Hugo Bastos de Sá Bruno

# Shape Optimization with Symmetric Galerkin Boundary Element Method

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Engenharia Civil of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Civil.

Advisor : Prof. Luiz Fernando Campos Ramos Martha
Co-Advisor: Prof. Ivan Fábio Mota de Menezes

Rio de Janeiro
March 2017

**Hugo Bastos de Sá Bruno**

# Shape Optimization with Symmetric Galerkin Boundary Element Method

Dissertation presented to the Programa de Pós–graduação em Engenharia Civil of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Civil. Approved by the undersigned Examination Committee.

**Prof. Luiz Fernando Campos Ramos Martha**
Advisor
Departamento de Engenharia Civil e Ambiental – PUC-Rio

**Prof. Ivan Fábio Mota de Menezes**
Co-Advisor
Departamento de Engenharia Mecânica – PUC-Rio

**Prof. Ney Augusto Dumont**
Departamento de Engenharia Civil e Ambiental – PUC-Rio

**Prof. Evandro Parente Júnior**
Universidade Federal do Ceará

**Prof. Márcio da Silveira Carvalho**
Vice Dean of Graduate Studies
Centro Técnico Científico – PUC-Rio

Rio de Janeiro, March 6th, 2017

**Hugo Bastos de Sá Bruno**

The author graduated in Civil Engineering from Universidade Federal Fluminense - UFF in 2014.

Bibliographic data

Bruno, Hugo Bastos de Sá

Shape Optimization with Symmetric Galerkin Boundary Element Method / Hugo Bastos de Sá Bruno; advisor: Luiz Fernando Campos Ramos Martha; co-advisor: Ivan Fábio Mota de Menezes. – 2017.

v., 75 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Civil e Ambiental.

Inclui bibliografia

1. Civil Engineering – Teses. 2. Otimização de Forma. 3. Método dos Elementos de Contorno. 4. Programação Cônica Quadrática. 5. Concentração de tensões. I. Martha, Luiz Fernando Campos Ramos. II. Menezes, Ivan Fábio Mota. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Civil e Ambiental. IV. Título.

CDD: 624

## Acknowledgment

To professors Luiz Fernando Martha and Ivan Fábio Mota de Menezes for the enlightening discussions and contributions.

To CNPq and PUC-Rio, for the finantial support, whithout which this work could not have been carried out.

To my family for all the support and caring.

To my girlfriend for the patience, encouragement and understanding.

To professors Ney Augusto Dumont e Evandro Parente Júnior for the relevant suggestions and corrections.

## Abstract

Bruno, Hugo Bastos de Sá ; Martha, Luiz Fernando Campos Ramos (Advisor); Menezes, Ivan Fábio Mota (Co-Advisor). **Shape Optimization with Symmetric Galerkin Boundary Element Method**. Rio de Janeiro, 2017. 75p. Tese de Dissertação – Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro.

In this work a numerical implementation of shape optimization in two-dimensional linear elasticity problems is proposed. The main goal is to propose a robust and efficient methodology for the solution of shape optimization problems regarding the minimization of stress concentration effects. In the proposed implementation, the structural analysis is performed by the Symmetric Galerkin Boundary Element Method (SGBEM), thus disposing of the mesh generation burden. The boundary stress evaluation is carried out by an accurate approach which is ideally suited for problems with stress concentrations. Another relevant feature of the proposed implementation is a suitable partition of the SGBEM equations which aims at reducing the computational effort associated with the structural analysis stage. The solution for the optimization problem is obtained by means of a modern numerical optimization method, the so-called Second Order Conic Programming (SOCP). Specifically, the solution for the non-linear optimization is sought by solving a sequence of SOCP subproblems.

## Keywords

Shape Optimization; Boundary Element Method; Second-order Conic Programming; Stress concentrations.

## Resumo

Bruno, Hugo Bastos de Sá ; Martha, Luiz Fernando Campos Ramos; Menezes, Ivan Fábio Mota. **Otimização de Forma com o Método de Elementos de Contorno Simétrico de Galerkin**. Rio de Janeiro, 2017. 75p. Dissertação de Mestrado – Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro.

Esse trabalho propõe uma implementação numérica para otimização de forma em problemas bi-dimensionais de elasticidade. O objetivo principal é propor uma metodologia eficiente e robusta para solução de problemas de otimização de forma considerando a minimização de concentração de tensões. Na implementação proposta, a análise estrutural é realizada pelo Método dos Elementos de Contorno Simétrico de Galerkin (MECSG), evitando-se assim a dispendiosa etapa de geração da malha. A avaliação das tensões no contorno é obtida por meio de um método preciso, ideal para problemas com concentrações de tensões. Outro aspecto relevante na implementação é a adequada partição das equações do MECSG de forma a reduzir, consideravelmente, o esforço computacional associado à etapa da análise estrutural. O problema de otimização é resolvido utilizando-se um método de otimização moderno, conhecido como Programação Cônica de Segunda Orderm (PCSO). Especificamente, busca-se a reposta do problema de otimização não linear por meio da solução de uma sequência de subproblemas de PCSO.

## Palavras-chave

Otimização de Forma; Método dos Elementos de Contorno; Programação Cônica Quadrática; Concentração de tensões.

# Table of contents

# List of figures

*There is no way around hard work, embrace it.*

**Roger Federer**, *Lessons from The Court.*

# 1
# Introduction

The structural design process, traditionally carried out by the so-called *trial-and-error* method, is a methodical investigation in which the engineer employs its own experience and intuition in order to obtain an affordable design which complies with certain criteria. However, there may exist a great number of feasible designs, each of which with its own pros and cons. On the other hand, modern engineering design has now turned its attention in finding optimal solutions, which may be defined in terms of cost, weight, resistance or a combination of these. Evidently, the *trial-and-error* method is no longer a suitable approach to obtain such solutions. To overcome this limitation, the design process is often formulated as a mathematical programming problem for which the solution may be sought by robust and efficient methods. In order to enable for solutions of real-life complex problems, numerical approaches are generally employed. Such approaches usually rely on combining mathematical programming algorithms with structural analysis methods. This class of formulations belongs to an area of study called structural optimization.

Structural optimization consists of finding the best structure, in terms of some design variables, in the sense that a given performance measure is optimized while still satisfying some design constraints. Depending on the choice of the design parameters the problem may be classified according to one of the following branches:

- **sizing optimization**: the parameters refers to some type of structural thickness or a typical size (for example, optimization of the beams' cross sectional area of a truss structure)
- **shape optimization**: the shape of a structure is optimized without changing its topology (for example, optimization of a perforated plate by changing the geometry of the holes)
- **topology optimization**: the structure is optimized by changing its topology (for example, creating holes in a continuum media )

Figure 1.1 depicts an example of each one of these branches.

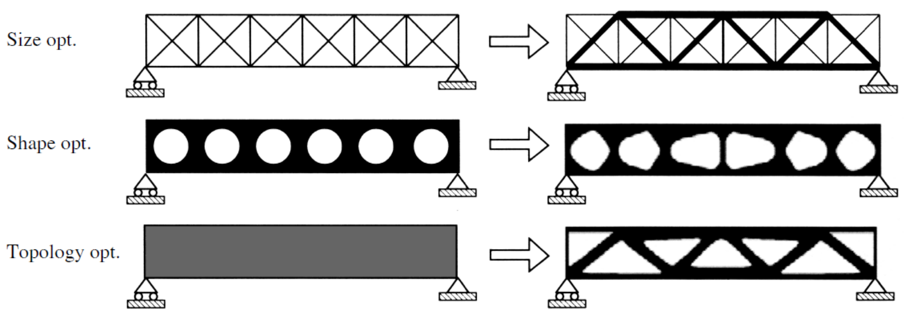


Figure 1.1: The three branches of structural optimization [1].

## 1.1 Literature Review

According to [2], the first two decades of numerical structural optimization were almost exclusively focused on sizing optimization. Real-life applications of sizing optimization includes the optimization of aircraft wings [3], truss structures [4], stiffeners in bending plates [5] and several others. Sizing optimization problems are particularly simple since, usually, no change to the geometric model is necessary. Despite its simplicity, sizing optimization is often regarded as the most limited approach, which is commonly associated with the fact that it only allows for subtle changes in the structure design.

On the other hand, the topology optimization is often regarded as the most sophisticated branch. Nowadays the study of topology optimization is mainly focused on the optimization of continuum structures by the so-called density-based formulation [6]. This formulation relies on defining the density value of each element of the domain's mesh as the design variables for the optimization problem. This methodology can handle a wide set of constraints while still providing high-performance and low-cost designs. However, as noted in [7], density-based topology optimization solutions are often considered as conceptual, given that a post-processing stage is needed to achieve a manufacturable design.

From a theoretical point of view the shape optimization branch is regarded as a subclass of topology optimization. Nevertheless, since the practical implementations are based on different techniques, these two branches are often treated separately. In general a structural shape optimization is formulated as a mathematical programming problem in which the boundary of the domain contains the design variables, the objective function is to minimize a given performance measure (*e.g.* volume, cost), and it is subject to a set of mechanical constraints (*e.g.* limits on displacements, yielding criteria). Suitable formulations allow for a straightforward integration between computer aided design (CAD) and the optimization procedure. This feature has significant practical benefits in the sense that design geometric constraints are generally easily enforced and that the final structure may be readily manufactured. In addition, several works, e.g. [8, 9, 10, 11], propose the integration between the topology and shape optimization in order to achieve an automated process linking the structural optimization to the actual manufacturing. This observation indicates that shape optimization is an indispensable tool in a real-life structural optimization process. In order to introduce such subject, the main aspects concerning the formulation and implementation of shape optimization methods are detailed in the following.

## Boundary Parametrization

One of the key ingredients in structural shape optimization is how to represent and control the structure's boundary shape. The main approaches towards the boundary parametrization in shape optimization are:

– **Parameter free approach**: This approach employs the model's discretization nodes as the design variables to the optimization problem. Although this approach leads to a simple representation of the boundary, and a straightforward numerical implementation, it also faces several drawbacks. Firstly, in order to obtain a manufacturable design, a post-processing stage is necessary to communicate the discrete solution with the CAD model, therefore deteriorating one of the main advantages of shape over topology optimization. Also, this approach often leads to a redundant number of design variables, which tends to compromise the efficiency of the numerical optimization methods. Lastly, the excessive number of design variable may also contribute to the undesired effect of producing jagged geometries [2].

– **Level-set**: The level-set method accounts for an implicit representation in which the boundary of the structure is defined as the iso-contour of a given scalar function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$. Usually, the level-set function is discretized in the domain nodes of a regular grid which in turn are defined as the design variables of the problem. The main advantage of such approach is its ability to represent smooth geometries with no changes to the initial grid. On the other hand, just like the parameter free approach, one of the main disadvantages of this approach is that a post-processing stage is also needed to communicate with a CAD model for manufacturing purposes. It is worth noting that this approach is particularly suitable for topology optimization problems, since the level-set is able to handle changes in the topology in a straightforward manner.

- **Polynomial representation**: The polynomial representation is a relatively old concept, introduced in [12], yet it constitutes the foundation for modern approaches. In this approach the design variables are comprised of the coefficients of given polynomials which model the boundary of the structure. One of the main advantages of this approach is related to the reduced number of design variables needed in the optimization modeling. In addition, the smoothness of the boundary is readily imposed by choosing the proper degree for the modeling polynomials. The main stumbling block associated to this approach is due to the oscillatory behavior of high degree polynomials which are usually necessary to represent the boundary's geometry.
- **Spline representation**: This approach may be regarded as an improvement for the polynomial representation in the sense that high order polynomials are no longer needed. In order to overcome such difficulty, this approach employs splines curves for modeling the boundary. The construction of splines by piecewise polynomials allows for the interpolation of several points without resorting to high order polynomials. Another advantage of this method is that the interpolation points may be defined as the design variables, thus leading to an intuitive modeling of the optimization problem.
- **Free-form representation**: The free-form representation is a natural enhancement of the spline representation. This approach employs splines curves which are usually present in modern computer-aided desing (CAD) softwares, such as *Bézier*, *B-splines* and non uniform rational *B-splines* (NURBS) curves. The main difference from the traditional *splines* is that these curves are modeled by control points which lie outside the curve. This feature allows for a more natural and intuitive design of the geometry. The choice of the control points as the design variables contributes to a straightforward communication of the optimization procedure with the CAD technology. Therefore, this is the most suitable approach for practical problems in the sense that the solution of the optimization may be readily used for manufacturing purposes.

### Structural Analysis

Another important subject in structural shape optimization is the choice of the structural analysis method to be employed. This choice is strongly related to the total time consuming of the optimization process as well as in the accuracy of the numerical results. Also, the dynamic geometry feature of shape optimization methods brings up tricky obstacles to most of the available analysis tools. The most common structural analysis methods employed in the structural shape optimization literature are introduced in the following.

– **Finite Element Method (FEM)**: FEM is by far the most widely used method in computational solid mechanics nowadays. It has been proved to be a robust and efficient numerical technique for obtaining the solution of boundary value problems of partial differential equations. Since early works [13], until the present time [14], FEM has been one of the most common techniques employed in the shape optimization literature. However, this choice usually leads to serious drawbacks associated with the mesh generation step. To begin with, shape optimization problems are generally solved by an iterative procedure in which the geometry changes at each step, thus requiring the burdensome remeshing for each iteration. Furthermore, mesh generators are far from being completely automated procedures and may even fail in the case of 3D complex geometries. At last, mesh generators are often based in *ad hoc* procedures, thus lacking of a mathematical formulation. This aspect imposes a great limitation in the sensitivity analysis stage given that the domain nodes' sensitivities are not well defined and, therefore, are usually neglected in practice [15]. In order to overcome such difficulties, several different approaches, such as mesh smoothing methods [14], have been proposed. Although these methods can alleviate much of the computational burden associated with the remeshing, they also lack of a mathematical background and thus place limitations to the sensitivity analysis stage. For a thorough review on field grid movement and mesh sensitivity analysis the reader is referred to [16].

– **Implicit Boundary methods**: Implicit boundary methods were originally proposed with the goal of bypassing the mesh generation step needed in FEM. In this class of methods only a non-conforming mesh, usually a structured grid, and an implicit representation of the boundary are needed. Therefore, in the context of a shape optimization procedure, regardless the changes in the geometry from a step to another, the initial grid may be left unaltered. Despite of looking as a promising alternative, the unavoidable implicit representation of the boundary leads to the same limitation as the level-set methods, namely a post-processing stage is needed. An application of such methods is proposed in [17], where the extended finite element method (X-FEM) is employed with a level-set method for the implicit boundary representation.

- **Boundary Element Method (BEM)**:The main feature of BEM is that it only requires the discretization of the boundary rather than the domain. According to [18], this advantage is particularly important for designing as the process usually involves a series of modifications which are more difficult to carry out using finite elements. This simple observation shows that BEM is a well-suited analysis tool to be integrated within a shape optimization procedure. Furthermore, unlike topology optimization problems, shape optimization does not require domain discretization, which further backs up the compatibility between BEM and shape optimization. The literature of shape optimization with BEM is quite vast and, among many others, includes [19, 20, 21, 22, 23].

### Optimization formulation

Several formulations for optimization of structural systems are available in the literature. According to [24], these formulations may be classified into three broad categories, i.e. the *nested analysis and design*, the *simultaneous analysis and design* and the *displacement two-phase approach.* In particular, for shape optimization problems, yet another formulation has been proposed, namely the *normal movement* approach. A brief overview of these approaches is introduced below.

- **nested analysis and design (NAND)**: NAND is the most common approach employed in the structural optimization literature. In this approach, only the structural design variables are treated as the optimization variables. All other response quantities, such as displacements, stresses and tractions, are kept outside the optimization procedure and treated as implicit functions of the design variables. The application of the NAND approach for shape optimization problems are found in [13, 21, 22, 25, 26, 27, 28].
- **simultaneous analysis and design (SAND)**: In this approach the state variables are incorporated into the optimization problem as design variables, thus including the equilibrium equations as equality constraints. The main feature is that no explicit structural analysis is needed. Actually, depending on the numerical optimization algorithm, the equilibrium is not even enforced at each iteration, but rather only at the final step. Therefore, this approach is particular suitable to handle non-linear

problems in the sense that the cumbersome solution of non-linear systems are no longer necessary. On the other hand, one of the drawbacks of this approach concerns the increase in the number of design variables, which may compromise the efficiency of the optimization solver. In [19] the authors employ the SAND formulation for shape optimization of 2D linear elasticity problems. In such work, the equality constraints are defined as the boundary integral equations of BEM, thus incorporating both the tractions and displacements as design variables, and an interior-point algorithm [29] is employed in order to efficiently solve the large-scale optimization problem.

- **displacement two-phase approach**: The displacement two-phase approach, introduced in [30], was originally developed for structural optimization of composite structures. The main idea of the method is to split the problem in a two-level optimization problem. In the inner loop, the state variables are fixed while the structural design variables are treated as the optimization variables. Conversely, in the outer loop, the state variables are handled as the optimization variables and the structural design variables remain fixed. The displacement two-phase approach has been mainly employed for sizing optimization of truss structures and to the knowledge of this author no work of shape optimization with this approach is available in the literature.
- **normal movement approach**: This approach belongs to the family of optimally criterion methods. In opposition to mathematical programming methods, this approach is based on an *ad hoc* procedure in which neither the objective function nor the constraints derivatives are required. In order to bypass the sensitivity analysis stage, this approach relies on a fixed search direction which, usually, is taken as the normal direction to the points chosen as the design variables. Subsequently, the step size is calculated by a normalized function which measures the performance of the objective function in each of the design variables. The next iteration is responsible for updating the search direction and the process is reinitialized. Although this method is extremely cheap, it only requires one structural analysis solution for each design variable per each iteration, global convergence is not guaranteed. Furthermore, there are cases in which the control points lies on a sharp edge and the normal direction is not well defined, leading to even more informal tricks in order to define the search direction. Examples of applications are given in [31] and [32], where the normal movement approach is employed in the shape optimization for minimization of stress concentrations of 2D and 3D structures.

## 1.2 Objective

In this work a numerical implementation of shape optimization in two-dimensional linear elasticity problems is proposed. The main goal is to provide a robust and efficient methodology capable of handling shape optimization problems for minimization of stress concentration effects. Such design methodology increases the fatigue strength as well as avoids crack initiation and propagation, thus enhancing both the life-span and safety of structures. In opposition to the most traditional design methodology ( i.e. minimization of volume subject to stress constraints), which aims at minimizing the material construction cost, the minimization of stress concentrations provides savings related to maintenance an repairing expenses.

In the proposed implementation the structural analysis is performed by the Symmetric Galerkin Boundary Element Method (SGBEM), thus disposing of the mesh generation burden. In addition, the boundary stress evaluation is carried out by an accurate approach which is ideally suited for problems with stress concentrations. As for the boundary parametrization, the free-form approach is employed, thus allowing for a straightforward integration between the geometric and optimization modeling. Furthermore, the NAND approach is employed for formulating the optimization problem, thus requiring the structural and sensitivity analysis for each step. In order to reduce the computational effort of such analysis, a matrix partition approach which exploits the features of the linear system associated with the SGBEM is proposed. Finally, the solution for the optimization problem is carried out by a powerful numerical optimization method, the so-called Second Order Conic Programming (SOCP). Specifically, the solution for the non-linear optimization is sought by solving a sequence of SOCP subproblems.

## 1.3 Outline

The remainder of the present work is organized as follows:

Chapter 2 discusses the main aspects concerning boundary representation and parametrization in the context of shape optimization problems. A brief overview of *polynomial interpolation*, *Bézier* and *B-splines* curves is introduced. Also, the design variable representation used for modeling the optimization problem is specified. Finally, an efficient evaluation of the structure's volume is proposed.

Chapter 3 details the implementation of SGBEM. Initially, the boundary integral equations and the *limit to the boundary* concept are introduced.

Afterwards, the main steps towards the numerical implementation of the method, such as discretization, weak formulation and system assembly, are discussed. In addition, the analytical treatment of singular integrals boundary conditions corners are reviewed. Finally, an accurate stress post-processing technique is detailed.

