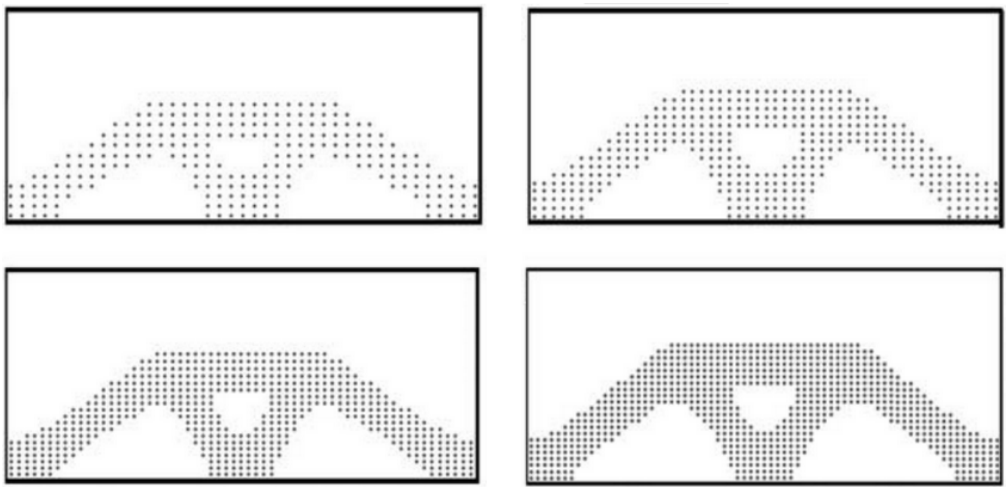


Figure A.4: Results proposed by (SHOBEIRI, 2015) for the Michell type structure, considering different discretizations. Starting from the top-right figure, the discretizations are 41×21; 51×26; 61×31 and 71×36.

A.3 Cantilever beam

The cantilever results using a BESO-FEM algorithm for compliance optimization obtained by HUANG and XIE (2010) is given in Figure A.5. The final topology presents a volume o $W^* = 50\%$. In Figure A.6 the final topology obtained with the SIMP method for the same problem is presented. The problem parameters are the same of Figure 4.16.

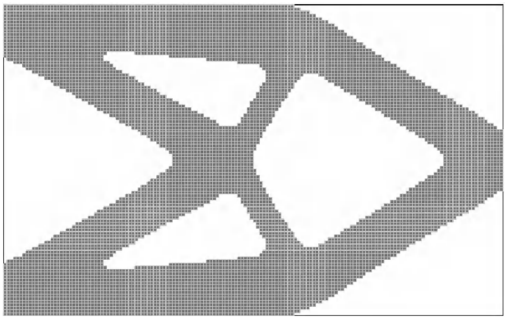


Figure A.5: Final topology of the cantilever beam obtained by HUANG and XIE (2010) using a BESO-FEM algorithm. The final topology presents a volume $W^* = 50\%$.

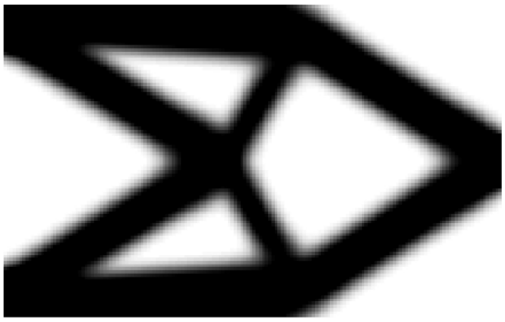


Figure A.6: Final topology of the cantilever beam using the SIMP method, provided by HUANG and XIE (2010).

A.4 MBB

The MBB beam results obtained by (HUANG AND XIE, 2010) using the BESO-FEM algorithm is presented in Figure A.7. The SIMP results for the same problem are provided in Figure A.8. The problem parameters are the same of Figure 4.22.