Chapter 4 address the sensitivity analysis of SGBEM. A specialized sensitivity analysis procedure is proposed in order to preserve the symmetry of the SGBEM equations, thus alleviating some of the computation effort in obtaining the derivatives. Furthermore, the sensitivity of stresses is carried out by an adaptation of the aforementioned stress post-processing technique.

Chapter 5 investigates a new proposed methodology in which stress constrained structural optimization problems are solved by means of modern conic optimization methods. Initially, a brief introduction of SOCP is present. Afterwards, it is shown how the most usual stress yield criteria can be cast into second-order conic constraints. Finally, the proposed formulation for stress constrained structural optimization by solving a sequence of SOCP subproblems is presented.

Chapter 6 presents the general framework of the shape optimization for minimization of stress concentrations. The optimization problem is formulated by adapting the aforementioned sequential SOCP approach. Numerical results are presented to demonstrate the robustness and efficiency of the proposed formulation. Additionally, an innovative block matrix partition approach is proposed in order to reduce the computation effort involved in structural analysis step by the SGBEM.

Finally, concluding remarks as well as suggestions for future works are addressed in Chapter 7.

# 2
# Boundary Representation

One of the key ingredients in Structural Shape Optimization is how to represent and control the structure's boundary shape. A common approach in the early works on Shape Optimization, e.g. [33], was to define the boundary parametrization in terms of the analysis' discretization nodes. Specifically, the boundary discretization nodes were defined as the design variables of the optimization problem. Although this approach leads to a natural boundary representation, numerical experiments have shown that the accuracy of the structural analysis often deteriorates as the optimization progresses. Another drawback of this approach is related to the excessive number of design variables which are required to represent the boundary. In the most recent literature on the subject, two main approaches are commonly employed, namely the free-form representation and the level-set method.

The level-set method accounts for an implicit representation in which the boundary of the structure is defined as the iso-contour of a given implicit function $\phi : \mathbb{R}^2 \to \mathbb{R}$ . Changes in the shape of the boundary are made by letting the level-set function dynamically change in time as the movement is described by a vector-valued function, usually called "*velocity field*", obtained as the solution of a *Hamilton-Jacobi* differential equation. The application of the level-set method in Structural Optimization was introduced by [34] in the context of a topology optimization procedure. In [17], the level-set method is combined with the extended finite element method (X-FEM) in a shape optimization for minimization of stress concentration effects.

The free-form approach is based on a parametric representation in which the boundary is defined by a set of parametric curves connected to each other. Each parametric curve is defined as the linear combination of given shape functions with its control points. Based on this approach a natural way to control the shape of the boundary is to choose the control points as the design variables of the optimization problem. One of the main advantages of this technique is that it enables for a straightforward integration between the geometric modeling and optimization formulation.

In this work the boundary is parametrized by means of the free-form approach using modern geometric modeling techniques in computer-aided design (CAD), namely *Bezier* and *B-Splines* curves. In the following, a brief overview of the formulation of such curves is introduced.

## 2.1 Parametric Curves

A parametric curve, in 2D, may be defined as a function $f : [a, b] \to \mathbb{R}^2$ of the interval $[a, b]$ into $\mathbb{R}^2$ which has continuous derivatives up to order $m \geq 1$, where $m$ is defined as the degree of the curve. In general, parametric curves may be defined as the linear combination of given shape functions with its control points as

$$\mathbf{c}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \sum_{i=1}^{n} N_i(t) \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \sum_{i=1}^{n} N_i(t)\ \mathbf{p}_i \tag{2-1}$$

where $\mathbf{c}(t)$ is the parametric curve, $N_i(t)$ are the shape functions and $\mathbf{p}_i$ are the control points with components $x_i$ and $y_i$.

The simplest example of a parametric curve is the straight line segment connecting two points, which may be defined as

$$\mathbf{l}(t, \mathbf{p}_1, \mathbf{p}_2) = \frac{t_1 - t}{t_1 - t_0}\mathbf{p}_1 + \frac{t - t_0}{t_1 - t_0}\mathbf{p}_2\,, \quad t \in [t_0, t_1] \tag{2-2}$$

where $\mathbf{p}_1$ and $\mathbf{p}_2$ are the control points while $t_0$ and $t_1$ are the initial and end parameters, respectively.

### Polynomial Interpolation Curves

A nice interpretation of expression in Equation (2-2) is to understand it as the convex combination between the two points, namely

$$\mathbf{l}(t) = (1 - \lambda(t))\,\mathbf{p}_1 + \lambda(t)\,\mathbf{p}_2, \quad 0 \leq \lambda(t) \leq 1 \tag{2-3}$$

where

$$\lambda(t) = \frac{t - t_0}{t_1 - t_0} \tag{2-4}$$

This idea can be further generalized to obtain curves with more control points. A simple process consists in taking the convex combination of two straight line segments which in turn connects three consecutive points, *i.e.*

$$\mathbf{c}(t, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = \frac{t_2 - t}{t_2 - t_0}\mathbf{l}_1(t, \mathbf{p}_1, \mathbf{p}_2) + \frac{t - t_0}{t_2 - t_0}\mathbf{l}_2(t, \mathbf{p}_2, \mathbf{p}_3)\,, \qquad t \in [t_0, t_2] \tag{2-5}$$

where $t_0 < t_1 < t_2$ are the control parameters.

An important property of this process is that the generated curves passes through the control points, thus giving a nice geometric interpretation. Evaluation of Equation (2-5) for the control parameters leads to

$$\mathbf{c}\left(t_0\right)=\mathbf{p}_1 \quad \mathbf{c}\left(t_1\right)=\mathbf{p}_2 \quad \mathbf{c}\left(t_2\right)=\mathbf{p}_3 \tag{2-6}$$

which shows that the curve actually interpolates the control points at the given control parameters.

The generalization of this process to an arbitrary number of points is given as

$$\mathbf{c}\left(t\right)=\lambda_0\left(t\right)\mathbf{p}_0+\lambda_1\left(t\right)\mathbf{p}_1+\cdots+\lambda_n\left(t\right)\mathbf{p}_n=\sum_{i=1}^{n}\lambda_i\left(t\right)\mathbf{p}_i \tag{2-7}$$

where $\lambda_i\left(t\right)$ are the Lagrange polynomials of degree $n$, *i.e.*

$$\lambda_i\left(t\right)=\prod_{\substack{0\leq j\leq n\\ j\neq i}}\frac{t-t_j}{t_i-t_j} \tag{2-8}$$

which are known to possess the following Kronecker Delta property

$$\lambda_i\left(t_k\right)=\begin{cases}1, & \text{if}\, k=i\\ 0, & \text{otherwise}\end{cases} \tag{2-9}$$

One of the main drawbacks of this class of curves is the oscillatory behavior associated with the Lagrange polynomials of high degree, also known as the Runge's phenomenon. This property leads to modeling difficulties given that a great number of points are usually necessary to define a curve. This oscillatory behavior is due to the fact that Equation (2-7) is not a true convex combination of the control points. Actually, by observing the quadratic case of equation (2-5), it can be seen that when $t$ is in $[t_0, t_1]$ the combination is not convex because $\mathbf{l}_2\left(t,\mathbf{p}_2,\mathbf{p}_3\right)$ is only convex within $[t_1, t_2]$. Accordingly, when $t$ is in $[t_1, t_2]$ then the combination is also not convex because $\mathbf{l}_1\left(t,\mathbf{p}_2,\mathbf{p}_3\right)$ is only convex within $[t_0, t_1]$.

## Bézier curve

Another interesting curve construction results in the so-called *Bézier* curves. This process follows the same previous idea except that, in order to obtain true convex combinations, all parameters are defined over the same interval, *i.e.* $[0, 1]$. For example, the linear case reduces to

$$\mathbf{l}\left(t,\mathbf{p}_1,\mathbf{p}_2\right)=\left(1-t\right)\mathbf{p}_1+t\mathbf{p}_2\,,\ \ t\in[0,1] \tag{2-10}$$

while the quadratic case is given as

$$\mathbf{c}\left(t,\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3\right) = (1-t)\,\mathbf{l}_1\left(t,\mathbf{p}_1,\mathbf{p}_2\right) + t\,\mathbf{l}_2\left(t,\mathbf{p}_2,\mathbf{p}_3\right), \qquad t \in [0,1] \tag{2-11}$$

This simple modification ensures that the curve lies within the convex hull of its control points, thus eliminating the oscillatory behavior of the polynomial interpolation curves. Unfortunately, this process also disposes of the Kronecker Delta property, *i.e.* these curves no longer pass through their control points. Nevertheless, this feature does not interfere in the design of *Bézier* curves, but rather makes the design of such curves even more intuitive.

The generalization of this construction for an arbitrary number of points is given as

$$\mathbf{c}\left(t\right) = b_0\left(t\right)\mathbf{p}_0 + b_1\left(t\right)\mathbf{p}_1 + \cdots + b_n\left(t\right)\mathbf{p}_n = \sum_{i=1}^{n} b_i\left(t\right)\mathbf{p}_i \tag{2-12}$$

where

$$b_i\left(t\right) = \left(\begin{array}{c} d \\ i \end{array}\right) t^i (1-t)^{d-i} \tag{2-13}$$

which are known as the Bernstein's polynomials [35].

Despite of avoiding the oscillatory behavior of the polynomial curves, the *Bézier* curves also suffers from the drawback of requiring high degree polynomials in order to define curves with many control points. This deficiency leads to a cumbersome modeling of such curves in the sense that moving one of the control points propagates changes to the entire curve. In addition, the processing time involved in the evaluation of the curve increases according to the respective degree, thus influencing the computational efficiency. A simple way to avoid this problem is to work with the so-called composite *Bézier* curves [36].

These curves can be obtained by joining several low-order *Bézier* curves, thus forming a piecewise polynomial. However, in order to enforce smoothness of the curve, the control points' location of adjacent *Bézier* curves must be chosen appropriately, thus placing some limitations to the modeling. Nevertheless, by slightly changing the aforementioned curve construction, it is possible to obtain piecewise polynomials which automatically tie together smoothly, the so-called *splines*.

### B-splines curves

The simplest example of a *spline* is the polyline which, in other words, is just a piecewise linear curve. The definition of a polyline may be given as

$$\mathbf{s}\left(t\right)=\begin{cases}\mathbf{l}_1\left(t,\mathbf{p}_1,\mathbf{p}_2\right) & t\in\left[t_1,t_2\right)\\ \mathbf{l}_2\left(t,\mathbf{p}_2,\mathbf{p}_3\right) & t\in\left[t_2,t_3\right)\\ \vdots & \vdots\\ \mathbf{l}_n\left(t,\mathbf{p}_n,\mathbf{p}_{n+1}\right) & t\in\left[t_n,t_{n+1}\right]\end{cases} \tag{2-14}$$

where $\left(\mathbf{p}_i\right)_{i=1}^{n}$ are the control points, $\mathbf{l}_i\left(t,\mathbf{p}_i,\mathbf{p}_{i+1}\right)$ are straight line segments connecting points $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$, and $\left(\mathbf{t}\right)_{i=1}^{n+1}$ are referred to as the knots of the curve.

Before presenting the general constructions of such splines, the following definitions are introduced.

**Definition 1** *Let $\mathbf{s}_d\left(t\right)$ be a spline curve of degree $d$. Then the sequence of non-decreasing real numbers defined by*

$$\mathbf{t}=\left\{t_1,t_2,\cdots,t_{n+d+1}\right\} \tag{2-15}$$

*is called the knot vector of $\mathbf{s}_d\left(t\right)$. Furthermore, if the first and last knots are repeated $d+1$ times, i.e.*

$$\begin{aligned}&t_1=t_2=\cdots=t_{d+1}\\&t_{n+1}=t_{n+2}=\cdots=t_{n+d+1}\end{aligned} \tag{2-16}$$

*then it is called an open knot vector.*

The general construction of higher degree splines is achieved by combining the weighted average scheme of the polynomial interpolation with the convex combination of the *Bézier* curves. To introduce such scheme the quadratic case is firstly investigated.

Let $\left(\mathbf{p}_i\right)_{i=1}^{4}$ be the four control points of a quadratic spline and $\left(\mathbf{t}_i\right)_{i=1}^{7}$ be the associated knot vector. The first step is to define the following linear segments as

$$\begin{aligned}\mathbf{l}_1\left(t,\mathbf{p}_1,\mathbf{p}_2\right)&=\frac{t_4-t}{t_4-t_2}\mathbf{p}_1+\frac{t-t_2}{t_4-t_2}\mathbf{p}_2\,,\quad t\in\left[t_2,t_4\right]\\ \mathbf{l}_2\left(t,\mathbf{p}_2,\mathbf{p}_3\right)&=\frac{t_5-t}{t_5-t_3}\mathbf{p}_2+\frac{t-t_3}{t_5-t_3}\mathbf{p}_3\,,\quad t\in\left[t_3,t_5\right]\\ \mathbf{l}_3\left(t,\mathbf{p}_3,\mathbf{p}_4\right)&=\frac{t_6-t}{t_6-t_4}\mathbf{p}_3+\frac{t-t_4}{t_6-t_4}\mathbf{p}_4\,,\quad t\in\left[t_4,t_6\right]\end{aligned} \tag{2-17}$$

The reason for choosing such different knot intervals in defining the linear segments is to ensure that the quadratic segments remain as true convex combinations. For example, if $t\in\left[t_3,t_4\right]$ then it also belongs both to $\left[t_2,t_4\right]$ and $\left[t_3,t_5\right]$. Therefore, the following quadratic segment

$$\mathbf{q}_3\left(t,\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3\right)=\frac{t_4-t}{t_4-t_3}\mathbf{l}_1\left(t,\mathbf{p}_1,\mathbf{p}_2\right)+\frac{t-t_3}{t_4-t_3}\mathbf{l}_2\left(t,\mathbf{p}_2,\mathbf{p}_3\right) \tag{2-18}$$

is a true convex combination of points $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. Likewise, the following quadratic segment

$$\mathbf{q}_4\left(t, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\right) = \frac{t_5 - t}{t_5 - t_4}\mathbf{l}_2\left(t, \mathbf{p}_2, \mathbf{p}_3\right) + \frac{t - t_4}{t_5 - t_4}\mathbf{l}_3\left(t, \mathbf{p}_3, \mathbf{p}_4\right) \tag{2-19}$$

is a true convex combination of points $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$.

With these expressions in hand, the quadratic *spline* may be defined as following piecewise polynomial

$$\mathbf{s}\left(t\right) = \begin{cases} \mathbf{q}_3\left(t, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\right) & t \in \left[t_3, t_4\right) \\ \mathbf{q}_4\left(t, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\right) & t \in \left[t_4, t_5\right] \end{cases} \tag{2-20}$$

According to this construction, a *spline* of degree $d$ with $n$ control points is given by

$$\mathbf{s}\left(t\right) = \begin{cases} \mathbf{f}_{d+1}\left(t, \mathbf{p}_1, \cdots, \mathbf{p}_{1+d}\right) & t \in \left[t_{d+1}, t_{d+2}\right) \\ \mathbf{f}_{d+2}\left(t, \mathbf{p}_2, \cdots, \mathbf{p}_{2+d}\right) & t \in \left[t_{d+2}, t_{d+3}\right) \\ \vdots & \vdots \\ \mathbf{f}_n\left(t, \mathbf{p}_{n-d}, \cdots, \mathbf{p}_n\right) & t \in \left[t_n, t_{n+1}\right] \end{cases} \tag{2-21}$$

where the polynomials $\mathbf{f}_i\left(t, \mathbf{p}_i, \cdots, \mathbf{p}_{i+d}\right)$ are given by the following recurrence relation

$$f_{i,d-r+1}\left(t\right) = \frac{t_{i+r} - t}{t_{i+r} - t_i} f_{i-1,d-r}\left(t\right) + \frac{t - t_i}{t_{i+r} - t_i} f_{i,d-r}\left(t\right) \tag{2-22}$$

for $i = d-r+1, \ldots, n$ and $r = d, d-1, \ldots, 1$, while $f_{i,0}\left(t\right) = \mathbf{p}_i$ for $i = 1, \ldots, n$.

Alternatively, the *spline* may also be written in the following appropriate form

$$\mathbf{s}\left(t\right) = \sum_{i=1}^{n} B_{i,d}\left(t\right) \mathbf{p}_i \tag{2-23}$$

where the so-called B-splines blending functions $B_{i,d}\left(t\right)$ are given as

$$B_{i,d}\left(t\right) = \frac{t - t_i}{t_{i+d} - t_i} B_{i,d-1}\left(t\right) + \frac{t_{i+1+d} - t}{t_{i+1+d} - t_{i+1}} B_{i+1,d-1}\left(t\right) \tag{2-24}$$

and

$$B_{i,0}\left(t\right) = \begin{cases} 1, & \text{if}\, t \in \left[t_i, t_{i+1}\right) \\ 0, & \text{otherwise} \end{cases} \tag{2-25}$$

The smoothness of the *B-spline* curves results from the following theorem

**Theorem 1** *Suppose that the number $t_{i+1}$ occurs $m$ times among the knots $\left(t_j\right)_{j=1-i1}^{m+d}$ with $m$ some integer bounded by $1 \leq m \leq d+1$, i.e.*

$$t_i < t_{i+1} = \ldots = t_{i+m} < t_{i+m+1} \tag{2-26}$$

*then the spline* $\mathbf{s}(t)$*, as defined in eq.* (2-21)*, has continuous derivatives up to order* $d - m$ *at the join* $t_i + 1$*.*

A detailed demonstration of this theorem is presented in [35].

This feature allows for a natural control over the smoothness of the B-spline by appropriately choosing the multiplicity of the knots. Also, if all knots are distinct, a global smoothness may be ensured by correctly choosing the degree of the B-spline. Specifically, a cubic *B-spline* curve, with no repeated knots, has continuous derivative up to order 2 everywhere along its parametric space. Furthermore, one can also enforce the B-spline to interpolate a given point by appropriately setting the multiplicity of a given knot to $m = d + 1$, where $d$ is the degree of the curve. Therefore, if an open knot vector is specified, the B-spline interpolates its initial and end point, which is a common practice in real-life modeling software.

## 2.2 Design variables representation

As aforementioned, in the context of the free-form representation, a natural way to control the shape of the boundary is to define the control points as the design variables of the optimization problem. The most common approach (in 2D problems), is to define the $x$ and/or $y$ coordinates of the control points as the design variables of the problem [25, 37, 38]. However, this approach often leads to a redundant number of design variables and also to difficulties in handling the design geometric constraints, inherent to shape optimization problems. An alternative representation defines the design variables as scalar values which control the movement of the control points along straight lines, the so-called *spans*.

Given the end points of a span, $\mathbf{s}_i^{\min} = [x_i^{\min}, y_i^{\min}]$ and $\mathbf{s}_i^{max} = [x_i^{\max}, y_i^{\max}]$, the movement of the associated control point is given as

$$\mathbf{p}_i = \mathbf{s}_i^{\min} + \mathbf{d}_i \alpha_i \tag{2-27}$$

where $\mathbf{d}_i$ is a direction vector defined as $\mathbf{d}_i = \mathbf{s}_i^{max} - \mathbf{s}_i^{\min}$ and $\alpha_i$ is a design variable associated with the span.

This approach can account for the case that both $x$ and $y$ coordinates are chosen as design variables. This case may be handled by simply defining two spans for a given control point, one in the $x$ direction and the other in the $y$ direction. Figure 2.1 depicts examples of this approach both for the regular case, where the control point is limited to move along a single direction, and for the particular case, where the control point is free to move in both directions.