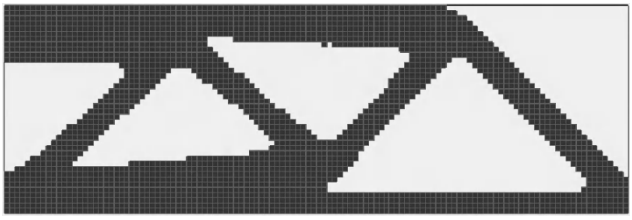


Figure A.7: Final MBB topology obtained by HUANG and XIE (2010) using the BESO-FEM for compliance optimization.

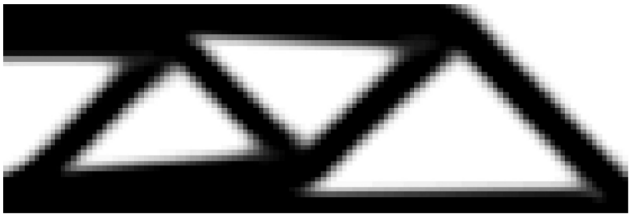


Figure A.8: Final topology obtained with the SIMP method, provided by HUANG and XIE (2010).

In his work SHOBEIRI (2016) also provides results for the MBB beam, given in Figure A.9. This results were obtained with a BESO-EFG method without dual-level interpolation between

discretization nodes and Gauss points. Using the symmetry, a discretization of 151×51 nodes is used.

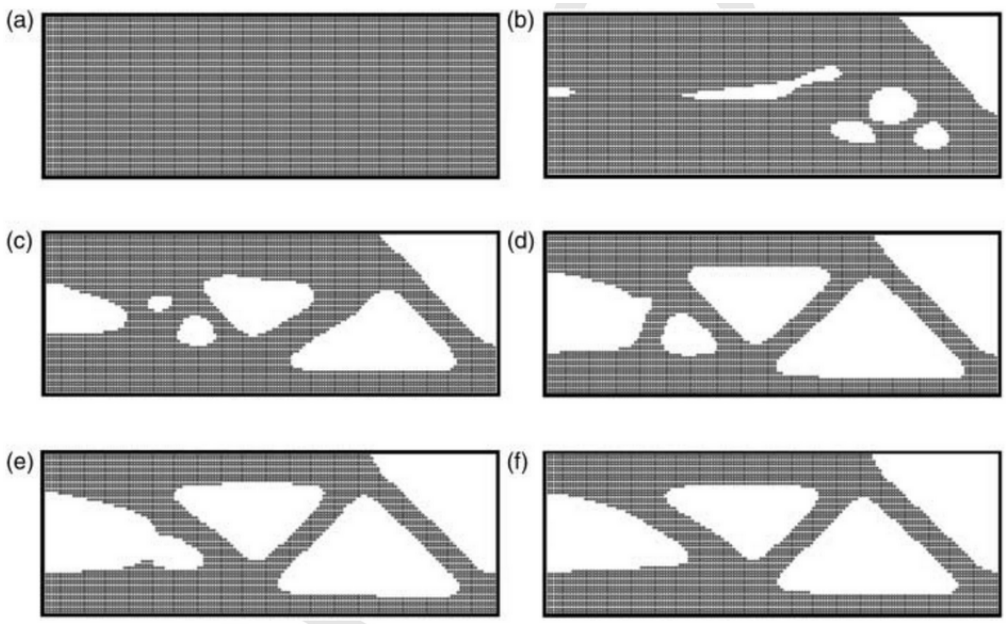


Figure A.9: SHOBEIRI (2016) results for the MBB beam using a BESO-EFG method. The topology at iterations 4, 9, 14, 22 and 34 are depicted.

A.5 Nonlinear cantilever beam

The results obtained by HUANG and XIE (2010) for the nonlinear cantilever beam of Figure 4.29 using the BESO-FEM are given in the Figure A.10. These results are similar to those of Figures 4.31 and 4.32.