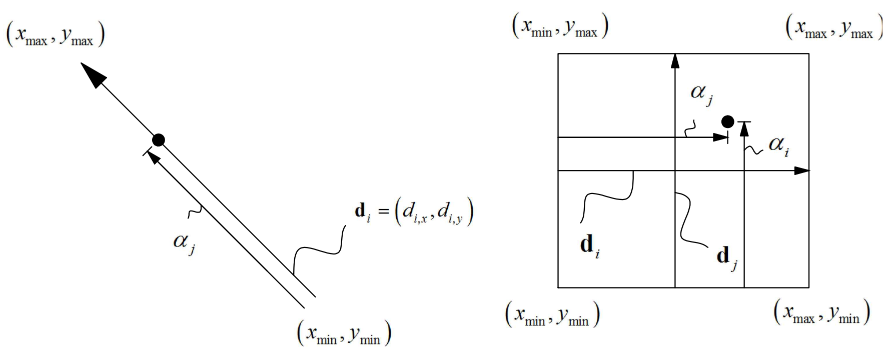


Figure 2.1: Design variables representation.

The sensitivity of the control points may be readily obtained by a simple differentiation of (2-27), *i.e.*

$$\dot{\mathbf{p}}_i := \frac{\partial \mathbf{p}_i}{\partial \alpha_i} = \mathbf{d}_i \tag{2-28}$$

where the superimposed dot represents the derivative with respect to the design variable.

For the sake of the structure volume evaluation, which will be shown later in this section, the movement of the control points may also be expressed in terms of their coordinates as

$$\begin{aligned} x_i &= x_i^{\min} + d_{i,x}\alpha_i \\ y_i &= y_i^{\min} + d_{i,y}\alpha_i \end{aligned} \tag{2-29}$$

where $d_{i,x}$ and $d_{i,y}$ are, respectively, the $x$ and $y$ components of the direction vector.

Considering all control points simultaneously, the following matrix expressions may be used

$$\begin{aligned} \mathbf{x} &= \mathbf{x}^{\min} + \mathbf{L}_x\alpha \\ \mathbf{y} &= \mathbf{y}^{\min} + \mathbf{L}_y\alpha \end{aligned} \tag{2-30}$$

where the $\mathbf{L}_x$ and $\mathbf{L}_y$ matrices are defined as

$$\begin{aligned} \mathbf{L}_x &= diag\,(d_{1,x}, d_{2,x}, \ \dots \ , d_{n,x}) \\ \mathbf{L}_y &= diag\,(d_{1,y}, d_{2,y}, \ \dots \ , d_{n,y}) \end{aligned} . \tag{2-31}$$

## 2.3 Volume evaluation

The structure's weight is a commonly employed performance measure in shape optimization and it is of great interest in both real-life and academic problems. In practical problems, the weight is directly related to the cost of the structure while in academic problems the weight is often enforced as a constraint in order to obtain a well-posed problem. Generally, in shape optimization problems, the material density, or thickness, is constant along the structure and therefore the weight may be obtained from the volume. The free-from representation allows for a very accurate and efficient technique, based on Green's Theorem, to calculate the structure's volume. Based on this technique, and making use of the design variable representation of Equation (2-30), it is shown that the volume may be obtained as a quadratic function of the design variables.

Let $\Gamma$ be a positively oriented, piece-wise smooth, simple closed curve in the plane, $\Omega$ be the region bounded by $\Gamma$ and $P$ and $Q$ be two arbitrary continuous functions. The Green's Theorem states that if $P$ and $Q$ have continuous partial derivatives on an open region that contains $\Omega$, then

$$\oint_{\Gamma} Pdx + \oint_{\Gamma} Qdy = \iint_{\Omega} \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) d\Omega \tag{2-32}$$

Choosing $P$ and $Q$ as

$$\begin{aligned} P(x,y) &= 0 \\ Q(x,y) &= x \end{aligned} \tag{2-33}$$

then the Green's Theorem results in

$$\oint_{\Gamma} xdy = \iint_{\Omega} d\Omega \tag{2-34}$$

where the expression on the right-hand side of Equation (2-34) is the domain's volume (for an unitary thickness).

Since the boundary is defined as a set of connected smooth curves, the line integral on the left-hand side of Equation (2-34) may be written as

$$\Gamma = \bigcup_{i=1}^{m} \Gamma_i \Rightarrow \oint_{\Gamma} xdy = \sum_{i=1}^{m} \oint_{\Gamma_i} xdy \tag{2-35}$$

Each curve $\Gamma_i$ has a parametric representation, as given in Equation (2-1), thus each of the line integrals on the right-hand side of Equation (2-35) may be rewritten in the following form

$$\oint_{\Gamma_i} xdy = \int_{t_0}^{t_1} \sum_{i=1}^{n} N_i(t)\, x_i \sum_{i=1}^{n} \frac{dN_i(t)}{dt} y_i dt \tag{2-36}$$

Taking the constant values outside the integration leads to

$$\oint_{\mathcal{C}_i} xdy = \sum_{i=1}^{n} x_i \left[ \int \sum_{i=1}^{n} N_i(t) \sum_{i=1}^{n} \frac{dN_i(t)}{dt} dt \right] y_i \tag{2-37}$$

which can also be written in following matrix form

$$\oint_{\mathcal{C}_i} xdy = (\mathbf{x}')^T \mathbf{V}' (\mathbf{y}') \tag{2-38}$$

where $\mathbf{x}'$ and $\mathbf{y}'$ are, respectively, vectors containing the $x$ and $y$ coordinates of curve $\Gamma_i$ and $\mathbf{V}$ is the matrix defined as

$$V_{ij} := \left[ \int N_i (t) \frac{dN_j (t)}{dt} dt \right] \tag{2-39}$$

By assigning nodal identities to each of the control points it is then possible to assembly these results in a global expression as

$$\sum_{i=1}^{m} \oint_{\Gamma_i} xdy = \mathbf{x}^T \mathbf{V} \mathbf{y} \tag{2-40}$$

where $\mathbf{x}$ and $\mathbf{y}$ are, respectively, vectors containing the global $x$ and $y$ coordinates of the boundary and $\mathbf{V}$ is obtained by a global assemble of the local matrices $\mathbf{V}'$.

Splitting the boundary as

$$\Gamma = \Gamma_f \cup \Gamma_o \tag{2-41}$$

where $\Gamma_f$ is the portion of the boundary which will remain fixed and $\Gamma_o$ the portion which will move during the optimization process, the volume of the structure may be obtained as

$$\oint_{\Gamma_f} xdy + \oint_{\Gamma_o} xdy = \mathbf{x}_f{}^T \mathbf{V}_f \mathbf{y}_f + \mathbf{x}_o{}^T \mathbf{V}_o \mathbf{y}_o \tag{2-42}$$

The first term in the right-hand side of Equation (2-42) may be evaluated in a straightforward manner, thus resulting in a constant scalar value. The second term can be further expanded by substituting the expressions from Equation (2-30), *i.e.*

$$\begin{aligned} \mathbf{x}_o{}^T \mathbf{V}_o \mathbf{y}_o &= \left(\mathbf{x}^{\min} + \mathbf{L}_x \alpha\right)^T \mathbf{V}_o \left(\mathbf{y}^{\min} + \mathbf{L}_y \alpha\right) \\ &= \alpha^T \mathbf{Q} \alpha + \alpha^T \mathbf{b} + \left(\mathbf{x}^{\min}\right)^T \mathbf{V}_o \left(\mathbf{y}^{\min}\right) \end{aligned} \tag{2-43}$$

where

$$\begin{aligned} \mathbf{Q} &= (\mathbf{L}_x)^T \mathbf{V}_o (\mathbf{L}_y) \\ \mathbf{b} &= \left[ (\mathbf{L}_x)^T \mathbf{V}_o \left(\mathbf{y}^{\min}\right) + (\mathbf{L}_y)^T \mathbf{V}_o \left(\mathbf{x}^{\min}\right) \right] \end{aligned} \tag{2-44}$$

Therefore, the volume of the structure can be define as the following quadratic function

$$V(\alpha) = \alpha^T \mathbf{Q} \alpha + \alpha^T \mathbf{b} + c \tag{2-45}$$

where

$$c = \left(\mathbf{x}^{\min}\right)^T \mathbf{V}_o \left(\mathbf{y}^{\min}\right) + \mathbf{x}_f{}^T \mathbf{V}_f \mathbf{y}_f \tag{2-46}$$

Differentiation of (2-45) yields the sensitivity of the volume as

$$\nabla V\left(\alpha\right)=2\mathbf{Q}\alpha+\mathbf{b} \tag{2-47}$$

It is worth noting that the integrals over the parametric space, defined in Equation (2-37), do not change during the optimization procedure, thus may be evaluated only once throughout the optimization process, preferably in a preprocessing stage.

# 3
# Symmetric Galerkin Boundary Element Method

The Boundary Element Method (BEM) is a numerical approach for solving partial differential equations in which the governing equations are stated as integrals identities which only take values on the boundary of the problem. Thus, one of the main advantages of such method relies on the reduction of the dimensionality order of the problem by one, *i.e.* the governing equations may be stated as line integrals equations for 2D problems and surface integrals equations for 3D problems. According to [39], another advantage of BEM is the high resolution of stresses, which makes this method well-suited for modeling problems of rapidly changing stresses, such as stress concentration problems.

Traditional BEM formulations rely on the use of the so called Collocation Method (CM). In this approach the discretized BEM integral equations are enforced to satisfy the governing equations in a strong form in which, generally, the discretization nodes are chosen as the collocation points. Applying the integrals equations on each of the collocation points leads to a system of linear equations which may be solved in order to recover the unknown boundary values. One of the main disadvantages of this approach is that it leads to a fully populated and non-symmetric system of linear equations, contributing to a high computational cost for its solution.

The Symmetric Galerkin Boundary Element Method (SGBEM) is an alternative approach in which the BEM integral equations are sought to satisfy the governing equations in a weak form. In this approach the weak form is obtained by enforcing the BEM integral equations in a Galerkin weighted-residual sense. In order to obtain a symmetric system of equations, both the single and double layer fundamental solutions are employed. Therefore, the main stumbling block regarding the SGBEM approach is due to the treatment of hypersingular integrals arising from the double layer fundamental solutions. In order to overcome this difficulty several approaches have been proposed in the literature such as the direct approach [40], indirect regularization techniques [41], analytical solutions [42] and the *limit to the boundary* approach [43]. Another great advantage of the SGBEM approach concerns the treatment of corners which may be handled in a more elegant and simpler way than

CM. Finally, the treatment of hypersingular integrals, in particular employing the *limit to the boundary* approach, allows for a very accurate and efficient technique [44] in order to obtain first order boundary derivatives, such as boundary stresses in elasticity problems.

## 3.1 Boundary Integral Equations

Consider a domain $\Omega$ in $\mathbb{R}^2$, as shown in Figure 3.1, with boundary $\Gamma$ subjected to body forces $b_i$, prescribed displacements $\bar{u}_i$ in $\Gamma_u$ and tractions $\bar{t}_i$ in $\Gamma_t$ (with $i = 1, 2$ or $i = x, y$). The equilibrium equations of elastostatics are given by

$$\sigma_{ij,j} = -b_i \;\; in \;\; \Omega \tag{3-1}$$

subjected to

$$\begin{aligned} u_i = \bar{u}_i \quad & on \quad \Gamma_u \\ t_i = \bar{t}_i \quad & on \quad \Gamma_t \end{aligned} \tag{3-2}$$

where $\sigma_{ij}$ are the stresses components.

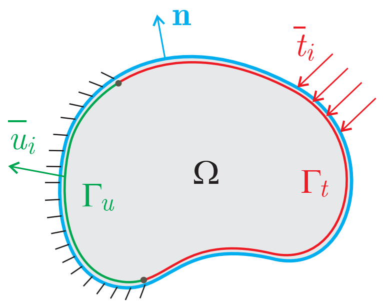


Figure 3.1: Elastostatic problem setting.

In the absence of body forces, *i.e.* $b_i = 0$, the *Somigliana Identity* restates the equilibrium Equations (3-1) by means of a boundary integral equation (BIE) which relates a displacement on a source point $P = (x_P, y_P) \notin \Gamma$ due to displacements and tractions on field points $Q = (x_Q, y_Q) \in \Gamma$, *i.e.*

$$ku_i(P) = \int_\Gamma G_{ij}^{uu}(P,Q) t_i(Q) d\Gamma - \int_\Gamma G_{ij}^{ut}(P,Q) u_i(Q) d\Gamma \tag{3-3}$$

where

$$k = \begin{cases} 1 \;\; \text{if} \;\; P \in \Omega \\ 0 \;\; \text{if} \;\; P \notin \Omega \cup \Gamma \end{cases} \tag{3-4}$$

and the displacement and traction kernel tensor functions $G_{ij}^{uu}$ and $G_{ij}^{ut}$ arises from the Kelvin fundamental solutions which are employed in the formulation of Equation (3-3). For a plane strain state the fundamental solutions are given as

$$G_{ij}^{uu}(P,Q) = \frac{1}{8\pi\mu(1-\nu)}\left[(3-4\nu)\ln\left(\frac{1}{r}\right)\delta_{ij} + r_{,i}r_{,j}\right] \tag{3-5}$$

$$G_{ij}^{ut}(P,Q) = \tfrac{-1}{4\pi(1-\nu)r}\left[\tfrac{\partial r}{\partial n}\left[(1-2\nu)\,\delta_{ij} + 2r_{,i}r_{,j}\right] + (1-2\nu)\left(n_i r_{,j} - n_k r_{,j}\right)\right] \tag{3-6}$$

where $\nu$ and $\mu$ are, respectively, the Poisson coefficient and the shear modulus, $n_j$ are the normal components on the point $Q$ and $r$ is the distance between source point $P$ and field point $Q$.

For a plane stress state the fundamental solutions may be readily obtained by a simply modification of the material properties as follows

$$E^{pstress} = \tfrac{E(1+2\nu)}{(1+\nu)^2};\quad \nu^{pstress} = \tfrac{\nu}{1+\nu}\ ; \tag{3-7}$$

Using the Hooke's Law, it is straightforward to obtain another boundary integral equation which relates the stresses on a source point $P$ due to the displacements and tractions on the boundary $\Gamma$ as

$$k\sigma_i(P) = \int_\Gamma D(P,Q)t_i(Q)d\Gamma - \int_\Gamma S(P,Q)u_i(Q)d\Gamma. \tag{3-8}$$

Rewriting this equation in terms of tractions $t_i$, according to $t_i = \sigma_{ij}m_j$, where $m_j$ are the normal components the source point $P$, gives

$$kt_i(P) = \int_\Gamma G_{ij}^{tu}(P,Q)t_i(Q)d\Gamma - \int_\Gamma G_{ij}^{tt}(P,Q)u_i(Q)d\Gamma \tag{3-9}$$

where

$$G_{ij}^{tu}(P,Q) = \tfrac{1}{4\pi(1-\nu)r}\left[\tfrac{\partial r}{\partial m}\left[(1-2\nu)\,\delta_{ij} + 2r_{,i}r_{,j}\right] - (1-2\nu)\left(m_i r_{,j} - m_k r_{,j}\right)\right] \tag{3-10}$$

and

$$\begin{aligned} G_{ij}^{tt}(P,Q) = \tfrac{\mu}{2\pi(1-\nu)r^2}\quad & (\delta_{ij}\delta_{kw} + \delta_{ki}\delta_{jw} - \delta_{jk}\delta_{iw} \\ & +2\delta_{jk}r_{,i}r_{,w} + 2\delta_{iw}r_{,j}r_{,k} - 8r_{,i}r_{,w}r_{,j}r_{,k})\, m_{,k}n_{,w} \end{aligned}. \tag{3-11}$$

It is important to note that, so far, Equations (3-3) and (3-9) are only valid for points which do not lie on the boundary. This is due to the fact that the kernels, defined in Equations (3-5), (3-6) and (3-11), become singular when $P$ approaches $Q$, thus resulting in boundary integrals which are not immediately well defined. Usually, in the literature [39, 18], a limit process is employed in order to move the source point to the boundary of the problem. In this process a small portion around the singularity point is excluded from the boundary, typically a sphere of radius $\varepsilon$. Then, after a suitable analytical treatment, the excluded region is reinserted back by taking the limit as $\varepsilon \to 0$. Alternatively, in this work, this problem is overcome by employing the *limit to the boundary* approach [45]. In this approach the source point is taken to the boundary by means of a limit definition, given as

$$u_i(P) = \lim_{P_I \to P} \left[ \int_\Gamma G^{uu}_{ij}(P_I, Q) t_i(Q) d\Gamma - \int_\Gamma G^{ut}_{ij}(P_I, Q) u_i(Q) d\Gamma \right] \tag{3-12}$$

$$t_i(P) = \lim_{P_I \to P} \left[ \int_\Gamma G^{tu}_{ij}(P_I, Q) t_i(Q) d\Gamma - \int_\Gamma G^{tt}_{ij}(P_I, Q) u_i(Q) d\Gamma \right] \tag{3-13}$$

for points $P_I$ approaching the boundary from inside the domain, and

$$\lim_{P_E \to P} \left[ \int_\Gamma G^{uu}_{ij}(P_E, Q) t_i(Q) d\Gamma - \int_\Gamma G^{ut}_{ij}(P_E, Q) u_i(Q) d\Gamma \right] = 0 \tag{3-14}$$

$$\lim_{P_E \to P} \left[ \int_\Gamma G^{tu}_{ij}(P_E, Q) t_i(Q) d\Gamma - \int_\Gamma G^{tt}_{ij}(P_E, Q) u_i(Q) d\Gamma \right] = 0 \tag{3-15}$$

for points $P_E$ approaching the boundary from outside the domain.

## 3.2 Numerical Implementation

The numerical solution for the differential Equation (3-1), subject to (3-2), by means of the BIEs (3-3) and (3-9) can be obtained by subdividing the boundary into a set of elements. In each boundary element the geometry, displacements and tractions are approximated through the interpolation of its nodal values

$$\begin{aligned} x(\xi) &= \sum_{i=1}^{ndof} N_i(\xi)\, x_i; \quad & u_x(\xi) &= \sum_{i=1}^{ndof} N_i(\xi)\, u_{xi} \quad & t_x(\xi) &= \sum_{i=1}^{ndof} N_i(\xi)\, t_{xi} \\ y(\xi) &= \sum_{i=1}^{ndof} N_i(\xi)\, y_i; \quad & u_y(\xi) &= \sum_{i=1}^{ndof} N_i(\xi)\, u_{y_i} \quad & t_y(\xi) &= \sum_{i=1}^{ndof} N_i(\xi)\, t_{y_i} \end{aligned} \tag{3-16}$$

where $N_i$ are the interpolation functions and $x_i$, $y_i$, $u_{xi}$, $u_{yi}$, $t_{xi}$ and $t_{yi}$ are nodal values of the degrees of freedom of a boundary element. These same relations may be conveniently rewritten in a matrix form as

$$\mathbf{u}(\xi) = \begin{bmatrix} u_x(\xi) \\ u_y(\xi) \end{bmatrix} = \mathbf{N}(\xi)\, \mathbf{u}_e; \quad \mathbf{t}(\xi) = \begin{bmatrix} t_x(\xi) \\ t_y(\xi) \end{bmatrix} = \mathbf{N}(\xi)\, \mathbf{t}_e; \tag{3-17}$$

where $\mathbf{u}_e$ and $\mathbf{t}_e$ are vectors containing the nodal values of displacements and tractions and $\mathbf{N}(\xi)$ is a matrix containing the interpolation functions.

Substituting the discretizations defined in Equation (3-17) into Equations (3-14) and (3-15) leads to

$$R^1(P) = \sum_{e=1}^{n_e} \left[ \begin{array}{l} \left( \lim\limits_{P_E \to P} \int\limits_0^1 \mathbf{G}^{uu}(P_E, Q(\xi))\,\mathbf{N}(\xi)\,J_e d\xi \right) \mathbf{t}_e \;\ldots \\ \qquad - \left( \lim\limits_{P_E \to P} \int\limits_0^1 \mathbf{G}^{ut}(P_E, Q(\xi))\,\mathbf{N}(\xi)\,J_e d\xi \right) \mathbf{u}_e \end{array} \right] \neq 0 \tag{3-18}$$

$$R^2(P) = \sum_{e=1}^{n_e} \left[ \begin{array}{l} \left( \lim\limits_{P_E \to P} \int\limits_0^1 \mathbf{G}^{tu}(P_E, Q(\xi))\,\mathbf{N}(\xi)\,J_e d\xi \right) \mathbf{t}_e \;\ldots \\ \qquad - \left( \lim\limits_{P_E \to P} \int\limits_0^1 \mathbf{G}^{tt}(P_E, Q(\xi))\,\mathbf{N}(\xi)\,J_e d\xi \right) \mathbf{u}_e \end{array} \right] \neq 0 \tag{3-19}$$

where $R^1(P)$ and $R^2(P)$ are the residual functions related to the interpolation error, $J_e$ is the Jacobian of the coordinate transformation and $n_e$ is the number of boundary elements.

Employing the Galerkin weighted-residual technique it is possible to obtain a system of linear equations which allows the determination of the boundary unknown values. The Galerkin method states

$$\int\limits_\Gamma w(P)\,R^1(P) d\Gamma = 0 \tag{3-20}$$

$$\int\limits_\Gamma w(P)\,R^2(P) d\Gamma = 0 \tag{3-21}$$

The weight function $w(P)$ is approximated by the same interpolation functions used in approximating the variable fields, this is

$$\mathbf{w} = \begin{bmatrix} w_x(\eta) \\ w_y(\eta) \end{bmatrix} = \mathbf{N}(\eta)\,\mathbf{w}_e \tag{3-22}$$

where $\mathbf{w}_e$ is a vector containing the values of the weight function in the degrees of freedom of a boundary element.

Rewriting Equations (3-20) and (3-21) in terms of the discretizations defined in Equation (3-17) gives

$$\mathbf{w}_{ep} \sum_{ep=1}^{M} \sum_{eQ=1}^{M} \left[ \begin{array}{l} \left( \lim\limits_{p_E \to p} \int\limits_0^1 \int\limits_0^1 \mathbf{N}^T(\eta)\,\mathbf{G}^{uu}(P_E(\eta), Q(\xi))\,\mathbf{N}(\xi)\,J_{ep} J_{eQ} d\xi d\eta \right) \mathbf{t}_{eQ} \;\ldots \\ \qquad - \left( \lim\limits_{p_E \to p} \int\limits_0^1 \int\limits_0^1 \mathbf{N}^T(\eta)\,\mathbf{G}^{ut}(P_E(\eta), Q(\xi))\,\mathbf{N}(\xi)\,J_{ep} J_{eQ} d\xi d\eta \right) \mathbf{u}_{eQ} \end{array} \right] = 0 \tag{3-23}$$

$$\mathbf{w}_{ep}\sum_{ep=1}^{M}\sum_{eQ=1}^{M}\left[\begin{array}{l}\left(\lim\limits_{p_E\to p}\int\limits_0^1\int\limits_0^1 \mathbf{N}^T(\eta)\,\mathbf{G}^{tu}(P_E(\eta),Q(\xi))\,\mathbf{N}(\xi)\,J_{ep}J_{eQ}d\xi d\eta\right)\mathbf{t}_{eQ}\ \ldots\\ \qquad -\left(\lim\limits_{p_E\to p}\int\limits_0^1\int\limits_0^1 \mathbf{N}^T(\eta)\,\mathbf{G}^{tt}(P_E(\eta),Q(\xi))\,\mathbf{N}(\xi)\,J_{ep}J_{eQ}d\xi d\eta\right)\mathbf{u}_{eQ}\end{array}\right]=0 \tag{3-24}$$

where $ep$ and $eQ$ are, respectively, a source element and a field element.

As the Galerkin method states that the weight function is non-zero everywhere, then the nodal values of the weight function, on the left hand side of Equation (3-23) and (3-24), may be canceled out, thus resulting in the following equations

$$\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\mathbf{I}^{uu}(\xi,\eta)\,d\xi d\eta\right)\mathbf{t}_{eQ}=\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\mathbf{I}^{ut}(\xi,\eta)\,d\xi d\eta\right)\mathbf{u}_{eQ} \tag{3-25}$$

$$\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\mathbf{I}^{tu}(\xi,\eta)\,d\xi d\eta\right)\mathbf{t}_{eQ}=\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\mathbf{I}^{tt}(\xi,\eta)\,d\xi d\eta\right)\mathbf{u}_{eQ} \tag{3-26}$$

where

$$\begin{aligned}\mathbf{I}^{uu}(\xi,\eta)&=\mathbf{N}^T(\eta)\,\mathbf{G}^{uu}(P_E(\eta),Q(\xi))\,\mathbf{N}(\xi)\,J_{ep}J_{eQ}\\ \mathbf{I}^{ut}(\xi,\eta)&=\mathbf{N}^T(\eta)\,\mathbf{G}^{ut}(P_E(\eta),Q(\xi))\,\mathbf{N}(\xi)\,J_{ep}J_{eQ}\\ \mathbf{I}^{tu}(\xi,\eta)&=\mathbf{N}^T(\eta)\,\mathbf{G}^{tu}(P_E(\eta),Q(\xi))\,\mathbf{N}(\xi)\,J_{ep}J_{eQ}\\ \mathbf{I}^{tt}(\xi,\eta)&=\mathbf{N}^T(\eta)\,\mathbf{G}^{tt}(P_E(\eta),Q(\xi))\,\mathbf{N}(\xi)\,J_{ep}J_{eQ}\end{aligned} \tag{3-27}$$

Rewriting Equations (3-25) and (3-26) in a matrix form gives

$$\begin{bmatrix}\mathbf{R}^{uu}_{\mathrm{uu}} & \mathbf{R}^{uu}_{\mathrm{ut}}\\ \mathbf{R}^{uu}_{\mathrm{tu}} & \mathbf{R}^{uu}_{\mathrm{tt}}\end{bmatrix}\begin{Bmatrix}\mathbf{t}_{\mathrm{u}}\\ \bar{\mathbf{t}}_{\mathrm{t}}\end{Bmatrix}=\begin{bmatrix}\mathbf{R}^{ut}_{\mathrm{uu}} & \mathbf{R}^{ut}_{\mathrm{ut}}\\ \mathbf{R}^{ut}_{\mathrm{tu}} & \mathbf{R}^{ut}_{\mathrm{tt}}\end{bmatrix}\begin{Bmatrix}\bar{\mathbf{u}}_{\mathrm{u}}\\ \mathbf{u}_{\mathrm{t}}\end{Bmatrix}\therefore\begin{cases}\mathbf{R}^{uu}_{\mathrm{uu}}\mathbf{t}_{\mathrm{u}}+\mathbf{R}^{uu}_{\mathrm{ut}}\bar{\mathbf{t}}_{\mathrm{t}}=\mathbf{R}^{ut}_{\mathrm{uu}}\bar{\mathbf{u}}_{\mathrm{u}}+\mathbf{R}^{ut}_{\mathrm{ut}}\mathbf{u}_{\mathrm{t}}\\ \mathbf{R}^{uu}_{\mathrm{tu}}\mathbf{t}_{\mathrm{u}}+\mathbf{R}^{uu}_{\mathrm{tt}}\bar{\mathbf{t}}_{\mathrm{t}}=\mathbf{R}^{ut}_{\mathrm{tu}}\bar{\mathbf{u}}_{\mathrm{u}}+\mathbf{R}^{ut}_{\mathrm{tt}}\mathbf{u}_{\mathrm{t}}\end{cases} \tag{3-28}$$

$$\begin{bmatrix}\mathbf{R}^{tu}_{\mathrm{uu}} & \mathbf{R}^{tu}_{\mathrm{ut}}\\ \mathbf{R}^{tu}_{\mathrm{tu}} & \mathbf{R}^{tu}_{\mathrm{tt}}\end{bmatrix}\begin{Bmatrix}\mathbf{t}_{\mathrm{u}}\\ \bar{\mathbf{t}}_{\mathrm{t}}\end{Bmatrix}=\begin{bmatrix}\mathbf{R}^{tt}_{\mathrm{uu}} & \mathbf{R}^{tt}_{\mathrm{ut}}\\ \mathbf{R}^{tt}_{\mathrm{tu}} & \mathbf{R}^{tt}_{\mathrm{tt}}\end{bmatrix}\begin{Bmatrix}\bar{\mathbf{u}}_{\mathrm{u}}\\ \mathbf{u}_{\mathrm{t}}\end{Bmatrix}\therefore\begin{cases}\mathbf{R}^{tu}_{\mathrm{uu}}\mathbf{t}_{\mathrm{u}}+\mathbf{R}^{tu}_{\mathrm{ut}}\bar{\mathbf{t}}_{\mathrm{t}}=\mathbf{R}^{tt}_{\mathrm{uu}}\bar{\mathbf{u}}_{\mathrm{u}}+\mathbf{R}^{tt}_{\mathrm{ut}}\mathbf{u}_{\mathrm{t}}\\ \mathbf{R}^{tu}_{\mathrm{tu}}\mathbf{t}_{\mathrm{u}}+\mathbf{R}^{tu}_{\mathrm{tt}}\bar{\mathbf{t}}_{\mathrm{t}}=\mathbf{R}^{tt}_{\mathrm{tu}}\bar{\mathbf{u}}_{\mathrm{u}}+\mathbf{R}^{tt}_{\mathrm{tt}}\mathbf{u}_{\mathrm{t}}\end{cases} \tag{3-29}$$

where $\mathbf{t}_{\mathrm{u}}$ and $\mathbf{u}_{\mathrm{t}}$ are unknown nodal values of tractions and displacements, and $\bar{\mathbf{t}}_{\mathrm{t}}$ and $\bar{\mathbf{u}}_{\mathrm{u}}$ are prescribed values.

The submatrices $\mathbf{R}$, defined in Equations (3-28) and (3-29), refers to the evaluation of Equations (3-25) and (3-26). The subscripts of each submatrix indicate the part of the boundary in which the integrals were evaluated. The subscript $u$ refers to the portion in which displacements are prescribed and the subscript $t$ refers to the portion in which the tractions are prescribed. The first subscript indicates the portion with respect to the source element and the second is related to the field element portion. The superscripts indicate which of the kernels are involved in the integral evaluation. Thus, for example,

$\mathbf{R}^{uu}_{\mathrm{uu}}$ is obtained by evaluating the integrals involving the kernel defined in Equation (3-5) where both the source and field elements belongs to the part of the boundary in which displacements are prescribed.

It is worth noting that the system of equations defined in Equation (3-28) is sufficient in determining the unknown nodal values. Rearranging the system in order to move all the unknown values to the left-hand side and all the known ones to the right-hand side gives

$$\begin{bmatrix} \mathbf{R}^{uu}_{\mathrm{uu}} & -\mathbf{R}^{ut}_{\mathrm{ut}} \\ \mathbf{R}^{uu}_{\mathrm{tu}} & -\mathbf{R}^{ut}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \mathbf{t}_{\mathrm{u}} \\ \mathbf{u}_{\mathrm{t}} \end{Bmatrix} = \begin{bmatrix} \mathbf{R}^{ut}_{\mathrm{uu}} & -\mathbf{R}^{uu}_{\mathrm{ut}} \\ \mathbf{R}^{ut}_{\mathrm{tu}} & -\mathbf{R}^{uu}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{u}}_{\mathrm{u}} \\ \bar{\mathbf{t}}_{\mathrm{t}} \end{Bmatrix} \tag{3-30}$$

Solving the above system in order to obtain the unknown nodal values is referred to as to solve the problem by the Galerkin Boundary Element Method (GBEM). In this case, the evaluation of integrals only involves terms with singularities of type $\log(r)$ (weak singularity) and $r^{-1}$ (strong singularity). The numerical treatment of such improper integrals can be carried out by means of suitable coordinate transformations together with regularization schemes [46]. However, the system defined in Equation (3-30) retains the same undesired property of the CM, *i.e.* it is non-symmetric.

Alternatively, the SGBEM employs both (3-28) and (3-29) system of equations, which together give the double as necessary equations to solve the problem. Therefore, SGBEM relies on a strategy to choose equations from both systems in order to obtain a symmetric linear system. This new system is obtained by choosing the first row block of equations from the system (3-28) and the second row block of system (3-29), resulting in

$$\begin{bmatrix} \mathbf{R}^{uu}_{\mathrm{uu}} & \mathbf{R}^{uu}_{\mathrm{ut}} \\ \mathbf{R}^{tu}_{\mathrm{tu}} & \mathbf{R}^{tu}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \mathbf{t}_{\mathrm{u}} \\ \bar{\mathbf{t}}_{\mathrm{t}} \end{Bmatrix} = \begin{bmatrix} \mathbf{R}^{ut}_{\mathrm{uu}} & \mathbf{R}^{ut}_{\mathrm{ut}} \\ \mathbf{R}^{tt}_{\mathrm{tu}} & \mathbf{R}^{tt}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{u}}_{\mathrm{u}} \\ \mathbf{u}_{\mathrm{t}} \end{Bmatrix} \tag{3-31}$$

Rearranging the system (3-31), one can obtain

$$\begin{bmatrix} \mathbf{R}^{uu}_{\mathrm{uu}} & -\mathbf{R}^{ut}_{\mathrm{ut}} \\ -\mathbf{R}^{tu}_{\mathrm{tu}} & \mathbf{R}^{tt}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \mathbf{t}_{\mathrm{u}} \\ \mathbf{u}_{\mathrm{t}} \end{Bmatrix} = \begin{bmatrix} \mathbf{R}^{ut}_{\mathrm{uu}} & -\mathbf{R}^{uu}_{\mathrm{ut}} \\ -\mathbf{R}^{tt}_{\mathrm{tu}} & \mathbf{R}^{tu}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{u}}_{\mathrm{u}} \\ \bar{\mathbf{t}}_{\mathrm{t}} \end{Bmatrix} \tag{3-32}$$

Considering the following symmetric properties of the kernels

$$\begin{aligned} G^{uu}_{ij}(\mathbf{p},\mathbf{Q}) &= G^{uu}_{ji}(\mathbf{Q},\mathbf{p}) \\ G^{tt}_{ij}(\mathbf{p},\mathbf{Q}) &= G^{tt}_{ji}(\mathbf{Q},\mathbf{p}) \\ G^{ut}_{ij}(\mathbf{p},\mathbf{Q}) &= G^{tu}_{ji}(\mathbf{Q},\mathbf{p}) \end{aligned} \tag{3-33}$$

it is then straightforward to rewrite the system as

$$\begin{bmatrix} \mathbf{R}^{uu}_{\mathrm{uu}} & -\mathbf{R}^{ut}_{\mathrm{ut}} \\ -(\mathbf{R}^{ut}_{\mathrm{ut}})^T & \mathbf{R}^{tt}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \mathbf{t}_{\mathrm{u}} \\ \mathbf{u}_{\mathrm{t}} \end{Bmatrix} = \begin{bmatrix} \mathbf{R}^{ut}_{\mathrm{uu}} & -\mathbf{R}^{uu}_{\mathrm{ut}} \\ -\mathbf{R}^{tt}_{\mathrm{tu}} & \mathbf{R}^{tu}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{u}}_{\mathrm{u}} \\ \bar{\mathbf{t}}_{\mathrm{t}} \end{Bmatrix} \tag{3-34}$$

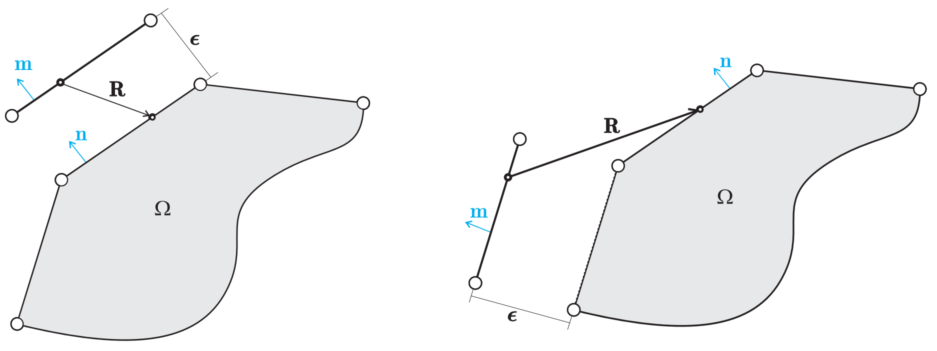
where the symmetry is now explicit.

It is worth mentioning that, according to [47], the Galerkin method preserves the properties of the partial differential operators which means that elliptic boundary value problems, such as linear elasticity, results in positive definite system matrices. Although this property holds directly for the left-hand side system of Equation (3-28) and for the right-hand side system of Equation (3-29) (with opposite sign), the positive definiteness of the SGBEM equations is only obtained if the system is rewritten in its block skew-symmetric form, given as

$$\begin{bmatrix} \mathbf{R}^{uu}_{\mathrm{uu}} & -\mathbf{R}^{ut}_{\mathrm{ut}} \\ (\mathbf{R}^{ut}_{\mathrm{ut}})^T & -\mathbf{R}^{tt}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \mathbf{t}_{\mathrm{u}} \\ \mathbf{u}_{\mathrm{t}} \end{Bmatrix} = \begin{bmatrix} \mathbf{R}^{ut}_{\mathrm{uu}} & -\mathbf{R}^{uu}_{\mathrm{ut}} \\ \mathbf{R}^{tt}_{\mathrm{tu}} & -\mathbf{R}^{tu}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{u}}_{\mathrm{u}} \\ \bar{\mathbf{t}}_{\mathrm{t}} \end{Bmatrix} \tag{3-35}$$

## 3.3 Singular Integrals

The main stumbling block towards the numerical implementation of the SGBEM is the treatment of integrals involving the kernel with singularity of type $r^{-2}$ (hypersingularity). To carry out the evaluation of these integrals a semi analytical method [43], which relies on the *limit to the boundary* approach, is employed. In this method the source element is shifted outside the boundary, in the normal direction, by a small distance $\varepsilon$. Then, given that no singularities are present in this case, the integrals become amenable analytical evaluation. Afterwards, with the analytical expressions in hand, the limit as $\varepsilon \to 0$ is taken. Figure 3.2 shows the setting of a shifted straight element in both the adjacent and coincident cases.



3.2(a): Coincident case. 3.2(b): Adjacent case.

Figure 3.2: Shifted element setting.

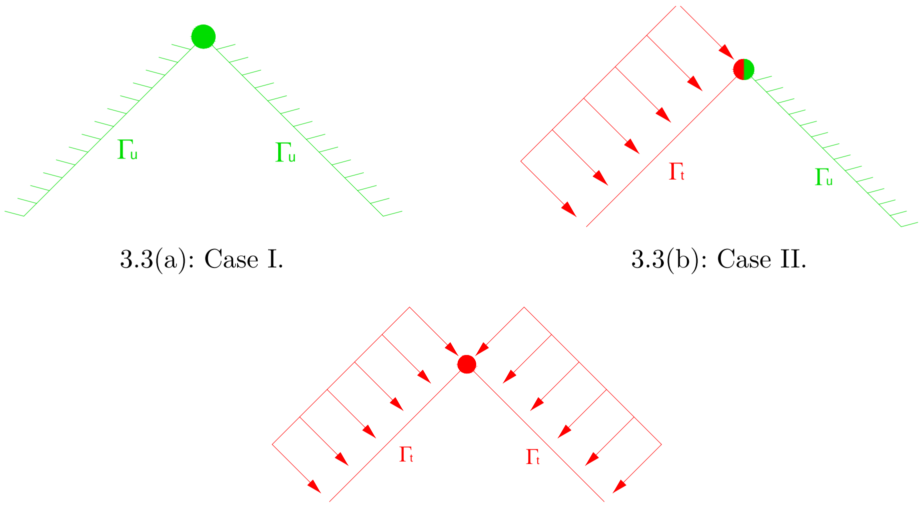
It is worth noting that in the case of straight elements the expressions are simple enough to be integrated analytically with respect to both variables $\xi$ and $\eta$. However, an additional step has to be taken in order to allow for analytical treatment of the integrals involving curved elements. This step is handled by an analytical regularization of the potentially singular terms by means of an algebraic manipulation. This regularization procedure separates the curvature contribution from a much more simplified expression, which looks essentially like the linear element expression. The curvature contribution becomes regular and therefore can be evaluated by means of a standard numerical quadrature. As for the linear contribution, the expression becomes simple enough to be integrated analytically with respect to one of the variables, $\xi$ or $\eta$. After this inner integration is obtained the outer integral becomes weakly singular and hence may be integrated numerically by a suitable quadrature scheme. Another interesting feature of this approach is that the limit as $\varepsilon \to 0$ is not immediately finite, actually there remains a divergent term of the type $\log(\varepsilon)$. Luckily the divergent term appears both in the coincident and adjacent cases, with opposite signs, allowing for its exactly cancellation in the corner treatment stage.