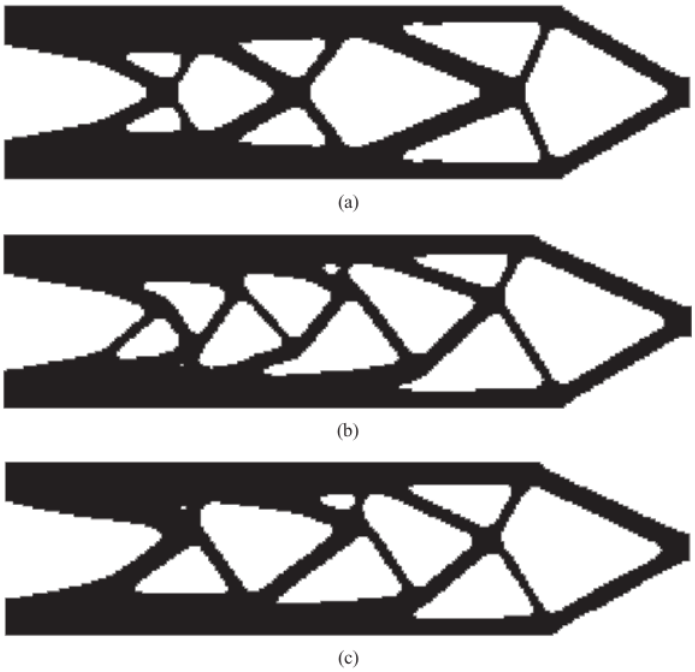


Figure A.10: Final topology obtained by (HUANG AND XIE, 2010) using BESO-FEM in end-compliance optimization. In (a) we have final topology using a linear elastic model, in (b) a non-linear model is used and the applied force is $P = 60kN$, and in (c) a nonlinear model is used with an applied force of $P = 144kN$.

The results obtained by (BUHL ET AL., 2000) are given in Figure A.11. One may notice that this result is considerably different from those of Figures 4.31 and 4.32. Another interesting result obtained by GOMES and SENNE (2014) using a method called *Sequential Piecewise Linear Programming* – SPLP – is given in Figure A.12.

(a) Initial geometry (b) Final topolgy

Figure A.11: Results obtained by (BUHL ET AL., 2000) for the nonlinear cantilever beam shown in (a). The load is $P = 144kN$ and a fictitious material of $E_y^0 = 3GPa$ and $\nu = 0.4$ is used.

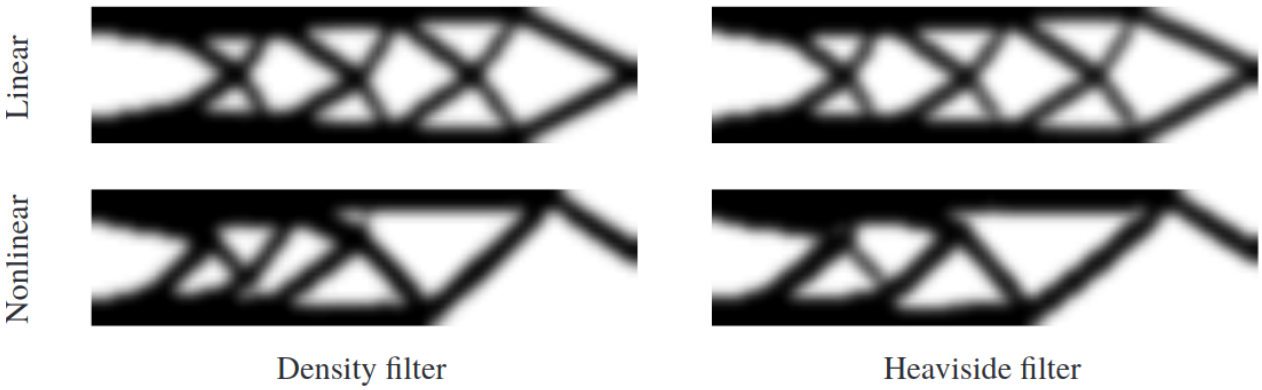


Figure A.12: Final topologies for the nonlinear cantilever beam obtained by GOMES and SENNE (2014) for different filtering schemes. The *linear* results are obtained with a small displacement model.