## 3.4 Corner Treatment

A nice feature of the SGBEM approach concerns the corner treatment stage. Unlike the CM approach, which handles the corners by an *ad hoc* procedure, in SGBEM the treatment of corners is handled in a natural and elegant way. This feature is a consequence of the use of the Galerkin method which allows, in a first analysis, to consider the degrees of freedom of each element independently of it neighbors, i.e. allows for discontinuous fields along the boundary. Therefore, the corner treatment stage, in SGBEM, corresponds to the enforcement of continuity of the fields of interest.

For elastostatics problems this stage corresponds to the enforcement of continuity of the displacements field, as it is required by the theory of elasticity. On the other hand, the tractions field may be left discontinuous, thus no further treatment is needed. In order to comply with these requirements, the three cases, as depicted in Figure 3.3, are considered.

The first case occurs when two adjacent elements are prescribed with displacement values and, consequently, there are two unknown nodal values of traction at the common node. Since the traction field is assumed to be discontinuous, this situation does not demand any further treatment. The second case arises in a situation where two adjacent elements are prescribed with different types of boundary conditions, i.e. one belongs to $\Gamma_t$ and the other to $\Gamma_u$. In this case two different values of displacements will occur in the node shared by the two elements, one being the prescribed displacement value and the other the unknown one. In order to enforce the continuity of the displacement field, the unknown value of displacement is imposed with the



3.3(a): Case I. 3.3(b): Case II. 3.3(c): Case III.

Figure 3.3: Corner treatment cases.

same value as the prescribed one. In terms of implementation, this procedure may be carried out likewise the enforcement of prescribed displacements in FEM, i.e. by static condensation of the final system. The third case occurs when two adjacent elements are prescribed with traction values. In this case the continuity of the displacement field is imposed by summing the equations of both degrees of freedom of the common node. In terms of implementation this procedure corresponds to the imposition of only one degree of freedom which is shared by both elements.

## 3.5 Stress Post-processing

To obtain stresses inside the domain, the Equation (3-8) may be employed in a straightforward manner, since no singularities are present in this case. On the other hand, the evaluation of stresses on the boundary involves hypersingular integrals, arising from the kernels of Equation (3-8), and cannot be evaluated directly. To overcome this problem, several authors, e.g. [39, 22], propose a simple differentiation of the displacement field in terms of the shape functions. Although this approach is less time consuming, given that no boundary integrals are needed, it leads to a discontinuous stress field and, as in FEM, an *ad hoc* scheme must be employed in order to enforce continuity of the stresses. On the other hand, despite its cumbersome implementation, the first approach is considered very accurate and therefore is ideally suited for problems with stress concentration. Since the Galerkin method has successfully enabled the treatment of the hypersingular integrals involved in the SGBEM approach, the same strategy may be employed in the evaluation of the boundary stresses.

Differentiating the interior displacement BIE (Equation (3-12)) with respect to the coordinates of the source point, one obtains

$$\frac{\partial u_i(P)}{\partial x_P} = \lim_{P_I \to P} \left[ \int_\Gamma \frac{\partial G_{ij}^{uu}(P_I,Q)}{\partial x_P} t_i(Q) d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_I,Q)}{\partial x_P} u_i(Q) d\Gamma \right] \quad (3\text{-}36)$$

$$\frac{\partial u_i(P)}{\partial y_P} = \lim_{P_I \to P} \left[ \int_\Gamma \frac{\partial G_{ij}^{uu}(P_I,Q)}{\partial y_P} t_i(Q) d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_I,Q)}{\partial y_P} u_i(Q) d\Gamma \right] \quad (3\text{-}37)$$

Proceeding with the same numerical implementation as in the SGBEM, and noting that the displacements and tractions are known *a priori*, these boundary equations lead to the following systems of equations

$$\mathbf{A}\mathbf{u}_{,x_P} = \mathbf{b}_x \quad (3\text{-}38)$$

$$\mathbf{A}\mathbf{u}_{,y_P} = \mathbf{b}_y \quad (3\text{-}39)$$

where $\mathbf{A}$ is a block diagonal matrix obtained by the product of shape functions, $\mathbf{u}_{,x_P}$ and $\mathbf{u}_{,y_P}$ are the nodal values of the derivatives of displacements and $\mathbf{b}_x$, $\mathbf{b}_y$ are obtained by evaluating the double integrals on the right hand side of Equations (3-36) and (3-37).

It is worth noting that this approach leads to the evaluation of a double integral over the boundary which consists in the most time consuming step of the method. Hence, in order to drastically reduce this computation, a strategy which uses both the interior and exterior limits to boundary is proposed in [44].

Differentiating the exterior displacement BIE (Equation (3-14)) with respect to the coordinates of the source point

$$0 = \lim_{P_E \to P} \left[ \int_\Gamma \frac{\partial G_{ij}^{uu}(P_E,Q)}{\partial x_P} t_i(Q) d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_E,Q)}{\partial x_P} u_i(Q) d\Gamma \right] \quad (3\text{-}40)$$

$$0 = \lim_{P_E \to P} \left[ \int_\Gamma \frac{\partial G_{ij}^{uu}(P_E,Q)}{\partial y_P} t_i(Q) d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_E,Q)}{\partial y_P} u_i(Q) d\Gamma \right] \quad (3\text{-}41)$$

and taking the difference between the interior and exterior limit gives

$$\frac{\partial u_i(P)}{\partial x_P} = \left( \lim_{P_I \to P} - \lim_{P_E \to P} \right) \left[ \int_\Gamma \frac{\partial G_{ij}^{uu}(P_{I-E},Q)}{\partial x_P} t_i(Q) d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_{I-E},Q)}{\partial x_P} u_i(Q) d\Gamma \right] \quad (3\text{-}42)$$

$$\frac{\partial u_i(P)}{\partial y_P} = \left(\lim_{P_I \to P} - \lim_{P_E \to P}\right)\left[\int_\Gamma \frac{\partial G_{ij}^{uu}(P_{I-E},Q)}{\partial y_P} t_i(Q) d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_{I-E},Q)}{\partial y_P} u_i(Q) d\Gamma\right] \quad (3\text{-}43)$$

Given that the exterior and interior limits are the same for the cases where no singularities are present, it is straightforward to observe that all the regular integrals, which are by far the most time consuming, vanish. The only integrals which remains after this process are the singular ones, for which analytical solutions are available, leading to a much faster computation of the right-hand side expression.

With the nodal displacement derivatives values in hand it is then straightforward to obtain the stresses on the boundary using the compatibility and constitutive equations. In the case of plane strain, this relation is given as

$$\begin{aligned} \sigma_{xx} &= \frac{2\mu\nu}{1-2\nu}\left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y}\right) + 2\mu\left(\frac{\partial u_x}{\partial x}\right) \\ \sigma_{yy} &= \frac{2\mu\nu}{1-2\nu}\left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y}\right) + 2\mu\left(\frac{\partial u_y}{\partial y}\right) \\ \sigma_{xy} &= \mu\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) \end{aligned} \quad (3\text{-}44)$$

For a plane stress state this relation may be readily obtained by modifying the material properties as shown in Equation 3-7.

# 4
# Sensitivity Analysis

In the context of structural optimization, the sensitivity analysis stage consists in obtaining the derivatives of given performance measures with respect to the design variables. In order to obtain such derivatives, three methods are often employed, namely the *finite difference method*, the *adjoint method* and the *direct method*. The *finite difference method* is the simplest one in terms of implementation, however it also presents itself as the least accurate and efficient among them. The *adjoint method* is indicated for problems in which the number of design variables exceeds the number of constraints. Hence, since in shape optimization problems the number of constraints is generally greater than the number of design variables, the *adjoint method* is particularly inappropriate. On the other hand, the *direct method* is indicated in the case that the number of design variables exceeds the number of constraints and therefore it is well-suited for shape optimization problems.

In the context of BEM, the *direct method* consists in the differentiation of the boundary integral equations, in its strong form, leading to analytical expressions for the displacements and tractions derivatives. On the other hand, in the context of FEM, the *direct method* is based on the differentiation of the equilibrium equations in its discrete weak form, leading to a semi-analyical derivative of the displacements. From this observation it can be inferred that, from a theoretical point of view, the sensitivity analysis with BEM is more accurate than with FEM.

In the following the derivatives of the boundary integral equations are introduced and later the SGBEM approach is employed to obtain the nodal derivatives values of both displacement and traction fields.

## 4.1
## SGBEM sensitivity equations

Differentiation of the boundary integral equations of the exterior, defined in Equations (3-3) and (3-9) , with respect to the design variables, leads to

$$\begin{aligned}&\int_{\Gamma} \dot{G}^{uu}_{ij}(P,Q)\,t_j(Q)\,d\Gamma + \int_{\Gamma} G^{uu}{}_{ij}(P,Q)\,t_j(Q)\,d\dot{\Gamma} + \int_{\Gamma} G^{uu}{}_{ij}(P,Q)\,\dot{t}_j(Q)\,d\Gamma \\ &- \int_{\Gamma} \dot{G}^{ut}_{ij}(P,Q)\,u_j(Q)\,d\Gamma - \int_{\Gamma} G^{ut}{}_{ij}(P,Q)\,u_j(Q)\,d\dot{\Gamma} - \int_{\Gamma} G^{ut}{}_{ij}(P,Q)\,\dot{u}_j(Q)\,d\Gamma = 0 \qquad (4\text{-}1)\end{aligned}$$

and

$$\begin{aligned}&\int_\Gamma \dot{G}^{tu}_{ij}\left(P,Q\right)t_j\left(Q\right)d\Gamma+\int_\Gamma G^{tu}{}_{ij}\left(P,Q\right)t_j\left(Q\right)d\dot{\Gamma}+\int_\Gamma G^{tu}{}_{ij}\left(P,Q\right)\dot{t}_j\left(Q\right)d\Gamma\\&-\int_\Gamma \dot{G}^{tt}_{ij}\left(P,Q\right)u_j\left(Q\right)d\Gamma-\int_\Gamma G^{tt}{}_{ij}\left(P,Q\right)u_j\left(Q\right)d\dot{\Gamma}-\int_\Gamma G^{tt}{}_{ij}\left(P,Q\right)\dot{u}_j\left(Q\right)d\Gamma=0\end{aligned}\tag{4-2}$$

where the superimposed dot denotes the derivative with respect to a design variable.

The *limit to the boundary* approach leads to

$$\begin{aligned}\lim_{P_E\to P}&\left[\int_\Gamma \dot{G}^{uu}_{ij}\left(P_E,Q\right)t_j\left(Q\right)d\Gamma+\int_\Gamma G^{uu}{}_{ij}\left(P_E,Q\right)t_j\left(Q\right)d\dot{\Gamma}+\int_\Gamma G^{uu}{}_{ij}\left(P_E,Q\right)\dot{t}_j\left(Q\right)d\Gamma\right.\\&\left.-\int_\Gamma \dot{G}^{ut}_{ij}\left(P_E,Q\right)u_j\left(Q\right)d\Gamma-\int_\Gamma G^{ut}{}_{ij}\left(P_E,Q\right)u_j\left(Q\right)d\dot{\Gamma}-\int_\Gamma G^{ut}{}_{ij}\left(P_E,Q\right)\dot{u}_j\left(Q\right)d\Gamma\right]=0\end{aligned}\tag{4-3}$$

and

$$\begin{aligned}\lim_{P_E\to P}&\left[\int_\Gamma \dot{G}^{tu}_{ij}\left(P_E,Q\right)t_j\left(Q\right)d\Gamma+\int_\Gamma G^{tu}{}_{ij}\left(P_E,Q\right)t_j\left(Q\right)d\dot{\Gamma}+\int_\Gamma G^{tu}{}_{ij}\left(P_E,Q\right)\dot{t}_j\left(Q\right)d\Gamma\right.\\&\left.-\int_\Gamma \dot{G}^{tt}_{ij}\left(P_E,Q\right)u_j\left(Q\right)d\Gamma-\int_\Gamma G^{tt}{}_{ij}\left(P_E,Q\right)u_j\left(Q\right)d\dot{\Gamma}-\int_\Gamma G^{tt}{}_{ij}\left(P_E,Q\right)\dot{u}_j\left(Q\right)d\Gamma\right]=0\end{aligned}\tag{4-4}$$

The interpolation of derivative nodal values may be carried out by using the same shape functions employed in the discretization of the problem, *i.e.*

$$\dot{\mathbf{u}}\left(\xi\right)=\begin{bmatrix}\dot{u}_x\left(\xi\right)\\\dot{u}_y\left(\xi\right)\end{bmatrix}=\mathbf{N}\left(\xi\right)\dot{\mathbf{u}}_e;\quad \dot{\mathbf{t}}\left(\xi\right)=\begin{bmatrix}\dot{t}_x\left(\xi\right)\\\dot{t}_y\left(\xi\right)\end{bmatrix}=\mathbf{N}\left(\xi\right)\dot{\mathbf{t}}_e;\tag{4-5}$$

Employing the Galerkin method to equations (4-3) and (4-4) results in

$$\begin{aligned}&\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\dot{\mathbf{I}}^{tu}\left(\xi,\eta\right)d\xi d\eta\right)\mathbf{t}_{eQ}+\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\mathbf{I}^{tu}\left(\xi,\eta\right)d\xi d\eta\right)\dot{\mathbf{t}}_{eQ}=\\&\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\dot{\mathbf{I}}^{tt}\left(\xi,\eta\right)d\xi d\eta\right)\mathbf{u}_{eQ}+\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\mathbf{I}^{tt}\left(\xi,\eta\right)d\xi d\eta\right)\dot{\mathbf{u}}_{eQ}\end{aligned}\tag{4-6}$$

$$\begin{aligned}&\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\dot{\mathbf{I}}^{uu}\left(\xi,\eta\right)d\xi d\eta\right)\mathbf{t}_{eQ}+\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\mathbf{I}^{uu}\left(\xi,\eta\right)d\xi d\eta\right)\dot{\mathbf{t}}_{eQ}=\\&\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\dot{\mathbf{I}}^{ut}\left(\xi,\eta\right)d\xi d\eta\right)\mathbf{u}_{eQ}+\sum_{ep=1}^{Ne}\sum_{eQ=1}^{Ne}\left(\lim_{P_E\to P}\int_0^1\int_0^1\mathbf{I}^{ut}\left(\xi,\eta\right)d\xi d\eta\right)\dot{\mathbf{u}}_{eQ}\end{aligned}\tag{4-7}$$

where

$$\begin{aligned}\dot{\mathbf{I}}^{uu}\left(\xi,\eta\right)&=\mathbf{N}^T\left(\eta\right)\left[\dot{\mathbf{G}}^{uu}\left(P_E\left(\eta\right),Q\left(\xi\right)\right)J_{eQ}+\mathbf{G}^{uu}\left(P_E\left(\eta\right),Q\left(\xi\right)\right)\dot{J}_{eQ}\right]\mathbf{N}\left(\xi\right)J_{ep}\\\dot{\mathbf{I}}^{ut}\left(\xi,\eta\right)&=\mathbf{N}^T\left(\eta\right)\left[\dot{\mathbf{G}}^{ut}\left(P_E\left(\eta\right),Q\left(\xi\right)\right)J_{eQ}+\mathbf{G}^{ut}\left(P_E\left(\eta\right),Q\left(\xi\right)\right)\dot{J}_{eQ}\right]\mathbf{N}\left(\xi\right)J_{ep}\\\dot{\mathbf{I}}^{tu}\left(\xi,\eta\right)&=\mathbf{N}^T\left(\eta\right)\left[\dot{\mathbf{G}}^{tu}\left(P_E\left(\eta\right),Q\left(\xi\right)\right)J_{eQ}+\mathbf{G}^{tu}\left(P_E\left(\eta\right),Q\left(\xi\right)\right)\dot{J}_{eQ}\right]\mathbf{N}\left(\xi\right)J_{ep}\\\dot{\mathbf{I}}^{tt}\left(\xi,\eta\right)&=\mathbf{N}^T\left(\eta\right)\left[\dot{\mathbf{G}}^{tt}\left(P_E\left(\eta\right),Q\left(\xi\right)\right)J_{eQ}+\mathbf{G}^{tt}\left(P_E\left(\eta\right),Q\left(\xi\right)\right)\dot{J}_{eQ}\right]\mathbf{N}\left(\xi\right)J_{ep}\end{aligned}\tag{4-8}$$

Based on the same assembly strategy of the SGBEM the discrete equations are given as

$$\begin{bmatrix} \mathbf{R}^{uu}_{\mathrm{uu}} & -\mathbf{R}^{ut}_{\mathrm{ut}} \\ -(\mathbf{R}^{ut}_{\mathrm{ut}})^T & \mathbf{R}^{tt}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{t}}_{\mathrm{u}} \\ \dot{\mathbf{u}}_{\mathrm{t}} \end{Bmatrix} = \begin{bmatrix} \mathbf{R}^{ut}_{\mathrm{uu}} & -\mathbf{R}^{uu}_{\mathrm{ut}} \\ -\mathbf{R}^{tt}_{\mathrm{tu}} & \mathbf{R}^{tu}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \dot{\bar{\mathbf{u}}}_{\mathrm{u}} \\ \dot{\bar{\mathbf{t}}}_{\mathrm{t}} \end{Bmatrix} + \begin{bmatrix} \dot{\mathbf{R}}^{ut}_{\mathrm{uu}} & -\dot{\mathbf{R}}^{uu}_{\mathrm{ut}} \\ -\dot{\mathbf{R}}^{tt}_{\mathrm{tu}} & \dot{\mathbf{R}}^{tu}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{u}}_{\mathrm{u}} \\ \bar{\mathbf{t}}_{\mathrm{t}} \end{Bmatrix} + \begin{bmatrix} -\dot{\mathbf{R}}^{uu}_{\mathrm{uu}} & \dot{\mathbf{R}}^{ut}_{\mathrm{ut}} \\ (\dot{\mathbf{R}}^{ut}_{\mathrm{ut}})^T & -\dot{\mathbf{R}}^{tt}_{\mathrm{tt}} \end{bmatrix} \begin{Bmatrix} \mathbf{t}_{\mathrm{u}} \\ \mathbf{u}_{\mathrm{t}} \end{Bmatrix} \tag{4-9}$$

where $\dot{\mathbf{t}}_u$ e $\dot{\mathbf{u}}_t$ are, respectively, unknown values of the displacements and tractions derivatives, $\dot{\bar{\mathbf{t}}}_t$ e $\dot{\bar{\mathbf{u}}}_u$ are the differentiated prescribed values of displacements and tractions and submatrices $\dot{\mathbf{R}}$ are obtained by the integrals defined in Equations (4-6) and (4-7)

## 4.2 Fundamental Solution Sensitivities

The derivatives of the fundamental solutions with respect to the design variables are given as

$$\dot{G}^{uu}_{ij}(P,Q) = \frac{1}{8\pi\mu(1-\nu)} \left[ (3-4\nu) \dot{\left( \ln\left(\frac{1}{r}\right)\right)} \delta_{ij} + \dot{r}_{,i}\, r_{,j} + r_{,i}\, \dot{r}_{,j} \right] \tag{4-10}$$

$$\begin{aligned} \dot{G}^{ut}_{ij}(P,Q) = \tfrac{-1}{4\pi(1-\nu)} \Bigg\{ & \dot{\left(\tfrac{1}{r}\right)} \left[ \tfrac{\partial r}{\partial n} \left[ (1-2\nu)\, \delta_{ij} + 2 r_{,i} r_{,j} \right] + (1-2\nu) \left( n_i r_{,j} - n_k r_{,j} \right) \right] \\ & + \tfrac{1}{r} \left[ \tfrac{\dot{\partial r}}{\partial n} \left[ (1-2\nu)\, \delta_{ij} + 2 r_{,i} r_{,j} \right] + \tfrac{\partial r}{\partial n} \left[ 2 \left( \dot{r}_{,i}\, r_{,j} + r_{,i}\, \dot{r}_{,j} \right) \right] \right] \\ & + \tfrac{1}{r} \left[ (1-2\nu) \left( \dot{n}_i\, r_{,j} + n_i\, \dot{r}_{,j} - \dot{n}_k\, r_{,j} - n_k\, \dot{r}_{,j} \right) \right] \Bigg\} \end{aligned} \tag{4-11}$$

$$\begin{aligned} \dot{G}^{tt}_{ij}(P,Q) = \tfrac{\mu}{2\pi(1-\nu)} = \Bigg\{ & \dot{\left(\tfrac{1}{r^2}\right)} \left[ \left( \delta_{ij}\delta_{kw} + \delta_{ki}\delta_{jw} - \delta_{jk}\delta_{iw} \right) m_{,k} n_{,w} \right] \\ & \dot{\left(\tfrac{1}{r^2}\right)} \left[ \left( +2\delta_{jk} r_{,i} r_{,w} + 2\delta_{iw} r_{,j} r_{,k} - 8 r_{,i} r_{,w} r_{,j} r_{,k} \right) m_{,k} n_{,w} \right] \\ & + \tfrac{1}{r^2} \left( \delta_{ij}\delta_{kw} + \delta_{ki}\delta_{jw} - \delta_{jk}\delta_{iw} + 2\delta_{jk} r_{,i} r_{,w} \right) \left( \dot{m}_{,k}\, n_{,w} + m_{,k}\, \dot{n}_{,w} \right) \\ & + \tfrac{1}{r^2} \left( +2\delta_{iw} r_{,j} r_{,k} - 8 r_{,i} r_{,w} r_{,j} r_{,k} \right) \left( \dot{m}_{,k}\, n_{,w} + m_{,k}\, \dot{n}_{,w} \right) \\ & + \tfrac{1}{r^2} \left[ 2\delta_{jk} \left( \dot{r}_{,i}\, r_{,w} + r_{,i}\, \dot{r}_{,w} \right) + 2\delta_{iw} \left( \dot{r}_{,j}\, r_{,k} + r_{,j}\, \dot{r}_{,k} \right) \right] m_{,k} n_{,w} \\ & - \tfrac{1}{r^2} \left[ 8 \left( \dot{r}_{,i}\, r_{,w} r_{,j} r_{,k} + r_{,i}\, \dot{r}_{,w}\, r_{,j} r_{,k} + r_{,i} r_{,w}\, \dot{r}_{,j}\, r_{,k} + r_{,i} r_{,w} r_{,j}\, \dot{r}_{,k} \right) \right] m_{,k} n_{,w} \Bigg\} \end{aligned} \tag{4-12}$$

where

$$\dot{\left( \ln\left(\frac{1}{r}\right)\right)} = -\frac{\dot{r}}{r} \tag{4-13}$$

$$\dot{(r_{,i})} = \frac{\dot{R}_i\, r - R_i\, \dot{r}}{r^2} \tag{4-14}$$

$$R_i = Q_i - p_i; \tag{4-15}$$

$$\dot{R}_i = \dot{Q}_i - \dot{p}_i; \tag{4-16}$$

$$\dot{n}_x = \tfrac{1}{J_{eQ}{}^2} \left[ \dot{\left( \tfrac{\partial Q_y}{\partial \xi} \right)} J_{eQ} - \tfrac{\partial Q_y}{\partial \xi}\, \dot{J}_{eQ} \right]; \quad \dot{m}_x = \tfrac{1}{J_{ep}{}^2} \left[ \dot{\left( \tfrac{\partial P_y}{\partial \xi} \right)} J_{ep} - \tfrac{\partial P_y}{\partial \xi}\, \dot{J}_{ep} \right]; \tag{4-17}$$

$$\dot{n}_y = -\tfrac{1}{J_{eQ}{}^2} \left[ \dot{\left( \tfrac{\partial Q_x}{\partial \xi} \right)} J_{eQ} - \tfrac{\partial Q_x}{\partial \xi}\, \dot{J}_{eQ} \right]; \quad \dot{m}_y = -\tfrac{1}{J_{ep}{}^2} \left[ \dot{\left( \tfrac{\partial P_x}{\partial \xi} \right)} J_{ep} - \tfrac{\partial P_x}{\partial \xi}\, \dot{J}_{ep} \right]; \tag{4-18}$$

$$\dot{J}_{eQ} = \frac{1}{J_{eQ}} \left[ \frac{\partial Q_x}{\partial \xi} \left( \frac{\partial \dot{Q}_x}{\partial \xi} \right) + \frac{\partial Q_y}{\partial \xi} \left( \frac{\partial \dot{Q}_y}{\partial \xi} \right) \right]; \quad \dot{J}_{ep} = \frac{1}{J_{ep}} \left[ \frac{\partial P_x}{\partial \xi} \left( \frac{\partial \dot{P}_x}{\partial \xi} \right) + \frac{\partial P_y}{\partial \xi} \left( \frac{\partial \dot{P}_y}{\partial \xi} \right) \right]; \tag{4-19}$$

It is worth noting that differentiating the fundamental solutions does not alter its singularity, thus the same treatment of the singular integrals, previously introduced, may also be employed for the singular integrals involving the derivatives of the fundamental solutions. However, given the great length of these expressions, the analytical treatment of these integrals is often cumbersome. Fortunately, the following theorem allows for a very simple way of obtaining such results.

**Theorem 1** *If $f$ and $\frac{\partial f}{\partial t}$ are continuous for $(x,t) \in [a,b] \times [\alpha,\beta]$, and $[a,b]$ is finite, then the function*

$$\Phi(t) = \int_a^b f(x,t)\,dx \tag{4-20}$$

*is differentiable and*

$$\frac{d\Phi(t)}{dt} = \frac{d}{dt} \int_a^b f(x,t)\,dx = \int_a^b \frac{\partial f(x,t)}{\partial t} dx \tag{4-21}$$

For a proof see, for example, [48].

Therefore, a straightforward way to obtain the analytical expressions for the singular integrals of Equations (4-6) and (4-7) is to simply differentiate the analytical results previously obtained for the singular integrals.

## 4.3 Stress sensitivities post-processing

The evaluation of the stress derivatives on the boundary may be carried out by a procedure similar to the technique applied in stress post-processing stage.

Differentiation of the boundary integral equations, for the interior of the domain, defined in Equations (3-36) and (3-37) , leads to

$$\begin{aligned} \frac{\partial \dot{u}_i(P)}{\partial x_P} = \lim_{P_I \to P} \Bigg[ & \int_\Gamma \frac{\partial \dot{G}^{uu}_{ij}(P_I,Q)}{\partial x_P}\, t_i(Q) d\Gamma + \int_\Gamma \frac{\partial G^{uu}_{ij}(P_I,Q)}{\partial x_P}\, \dot{t}_i(Q)\, d\Gamma + \int_\Gamma \frac{\partial G^{uu}_{ij}(P_I,Q)}{\partial x_P} t_i(Q)\, d\dot{\Gamma} \\ & - \int_\Gamma \frac{\partial \dot{G}^{ut}_{ij}(P_I,Q)}{\partial x_P}\, u_i(Q) d\Gamma - \int_\Gamma \frac{\partial G^{ut}_{ij}(P_I,Q)}{\partial x_P}\, \dot{u}_i(Q)\, d\Gamma - \int_\Gamma \frac{\partial G^{ut}_{ij}(P_I,Q)}{\partial x_P} u_i(Q)\, d\dot{\Gamma} \Bigg] \end{aligned} \tag{4-22}$$

$$\begin{aligned} \frac{\partial \dot{u}_i(P)}{\partial y_P} = \lim_{P_I \to P} \Bigg[ & \int_\Gamma \frac{\partial \dot{G}^{uu}_{ij}(P_I,Q)}{\partial y_P}\, t_i(Q) d\Gamma + \int_\Gamma \frac{\partial G^{uu}_{ij}(P_I,Q)}{\partial y_P}\, \dot{t}_i(Q)\, d\Gamma + \int_\Gamma \frac{\partial G^{uu}_{ij}(P_I,Q)}{\partial y_P} t_i(Q)\, d\dot{\Gamma} \\ & - \int_\Gamma \frac{\partial \dot{G}^{ut}_{ij}(P_I,Q)}{\partial y_P}\, u_i(Q) d\Gamma - \int_\Gamma \frac{\partial G^{ut}_{ij}(P_I,Q)}{\partial y_P}\, \dot{u}_i(Q)\, d\Gamma - \int_\Gamma \frac{\partial G^{ut}_{ij}(P_I,Q)}{\partial y_P} u_i(Q)\, d\dot{\Gamma} \Bigg] \end{aligned} \tag{4-23}$$

Differentiation of the boundary integral equations, for the exterior, defined in Equations (3-40) and (3-41), leads to

$$\begin{aligned} 0 = \lim_{P_E \to P} \Bigg[ & \int_\Gamma \frac{\partial \dot{G}_{ij}^{uu}(P_E,Q)}{\partial y_P} t_i(Q) d\Gamma + \int_\Gamma \frac{\partial G_{ij}^{uu}(P_E,Q)}{\partial y_P} \dot{t}_i(Q)\, d\Gamma + \int_\Gamma \frac{\partial G_{ij}^{uu}(P_E,Q)}{\partial y_P} t_i(Q)\, d\dot{\Gamma} \\ & - \int_\Gamma \frac{\partial \dot{G}_{ij}^{ut}(P_E,Q)}{\partial y_P} u_i(Q) d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_E,Q)}{\partial y_P} \dot{u}_i(Q)\, d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_E,Q)}{\partial y_P} u_i(Q)\, d\dot{\Gamma} \Bigg] \end{aligned} \tag{4-24}$$

$$\begin{aligned} 0 = \lim_{P_E \to P} \Bigg[ & \int_\Gamma \frac{\partial \dot{G}_{ij}^{uu}(P_E,Q)}{\partial y_P} t_i(Q) d\Gamma + \int_\Gamma \frac{\partial G_{ij}^{uu}(P_E,Q)}{\partial y_P} \dot{t}_i(Q)\, d\Gamma + \int_\Gamma \frac{\partial G_{ij}^{uu}(P_E,Q)}{\partial y_P} t_i(Q)\, d\dot{\Gamma} \\ & - \int_\Gamma \frac{\partial \dot{G}_{ij}^{ut}(P_E,Q)}{\partial y_P} u_i(Q) d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_E,Q)}{\partial y_P} \dot{u}_i(Q)\, d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_E,Q)}{\partial y_P} u_i(Q)\, d\dot{\Gamma} \Bigg] \end{aligned} \tag{4-25}$$

Taking the difference between the interior and exterior equations gives

$$\begin{aligned} \frac{\partial \dot{u}_i(P)}{\partial y_P} = \left( \lim_{P_I \to P} - \lim_{P_E \to P} \right) \Bigg[ & \int_\Gamma \frac{\partial \dot{G}_{ij}^{uu}(P_{I-E},Q)}{\partial y_P} t_i(Q) d\Gamma + \int_\Gamma \frac{\partial G_{ij}^{uu}(P_{I-E},Q)}{\partial y_P} \dot{t}_i(Q)\, d\Gamma + \int_\Gamma \frac{\partial G_{ij}^{uu}(P_{I-E},Q)}{\partial y_P} t_i(Q)\, d\dot{\Gamma} \\ & - \int_\Gamma \frac{\partial \dot{G}_{ij}^{ut}(P_{I-E},Q)}{\partial y_P} u_i(Q) d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_{I-E},Q)}{\partial y_P} \dot{u}_i(Q)\, d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_{I-E},Q)}{\partial y_P} u_i(Q)\, d\dot{\Gamma} \Bigg] \end{aligned} \tag{4-26}$$

$$\begin{aligned} \frac{\partial \dot{u}_i(P)}{\partial x_P} = \left( \lim_{P_I \to P} - \lim_{P_E \to P} \right) \Bigg[ & \int_\Gamma \frac{\partial \dot{G}_{ij}^{uu}(P_{I-E},Q)}{\partial x_P} t_i(Q) d\Gamma + \int_\Gamma \frac{\partial G_{ij}^{uu}(P_{I-E},Q)}{\partial x_P} \dot{t}_i(Q)\, d\Gamma + \int_\Gamma \frac{\partial G_{ij}^{uu}(P_{I-E},Q)}{\partial x_P} t_i(Q)\, d\dot{\Gamma} \\ & - \int_\Gamma \frac{\partial \dot{G}_{ij}^{ut}(P_{I-E},Q)}{\partial x_P} u_i(Q) d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_{I-E},Q)}{\partial x_P} \dot{u}_i(Q)\, d\Gamma - \int_\Gamma \frac{\partial G_{ij}^{ut}(P_{I-E},Q)}{\partial x_P} u_i(Q)\, d\dot{\Gamma} \Bigg] \end{aligned} \tag{4-27}$$

Proceeding with the same numerical implementation as in SGBEM, and noting that the displacements, tractions and their derivatives are known *a priori*, these boundary integral equations lead to following linear systems

$$\mathbf{A}\, \dot{\mathbf{u}}_{,x_P} = \mathbf{c}_x \tag{4-28}$$

$$\mathbf{A}\, \dot{\mathbf{u}}_{,y_P} = \mathbf{c}_y \tag{4-29}$$

where $\mathbf{A}$ is the same block diagonal matrix obtained in the stress post processing stage, $\dot{\mathbf{u}}_{,x_P}$ and $\dot{\mathbf{u}}_{,y_P}$ are the nodal values of the derivatives of the differentiated displacements and $\mathbf{c}_x$ and $\mathbf{c}_y$ are obtained by evaluating the double integrals on the right hand side of Equations (4-26) and (4-27).

It is worth mentioning that, by employing this approach, all the regular integrals vanish and only the singular integrals must be evaluated. In addition, as aforementioned, the analytical expressions for these singular integrals may be readily obtained by appropriately switching the order of integration/differentiation.

With the derivatives of differentiated displacements in hand it is then straightforward to obtain the stresses derivatives on the boundary by simply differentiating the stress/displacements relations defined in Equation (3-44), *i.e.*

$$
\begin{aligned}
\dot{\sigma}_{xx} &= \tfrac{2\mu\nu}{1-2\nu}\left(\frac{\partial \dot{u}_x}{\partial x}+\frac{\partial \dot{u}_y}{\partial y}\right)+2\mu\left(\frac{\partial \dot{u}_x}{\partial x}\right)\\
\dot{\sigma}_{yy} &= \tfrac{2\mu\nu}{1-2\nu}\left(\frac{\partial \dot{u}_x}{\partial x}+\frac{\partial \dot{u}_y}{\partial y}\right)+2\mu\left(\frac{\partial \dot{u}_y}{\partial y}\right)\\
\dot{\sigma}_{xy} &= \mu\left(\frac{\partial \dot{u}_x}{\partial y}+\frac{\partial \dot{u}_y}{\partial x}\right)
\end{aligned}
\tag{4-30}
$$

# 5
# Structural Optimization with SOCP

The second-order cone programming (SOCP) [49] is a modern optimization technique which belongs to the extensive field of conic programming (CP) [50]. SOCP is capable of handling nonlinear convex problems including linear, quadratic and second-order cone constraints. Hence, linear programming (LP), convex quadratic programming (QP) and convex quadratically constrained quadratic programming (QCQP) are all particular cases of SOCP. Furthermore, robust and efficient primal-dual interior points algorithms are available for SOCP, thus allowing for fast solutions of large scale-optimization problems. To expose the versatility of SOCP, in [51] the authors introduce a great variety of problems which can be cast as SOCP problems, including sums and maximum of norms, problems with hyperbolic constraints and robust linear programming.

Despite its great features, the application of SOCP in engineering still remains an open field. The limited literature on this subject includes works like antenna array weight design [52], grasping force optimization [53] and multi-load truss topology optimization [50]. Nevertheless, the most expressive application of SOCP in civil engineering, specifically in geomechanics, is in the field of limit analysis [54].

Limit analysis is a structural analysis field in which the main goal is to estimate the collapse load of a given structural system in a efficient manner. In general, the limit analysis is formulated as a nonlinear optimization problem in which a given yield criterion is posed as a constraint. Particular yield criteria, such as the von Mises and Drucker-Prager, are amenable to be cast into a second-order cone constraint, thus allowing for the efficient and robust solution of such problems by the SOCP. Based on this idea, this work proposes the application of SOCP in the context of structural optimization with stress constraints. The proposed procedure relies on the solution of a sequence of SOCP subproblems in which the stress constrains are cast into second-order cone constraints.

## 5.1

## Second-order cone programming

Conic programming is a class of mathematical programming problems in which the objective function is convex and the constraint set is given as the intersection of an affine subspace with a convex cone. A general conic optimization problem may be stated as

$$\begin{cases} \min & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t.} & \begin{array}{l} \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \in \mathcal{K} \end{array} \end{cases} \tag{5-1}$$

where $\mathbf{x}$ are the design variables, $\mathbf{A}\mathbf{x} = \mathbf{b}$ is a set of linear constraints and $\mathcal{K}$ is the convex cone associated to the problem.

Some examples of conic programming are linear programming (LP), convex quadratic programming (Convex QP), second-order cone progamming (SOCP) and the semidefinite programming (SDP). Figure 5.1 depicts the different subfields belonging to the conic programming.

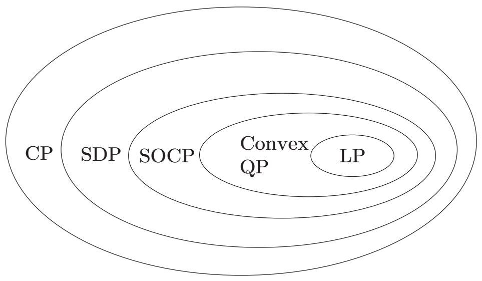


Figure 5.1: Conic programming subfields.

Each one of these subfields is distinguished by the type of cone associated to the problem. For example, in LP problems the associated cone is the so-called $\mathbb{R}_+$ cone, which is defined as

$$\mathbb{R}_+ = \{x \in \mathbb{R} | x \geq 0\} \tag{5-2}$$

$$\begin{cases} \min & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t.} & \begin{array}{l} \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq 0 \end{array} \end{cases} \quad \textit{or} \quad \begin{cases} \min & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t.} & \begin{array}{l} \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \in \mathbb{R}_+ \end{array} \end{cases} \tag{5-3}$$

Analogously, in SOCP problems the associated cone is called the second-order cone, also known as the ice-cream or the Lorentz cone, which is defined as

$$\mathcal{K}^q = \{\mathbf{x} \in \mathbb{R}^n | x_1 \geq \|x_{2:n}\|, x_1 \geq 0\} \tag{5-4}$$

Thus, a general SOCP problem may be written as

$$\begin{cases} \min & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \\ \text{s.t.} & x_1 \geq \|x_{2:n}\| \\ & x_1 \geq 0 \end{cases} \quad \mathit{or} \quad \begin{cases} \min & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t.} & \begin{array}{l} \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \in \mathcal{K}^q \end{array} \end{cases} \tag{5-5}$$

It is worth to mention that the $\mathcal{K}^q$ cones belongs to a class of cones called self-scaled (see [55] for a detailed definition). In [56] the extension of the primal-dual interior point algorithms for convex programming problems with self-scaled cones is presented. This formulation allows for a robust and highly efficient numerical implementations for solving SOCP. According to [51], regarding the interior-point algorithm, worst-case theoretical analysis shows that the number of iterations required to solve a SOCP problem grows at most as the square root of the number of design variables, while numerical experiments indicate that the typical number of iterations ranges between 5 and 50, almost independently of the problem size. This feature allows for the solution of large scale problems with minor computation expenses. A step by step numerical implementation of the primal-dual interior-point algorithm for conic quadratic optimization is introduced in [55].

## 5.2 Second-order cone representations of material yield criteria

One of the main steps involved in developing the proposed procedure is to cast the yield criteria/stress constraints into second-oder cone constraints. All the most common yield criteria used in practice are amenable to be cast into semidefinite conic form, while in some particular cases the criteria may be cast into second-order cone constraints. In [57] the conic representation of the most common yield criterion are introduced. Specifically, in such work, the Mohr-Coulomb, Rankine and Tresca criteria are all represented by positive semidefinite cones, while the von Mises and Drucker Prager criteria are rewritten as second-order conic constraints. To elucidate such process, the casting of the von Mises yield criterion into a second order cone is demonstrated next.

### Second-order cone representation of the von Mises criterion

The von Mises criterion states that yielding begins when the octahedral shearing stress reaches a critical value $k$. For a plane stress state the criterion is given as

$$\tau_{oct} = \sqrt{{\sigma_{xx}}^2 + {\sigma_{yy}}^2 - \sigma_{xx}\sigma_{yy} + 3{\tau_{xy}}^2} \leq k \tag{5-6}$$

where $\tau_{oct}$ is the octahedral shearing stress, $\sigma_{xx}$, $\sigma_{yy}$ and $\tau_{xy}$ are the Cauchy stress components and $k$ is the yield stress in pure shear.

The expression of Equation (5-6) may be written in the following matrix form

$$\tau_{oct} = \sqrt{\sigma \mathbf{M} \sigma} \leq k \tag{5-7}$$

where

$$\mathbf{M} := \begin{bmatrix} 1 & -0.5 & 0 \\ -0.5 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \; ; \; \sigma := \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} \; ; \tag{5-8}$$

Given the positive definiteness of matrix $\mathbf{M}$, a Cholesky factorization may be performed, resulting in

$$\mathbf{M} := \mathbf{L}^T \mathbf{L} \tag{5-9}$$

The following change of variables

$$\mathbf{y} = \mathbf{L}\sigma \tag{5-10}$$

leads to

$$\tau_{oct} = \sqrt{\sigma \mathbf{M} \sigma} = \sqrt{\mathbf{y}^T \mathbf{y}} \leq k \Leftrightarrow \tau_{oct} = \|\mathbf{y}\| \leq k \tag{5-11}$$

Defining $\mathbf{z}$ as

$$\mathbf{z} := \left[ \begin{array}{cccc} k & y_1 & y_2 & y_3 \end{array} \right] \tag{5-12}$$

then the criterion is restated as

$$\mathcal{C}^{VM} = \left\{ \mathbf{z} \in \mathbb{R}^4 | z_1 \geq \|z_{2:4}\|, z_1 \geq 0 \right\} \tag{5-13}$$

where the second-order cone $\mathcal{C}^{VM}$ is known as the von Mises cone.

## 5.3 Minimization of Stress Concentrations using SOCP

The minimization of stress concentrations plays a central role in the area of structural shape optimization. The main objective is to reduce the stress concentration effects which are usually caused by geometric discontinuities such as cracks, sharp corners and holes. Since these discontinuities usually occur on the boundary of the structure, the shape optimization framework is particularly suitable for this problem.

In general, the problem of minimizing stress concentrations is formulated as the minimization of the maximum stress measure subjected to a volume constraint. Typically the von Mises, e.g. [58] and [31], or the principal stresses, e.g.

[59] and [13], are chosen as the stress measure. A great inconvenient regarding this formulation is the lack of smoothness of the maximum function, thus impairing the global convergence of mathematical programming methods. To overcome this problem, in [58] the authors employ a differentiable approximation of the maximum function, often referred to as *smooth maximum*. However, this approach is based on numerical parameters which have to be well-adjusted in order to obtain a stable approximation. In [31], [13] and [60] the authors proposed an objective function based on the square deviation between the actual stresses and a target value. Unfortunately, a target stress is rarely known *a priori*, thus making this alternative not well-suited for practical applications. In [25] a bound formulation is proposed in which the *min-max* formulation is converted into a simple optimization problem in terms of an unknown bound on the stresses. Such bound formulation then states the problem as a linear objective function comprised of the unknown bound value and non-linear stress constraints. Following this idea, and based on a linear approximation of the stresses, both [25] and [59] propose the use of a sequential linear programming algorithm.

In this present work the optimization problem is solved by means of a sequential conic quadratic programming procedure. In this framework, the stress constraints are approximated by second order cone constraints and each subproblem is efficiently solved by a SOCP primal-dual interior point algorithm [61].

## 5.4 A sequential SOCP framework for minimization of stress concentrations

Based on the *min-max* formulation, and choosing the von Mises criterion as the stress measure, the problem of minimizing the stress concentrations may be state as the following non-linear optimization problem

$$\left\{ \begin{array}{ll} \min\limits_{\alpha} & \underset{i=1...n}{max}\, \sigma_i^{VM}(\alpha) \\ \text{s.t.} & \begin{array}{l} A(\alpha) <= A_o \\ \alpha_j^{\min} \leq \alpha_j \leq \alpha_j^{\max} \;\; \forall j = 1...m \end{array} \end{array} \right. \tag{5-14}$$

where $\alpha$ is the vector of design variables, $\alpha_j^{\min}$ and $\alpha_j^{\max}$ are upper and lower bounds the design variables, $\sigma_i^{VM}$ represents the von Mises stress value on *i-th* discretization node, $A_o$ is a prescribed volume fraction and $A(\alpha)$ is the volume of the structure.

The bound formulation implies in converting the optimization problem of Equation (5-14) into

$$\begin{cases} \min\limits_{t,\alpha} & t \\ & \sigma_i^{VM}(\alpha) <= t \\ \text{s.t.} & A(\alpha) <= A_o \\ & \alpha_j^{\min} \leq \alpha_j \leq \alpha_j^{\max} \ \forall j = 1...m \end{cases} \tag{5-15}$$

where the new variable $t$ is the unknown bound on the stresses.

Substituting the quadratic form of Equation (2-45), for the volume, and the matrix form of Equation (5-8), for the von Mises stresses, gives

$$\begin{cases} \min\limits_{t,\alpha} & t \\ & \sqrt{\sigma_i(\rho)^{\mathrm{T}}\mathbf{M}\,\sigma_i(\rho)} <= t \\ \text{s.t.} & \alpha^T\mathbf{Q}\alpha + \alpha^T\mathbf{b} + c <= A_o \\ & \alpha_j^{\min} \leq \alpha_j \leq \alpha_j^{\max} \ \forall j = 1...m \end{cases} \tag{5-16}$$

The first order Taylor's series expansion of stresses with respect to the design densities is given as

$$\sigma_i(\rho) \approx \sigma_i(\rho_0) + \nabla\sigma_i(\rho_0)\,\Delta\rho \tag{5-17}$$

where $\rho_0$ is the material distribution of the current iteration and

$$\Delta\rho = \rho - \rho_0 \tag{5-18}$$

Substituting the approximation of Equation (5-17) into the stress constraints of Equation (5-16) gives

$$\sqrt{(\sigma_i(\mathbf{x}_0) + \nabla\sigma_i(\mathbf{x}_0)\,\Delta\mathbf{x})\,\mathbf{L}^T\mathbf{L}\,(\sigma_i(\mathbf{x}_0) + \nabla\sigma_i(\mathbf{x}_0)\,\Delta\mathbf{x}_i)} - \sigma_y \leq 0 \tag{5-19}$$

Casting of the von Mises criterion into second-order conic constraints, the problem (5-16) may be approximated by the following SOCP subproblem.

$$\begin{cases} \min\limits_{t,\alpha} & t \\ & \mathbf{y}_i = \mathbf{L}\,(\sigma_i(\alpha_0) + \nabla\sigma_i(\alpha_0)\,\Delta\alpha) \ \ \forall i = 1...n \\ \text{s.t.} & \|\mathbf{y}_i\| \leq k \ \forall i = 1...n \\ & V(\alpha_0) + \nabla V(\alpha_0)\,\Delta\alpha \leq V_o \\ & \alpha_j^{\min} \leq \alpha_j \leq \alpha_j^{\max} \ \forall j = 1...m \end{cases} \tag{5-20}$$

where the gradient of the volume function is defined in Equation (2-47) and the gradient of the stresses is obtained by the sensitivity analysis with the SGBEM.

The proposed procedure then seeks to solve the real problem by solving a sequence of subproblems of the type of (5-20). From a practical point of view, each subproblem seeks to approximate the real problem through a quadratic conic model which in turn must be solved in order to determine a step of the optimization. In general, this class of sequential optimization algorithms can be implemented using either *line search* or *trust-region* methods. The *line search* method requires a great number of evaluations of both the objective function and the constraints of the problem, thus, since each evaluation of the topology optimization problem requires a complete structural analysis, this method leads to an expensive computational alternative. On the other hand, the *trust-region* approach only requires one step of the structural analysis for each iteration, thus alleviating much of the computational effort involved in each optimization step. In addition, the *trust-region* method can be naturally coupled to subproblem (5-20) by simply introducing the following second-order cone constraint

$$\|\Delta\rho\| \leq r_k \tag{5-21}$$

where $r$ is the trust-region radius of the *k-th* iteration of the optimization.

The constraint (5-21) restrains the subproblem solution to the ball of radius $r$, thus defining the *trust-region* for the current iteration of the problem. Another key ingredient of the method is the choice of the radius for the *trust-region* in each iteration of the problem. In general, this choice is based on the agreement between the real problem and the quadratic conic model adopted in the approximation of each subproblem. In the present work, the model is based on the approximation of the von Mises stresses in each iteration of the real problem, thus the agreement defined as the following rate

$$\varepsilon_k := \left\| \frac{\sqrt{\sigma_j(\rho_0+\Delta\rho)^{\mathrm{T}}\mathbf{M}\,\sigma_j\,(\rho_0+\Delta\rho)} - \sqrt{\sigma_j(\rho_0)^{\mathrm{T}}\mathbf{M}\,\sigma_j\,(\rho_0)}}{\|\mathbf{y}_j\| - \sqrt{\sigma_j(\rho_0)^{\mathrm{T}}\mathbf{M}\,\sigma_j\,(\rho_0)}} \right\|_\infty \tag{5-22}$$

which measures the error between the real stresses and the approximation employed.

Based on this agreement the following simple algorithm, proposed in [62], may be employed here in order to choose the *trust-region* radius and also to accept or reject the step in each iteration of the optimization procedure.

**Algorithm 1** Trust-region

```
for k=0,1,2,... do
    Obtain $\Delta\rho_k$ by solving subproblem (??)
    Evaluate $\varepsilon_k$ as in (5-22)
    if $\varepsilon_{k+1}$ <0.25 then
        $r_{k+1} = 0.25 r_k$
    else
        if $\varepsilon_k > 0.75$ and $\|\Delta\rho_k\| = r_k$ then
            $r_{k+1} = 2r_k$
        else
            $r_{k+1} = r_k$
        end if
        $\rho_{k+1} = \rho_k + \Delta\rho_k$
    end if
end for
```

## 5.5 Plate with a hole example

To explore the efficiency and robustness of the proposed approach, the numerical solution of a well-known benchmark problem is investigated. Such problem consists in minimizing the stress concentrations by optimizing the shape around the hole of an infinite perforated plate. The initial hole is usually assumed to be circular and the plate subjected to biaxial loading. Figure 5.2 depicts the problem with its initial shape and boundary conditions.

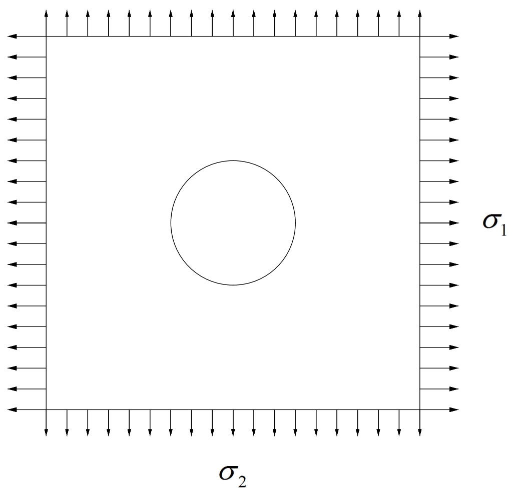


Figure 5.2: Infinite perforate plate under biaxial loading.

According to [63] the analytical solution for this setting is comprised of an ellipse shaped hole, for which the ratio between its major and minor axes is defined by $\frac{a}{b} = \frac{\sigma_1}{\sigma_2}$, surrounded by an uniformly distributed stress of intensity $\sigma_f = 1.5\,\sigma_1$.

In order to verify the proposed implementation, the following numerical example, illustrated in Figure 5.3, is investigated. Given the double symmetry of the problem, only a quarter of the plate is modeled.

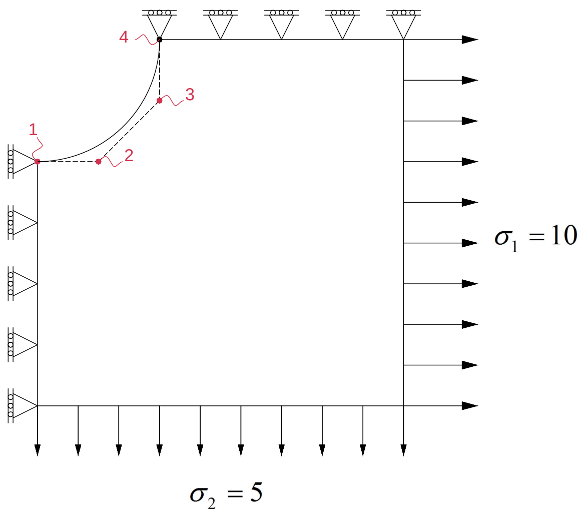


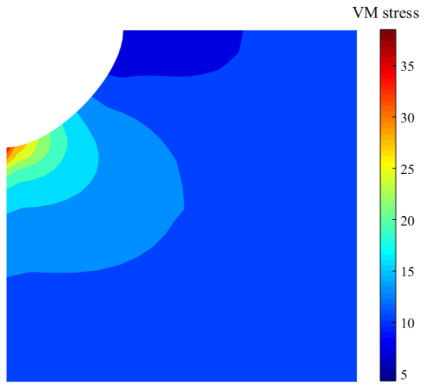
Figure 5.3: Numerical example of an infinite perforated plate.

In this initial setting, the circular hole is approximated by a cubic *Bézier* curve with control points 1, 2, 3 and 4. As for the optimization modeling, points 2 and 3 are free to move in either directions and no bounds are specified for the initial and end points of each *span*, *i.e.* points 2 and 3 are free to move anywhere. In addition, to ensure $\mathcal{C}^1$ continuity of the *Bézier* curves on the line of symmetry, points 1 and 4 are linked to the displacements of points 2 and 3 by suitable linear constraints.

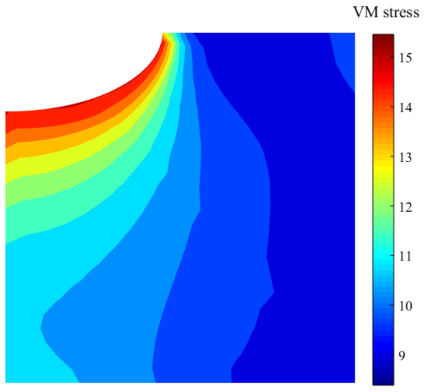
Figure 5.5 shows stress distributions for the initial and final shapes, while Figure 5.4 shows the von Mises stress around the optimized hole. Both results are compared with the analytical solution provided in [63].

The ratio between the principal axis of the final *Bézier* curve was approximately evaluated as 2.003, very close to the analytical one, i.e. 2. Also, the von Mises stress around the hole is approximately uniform, with maximum intensity of 15.5, sufficiently close to the analytical stress solution of 15. Figure 5.4 shows the initial and final von Mises stress along the optimized edge.

In order to assess the efficiency of the proposed approach, a comparison with other available methods is carried out. The first comparative test consists in solving the problem by means of the SQP algorithm according to the



5.4(a): Initial shape stress distribution.



5.4(b): Final shape stress distribution.

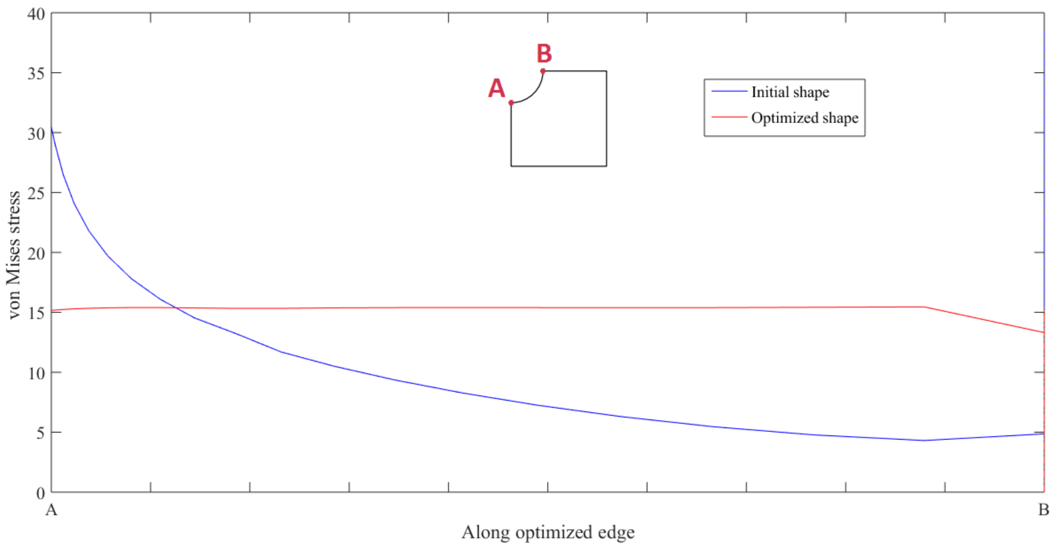


Figure 5.4: von Mises stress along optimized edge.

bound formulation. Following this approach, the stress and area constraints are handled as general non-linear constraints. The second alternative consists in employing the smooth maximum approach in order to obtain a differentiable objective function. This method is accomplished by taking the p-norm of the von Mises stress values in the discretization nodes around the hole. In addition, the area constraint is handled as a general non linear constraint and the optimization problem is carried out by the SQP algorithm. Table 5.1 shows the numerical comparative results related to the computational effort required by each one of the methods.

These results highlight the assets of the proposed approach over the alternative methods available in the literature. To begin with, the sequential SOCP approach required less iterations than the other methods. Secondly, due to the *trust-region* method, the number of structural analysis evaluations was

Table 5.1: Comparison between the alternative optimization formulations.

| Method | Iterations | Structural analysis evaluations | Total elapsed time |
|---|---|---|---|
| Sequential SOCP | 6 | 6 | 21.04 s |
| Bound formulation with SQP | 8 | 54 | 37,18 s |
| *Smooth maximum* with SQP | 21 | 135 | 96.36 s |

kept to a minimum, i.e. one per iteration. On the other hand, the other methods were carried out by a SQP solver which employs the line search algorithm, thus resulting in extra analysis evaluations for each step of the optimization. Finally, the total elapsed time for the sequential SOCP was remarkably lower, thus indicating that the proposed approach is particularly suitable for the minimization of stress concentrations.

# 6 SGBEM block partition approach

The implementation of the SOCP by the primal-dual interior points algorithm allows for an extremely efficient solution of subproblems (5-20). Furthermore, employing the procedure described in Chapter 2, the structure's volume and its gradient may be evaluated very efficiently, reducing to the simple evaluation of quadratic and linear functions of the design variables. On the other hand, the evaluation of stresses and their derivatives generally represents the most time consuming step in shape optimization problems. Although the SGBEM relieves the burden of the mesh generation step, the solution of the boundary value problem, required at each iteration of the optimization, is usually computationally expensive. Therefore, in order to reduce the computational effort associated with this step, a block matrix partition approach is proposed. This approach relies on the splitting of the boundary in two parts, one which remain fixed and the other which moves during the optimization procedure. The double integrals associated with the fixed boundary remain constant and therefore may be evaluated only once throughout the optimization. In addition, a suitable partition of the SGBEM equations is proposed with the goal of reducing the dimension of the systems which must be solved at each iteration of process.

## 6.1 The partition approach

The first step of this approach is to split the boundary as

$$\Gamma = \Gamma_u \cup \Gamma_t \cup \Gamma_f \tag{6-1}$$

where $\Gamma_u$ is the part of the boundary where the displacements are prescribed, $\Gamma_t$ is the part where the tractions are prescribed and $\Gamma_f$ is the part to be optimized.

In the present work, it is always assumed that both $\Gamma_u$ and $\Gamma_t$ are not allowed to move during the optimization process. It is also assumed that the $\Gamma_f$ part of the boundary is prescribed with null tractions, i.e. $\bar{t}_f = 0$ in $\Gamma_f$, and that the displacements in $\Gamma_u$ are fixed, i.e. $\bar{u}_u = 0$ in $\Gamma_u$. Given these assumptions and the splitting of the boundary as in Equation (6-1), the SGBEM leads to the following linear system of equations.

$$\begin{bmatrix} \mathbf{R}_{uu}^{uu} & -\mathbf{R}_{ut}^{ut} & -\mathbf{R}_{uf}^{ut} \\ (-\mathbf{R}_{ut}^{ut})^T & \mathbf{R}_{tt}^{tt} & \mathbf{R}_{tf}^{tt} \\ \left(-\mathbf{R}_{uf}^{ut}\right)^T & \left(\mathbf{R}_{tf}^{tt}\right)^T & \mathbf{R}_{ff}^{tt} \end{bmatrix} \begin{Bmatrix} \mathbf{t}_u \\ \mathbf{u}_t \\ \mathbf{u}_f \end{Bmatrix} = \begin{bmatrix} \mathbf{R}_{uu}^{ut} & -\mathbf{R}_{ut}^{uu} & -\mathbf{R}_{uf}^{uu} \\ -\mathbf{R}_{tu}^{tt} & \mathbf{R}_{tt}^{tu} & \mathbf{R}_{tf}^{tu} \\ -\mathbf{R}_{fu}^{tt} & \mathbf{R}_{ft}^{tu} & \mathbf{R}_{ff}^{tu} \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{u}}_u \\ \bar{\mathbf{t}}_t \\ \bar{\mathbf{t}}_f \end{Bmatrix} \tag{6-2}$$

which can be simplified to

$$\begin{bmatrix} \mathbf{R}_{uu}^{uu} & -\mathbf{R}_{ut}^{ut} & -\mathbf{R}_{uf}^{ut} \\ (-\mathbf{R}_{ut}^{ut})^T & \mathbf{R}_{tt}^{tt} & \mathbf{R}_{tf}^{tt} \\ \left(-\mathbf{R}_{uf}^{ut}\right)^T & \left(\mathbf{R}_{tf}^{tt}\right)^T & \mathbf{R}_{ff}^{tt} \end{bmatrix} \begin{Bmatrix} \mathbf{t}_u \\ \mathbf{u}_t \\ \mathbf{u}_f \end{Bmatrix} = \begin{Bmatrix} \mathbf{b}_u \\ \mathbf{b}_t \\ \mathbf{b}_f \end{Bmatrix} \tag{6-3}$$

where

$$\mathbf{b}_u = -\mathbf{R}_{ut}^{uu}\bar{\mathbf{t}}_t; \quad \mathbf{b}_t = \mathbf{R}_{tt}^{tu}\bar{\mathbf{t}}_t; \quad \mathbf{b}_f = \mathbf{R}_{ft}^{tu}\bar{\mathbf{t}}_t; \tag{6-4}$$

Due to the assumption that both $\Gamma_u$ and $\Gamma_t$ are not allowed to move, the double integrals involved in the case that both the source and the field element belongs to $\Gamma_u \cup \Gamma_t$ remain constant throughout the optimization process. Then it is straightforward to observe that matrices $\mathbf{R}_{uu}^{uu}$, $\mathbf{R}_{ut}^{ut}$, $\mathbf{R}_{tt}^{tt}$ and vectors $\bar{\mathbf{b}}_u$ and $\bar{\mathbf{b}}_t$ remain unaltered as the optimization progress.

Based on this simple observation, the following block partition of system (6-3) is proposed

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{Bmatrix} \tag{6-5}$$

where

$$\mathbf{A}_{11} = \begin{bmatrix} \mathbf{R}_{uu}^{uu} & -\mathbf{R}_{ut}^{ut} \\ (-\mathbf{R}_{ut}^{ut})^T & \mathbf{R}_{tt}^{tt} \end{bmatrix}; \quad \mathbf{A}_{12} = \begin{bmatrix} -\mathbf{R}_{uf}^{ut} \\ \mathbf{R}_{tf}^{tt} \end{bmatrix}; \quad \mathbf{A}_{22} = \mathbf{R}_{ff}^{tt}; \tag{6-6}$$

and

$$\begin{aligned} \mathbf{x}_1 &= \begin{bmatrix} \mathbf{t}_u \\ \mathbf{u}_t \end{bmatrix}; & \mathbf{b}_1 &= \begin{bmatrix} \mathbf{b}_u \\ \mathbf{b}_t \end{bmatrix}; \\ \mathbf{x}_2 &= \mathbf{u}_f & \mathbf{b}_2 &= \mathbf{b}_f; \end{aligned} \tag{6-7}$$

Both the block matrix $\mathbf{A}_{11}$ and the $\mathbf{b}_1$ vector may be evaluated only once, possibly in a preprocessing stage. The remaining quantities, i.e. $\mathbf{A}_{12}$, $\mathbf{A}_{22}$ and $\mathbf{b}_2$ must be updated for each iteration of the optimization procedure.

Based on this partition the system (6-5) may be solved as

$$\begin{aligned} \mathbf{x}_2 &= \left(\mathbf{A}_{22} - \mathbf{A}_{12}{}^T(\mathbf{A}_{11})^{-1}\mathbf{A}_{12}\right)^{-1}\left(\mathbf{b}_2 - \mathbf{A}_{12}{}^T(\mathbf{A}_{11})^{-1}\mathbf{b}_1\right) \\ \mathbf{x}_1 &= (\mathbf{A}_{11})^{-1}(\mathbf{b}_1 - \mathbf{A}_{12}\mathbf{x}_2) \end{aligned} \tag{6-8}$$

where the matrix

$$\left(\mathbf{A}_{22} - \mathbf{A}_{12}{}^T(\mathbf{A}_{11})^{-1}\mathbf{A}_{12}\right)$$

is called the Schur's complement of $\mathbf{A}_{11}$ in $\mathbf{A}$.

Before presenting an efficient algorithm for solving (6-8) a thorough analysis about the positive definiteness of the matrices involved in this system is presented. The following observations are based on the fact that the SGBEM system matrix is always positive definite and results from the following proposition

**Proposition 1** *Let $A$ be a positive definite matrix of the form*

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix} \tag{6-9}$$

*then (1) $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$ are positive definite. (2)*

$$\left(\mathbf{A}_{22} - {\mathbf{A}_{12}}^T (\mathbf{A}_{11})^{-1} \mathbf{A}_{12}\right)$$

*is positive definite.*

*Proof.* (1) Given that $\mathbf{A}$ is positive definite then

$$\mathbf{z}^T \mathbf{A}\, \mathbf{z} \geq 0\, \forall \mathbf{z} \in \mathbb{R}^n \equiv \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}^T \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ {\mathbf{A}_{12}}^T & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} \geq 0\, \forall \mathbf{z}_1 \in \mathbb{R}^{n_1}, \forall \mathbf{z}_2 \in \mathbb{R}^{n_2} \tag{6-10}$$

thus

$$\begin{bmatrix} \mathbf{z}_1 \\ 0 \end{bmatrix}^T \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ {\mathbf{A}_{12}}^T & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ 0 \end{bmatrix} \geq 0\, \forall \mathbf{z}_1 \in \mathbb{R}^{n_1} \Rightarrow {\mathbf{z}_1}^T \mathbf{A}_{11}\, \mathbf{z}_1 \geq 0\, \forall \mathbf{z}_1 \in \mathbb{R}^{n_1} \tag{6-11}$$

which readily shows that $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$ are positive definite. ■

*Proof.* (2) Given that $\mathbf{A}$ is block symmetric then it may be decomposed as (see [49])

$$\mathbf{A} = \mathbf{N}^T \mathbf{D} \mathbf{N} \tag{6-12}$$

where

$$\mathbf{N} = \begin{bmatrix} I & {\mathbf{A_{11}}}^{-1} \mathbf{A_{12}} \\ 0 & I \end{bmatrix}; \quad \mathbf{D} = \begin{bmatrix} \mathbf{A_{11}} & 0 \\ 0 & \mathbf{A_{22}} - {\mathbf{A_{12}}}^T {\mathbf{A_{11}}}^{-1} \mathbf{A_{12}} \end{bmatrix}; \tag{6-13}$$

Observing that $\mathbf{N}$ is always invertible with

$$\mathbf{N}^{-1} = \begin{bmatrix} \mathbf{I} & -{\mathbf{A_{11}}}^{-1} \mathbf{A_{12}} \\ 0 & \mathbf{I} \end{bmatrix} \tag{6-14}$$

then it readily follows that $\mathbf{N}$ is a full row rank matrix.

Therefore, it also follows from the positive definiteness of $\mathbf{A}$ that

$$\begin{aligned} \mathbf{z}^T \mathbf{A}\, \mathbf{z} \geqslant 0\, \forall \mathbf{z} \in \mathbb{R}^n \quad &\Rightarrow \mathbf{z}^T \mathbf{N}^T \mathbf{D} \mathbf{N}\, \mathbf{z} \geqslant 0\, \forall \mathbf{z} \in \mathbb{R}^n \\ &\Rightarrow \ \mathbf{y}^T \mathbf{D} \mathbf{y} \geqslant 0\, \forall \mathbf{y} \in \mathbb{R}^n, \quad \mathbf{y} = \mathbf{N}\mathbf{z} \end{aligned} \tag{6-15}$$

thus proving that $\mathbf{D}$ is positive definite.

Finally, using Proposition 1, it readily follows that $\left(\mathbf{A}_{22} - {\mathbf{A}_{12}}^T (\mathbf{A}_{11})^{-1} \mathbf{A}_{12}\right)$ is also positive definite. ■

These results show that solving system (6-5) by means of the scheme (6-8) is always possible. Furthermore, given that the matrices $\mathbf{A}_{11}$ and $\left(\mathbf{A}_{22} - {\mathbf{A}_{12}}^T (\mathbf{A}_{11})^{-1} \mathbf{A}_{12}\right)$ are positive definite, specialized procedures may be employed, thus leading to an efficient and numerically stable procedure. Based on these facts the following algorithm is proposed.

**Algorithm 2** Partitioned system solution algorithm

```
1. Perform the LU factorization of $\mathbf{A}_{11} = \mathbf{LU}$
2. Solve $\mathbf{c}_1 := \mathbf{L}^{-1}\mathbf{b}_1$ by back substitution
3. Solve $\mathbf{B}_{12} := {\mathbf{A}_{12}}^T\mathbf{U}^{-1}$ and $\mathbf{B}_{21} := \mathbf{L}^{-1}\mathbf{A}_{12}$ by multiple back substitution
4. Solve for $x_2 = (\mathbf{A}_{22} - \mathbf{B}_{12}\mathbf{B}_{21})^{-1}(\mathbf{b}_2 - \mathbf{B}_{12}\mathbf{c}_1)$
5. Solve $\mathbf{x}_1 = \mathbf{U}^{-1}(\mathbf{c}_1 - \mathbf{B}_{12}\mathbf{x}_2)$ by back substitution
```

It is worth noting that $\mathbf{A}_{11}$ and $\mathbf{b}_1$ remain constant, thus steps 1 and 2 may be executed only once throughout the optimization process. In addition, the back substitution in steps 2 3 and 5 requires less computational effort than a standard matrix-vector multiplication and therefore may be carried out extremely fast. At last, given that $(\mathbf{A}_{22} - \mathbf{B}_{12}\mathbf{B}_{21})^{-1}$ is positive definite, the system in step 4 may be solved by specialized solvers.

This block decomposition approach may also be applied in the sensitivity analysis stage in a straightforward manner. Given the splitting of the boundary, as defined in Equation (6-1), the discretized equations of the sensitivity analysis stage with SGBEM are given as

$$\begin{aligned}
&\begin{bmatrix} \mathbf{R}^{uu}_{uu} & -\mathbf{R}^{ut}_{ut} & -\mathbf{R}^{ut}_{uf} \\ (-\mathbf{R}^{ut}_{ut})^T & \mathbf{R}^{tt}_{tt} & \mathbf{R}^{tt}_{tf} \\ \left(-\mathbf{R}^{ut}_{uf}\right)^T & \left(\mathbf{R}^{tt}_{tf}\right)^T & \mathbf{R}^{tt}_{ff} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{t}}_u \\ \dot{\mathbf{u}}_t \\ \dot{\mathbf{u}}_f \end{bmatrix} + \begin{bmatrix} \dot{\mathbf{R}}^{uu}_{uu} & -\dot{\mathbf{R}}^{ut}_{ut} & -\dot{\mathbf{R}}^{ut}_{uf} \\ -\dot{\mathbf{R}}^{tu}_{tu} & \dot{\mathbf{R}}^{tt}_{tt} & \dot{\mathbf{R}}^{tt}_{tf} \\ -\dot{\mathbf{R}}^{tu}_{fu} & \dot{\mathbf{R}}^{tt}_{ft} & \dot{\mathbf{R}}^{tt}_{ff} \end{bmatrix} \begin{bmatrix} \mathbf{t}_u \\ \mathbf{u}_t \\ \mathbf{u}_f \end{bmatrix} = \\
&= \begin{bmatrix} \mathbf{R}^{ut}_{uu} & -\mathbf{R}^{uu}_{ut} & -\mathbf{R}^{uu}_{uf} \\ -\mathbf{R}^{tt}_{tu} & \mathbf{R}^{tu}_{tt} & \mathbf{R}^{tu}_{tf} \\ -\mathbf{R}^{tt}_{fu} & \mathbf{R}^{tu}_{ft} & \mathbf{R}^{tu}_{ff} \end{bmatrix} \begin{bmatrix} \dot{\bar{\mathbf{u}}}_u \\ \dot{\bar{\mathbf{t}}}_t \\ \dot{\bar{\mathbf{t}}}_f \end{bmatrix} + \begin{bmatrix} \dot{\mathbf{R}}^{ut}_{uu} & -\dot{\mathbf{R}}^{uu}_{ut} & -\dot{\mathbf{R}}^{uu}_{uf} \\ -\dot{\mathbf{R}}^{tt}_{tu} & \dot{\mathbf{R}}^{tu}_{tt} & \dot{\mathbf{R}}^{tu}_{tf} \\ -\dot{\mathbf{R}}^{tt}_{fu} & \dot{\mathbf{R}}^{tu}_{ft} & \dot{\mathbf{R}}^{tu}_{ff} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{u}}_u \\ \bar{\mathbf{t}}_t \\ \bar{\mathbf{t}}_f \end{bmatrix}
\end{aligned} \tag{6-16}$$

Observing that the boundary conditions are not altered by the geometry modifications, *i.e.*

$$\begin{bmatrix} \dot{\bar{\mathbf{u}}}_u \\ \dot{\bar{\mathbf{t}}}_t \\ \dot{\bar{\mathbf{t}}}_f \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{6-17}$$

and that the double integrals associated with the fixed boundary remain constant

$$\begin{bmatrix} \dot{\mathbf{R}}^{uu}_{uu} & -\dot{\mathbf{R}}^{ut}_{ut} \\ \left(-\dot{\mathbf{R}}^{ut}_{ut}\right)^T & \dot{\mathbf{R}}^{tt}_{tt} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; \quad \begin{bmatrix} \dot{\mathbf{R}}^{ut}_{uu} & -\dot{\mathbf{R}}^{uu}_{ut} \\ -\dot{\mathbf{R}}^{tt}_{tu} & \dot{\mathbf{R}}^{tu}_{tt} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; \tag{6-18}$$

the system (6-16) may be simplified as

$$\begin{bmatrix} \mathbf{R}^{uu}_{uu} & -\mathbf{R}^{ut}_{ut} & -\mathbf{R}^{ut}_{uf} \\ (-\mathbf{R}^{ut}_{ut})^T & \mathbf{R}^{tt}_{tt} & \mathbf{R}^{tt}_{tf} \\ \left(-\mathbf{R}^{ut}_{uf}\right)^T & \left(\mathbf{R}^{tt}_{tf}\right)^T & \mathbf{R}^{tt}_{ff} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{t}}_u \\ \dot{\mathbf{u}}_t \\ \dot{\mathbf{u}}_f \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{b}}_u \\ \dot{\mathbf{b}}_t \\ \dot{\mathbf{b}}_f \end{bmatrix} \tag{6-19}$$

where

$$\begin{aligned} \dot{\mathbf{b}}_u &= \dot{\mathbf{R}}^{ut}_{uf}\mathbf{u}_f \\ \dot{\mathbf{b}}_t &= -\dot{\mathbf{R}}^{tt}_{tf}\mathbf{u}_f \\ \dot{\mathbf{b}}_f &= \dot{\mathbf{R}}^{tu}_{ft}\bar{\mathbf{t}}_t + \left(\dot{\mathbf{R}}^{ut}_{uf}\right)^T\mathbf{t}_u - \left(\dot{\mathbf{R}}^{tt}_{tf}\right)^T\mathbf{u}_t - \dot{\mathbf{R}}^{tt}_{ff}\mathbf{u}_f \end{aligned} \tag{6-20}$$

Following the same idea, the block partition of the system (6-19) is given as

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}^T_{12} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{b}}_1 \\ \dot{\mathbf{b}}_2 \end{bmatrix} \tag{6-21}$$

where

$$\mathbf{A}_{11} = \begin{bmatrix} \mathbf{R}^{uu}_{uu} & -\mathbf{R}^{ut}_{ut} \\ (-\mathbf{R}^{ut}_{ut})^T & \mathbf{R}^{tt}_{tt} \end{bmatrix}; \quad \mathbf{A}_{12} = \begin{bmatrix} -\mathbf{R}^{ut}_{uf} \\ \mathbf{R}^{tt}_{tf} \end{bmatrix}; \quad \mathbf{A}_{22} = \mathbf{R}^{tt}_{ff}; \tag{6-22}$$

and

$$\begin{aligned} &\dot{\mathbf{x}}_1 = \begin{bmatrix} \dot{\mathbf{t}}_u \\ \dot{\mathbf{u}}_t \end{bmatrix}; \quad \dot{\mathbf{b}}_1 = \begin{bmatrix} \dot{\mathbf{R}}^{ut}_{uf}\mathbf{u}_f \\ -\dot{\mathbf{R}}^{tt}_{tf}\mathbf{u}_f \end{bmatrix}; \\ &\dot{\mathbf{x}}_2 = \dot{\mathbf{u}}_f \qquad \dot{\mathbf{b}}_2 = \dot{\mathbf{R}}^{tu}_{ft}\bar{\mathbf{t}}_t + \left(\dot{\mathbf{R}}^{ut}_{uf}\right)^T\mathbf{t}_u - \left(\dot{\mathbf{R}}^{tt}_{tf}\right)^T\mathbf{u}_t - \dot{\mathbf{R}}^{tt}_{ff}\mathbf{u}_f; \end{aligned} \tag{6-23}$$

Therefore, the system may be solved as

$$\begin{aligned} \dot{\mathbf{x}}_2 &= \left(\mathbf{A}_{22} - \mathbf{A}_{12}{}^T(\mathbf{A}_{11})^{-1}\mathbf{A}_{12}\right)^{-1}\left(\dot{\mathbf{b}}_2 - \mathbf{A}_{12}{}^T(\mathbf{A}_{11})^{-1}\dot{\mathbf{b}}_1\right) \\ \dot{\mathbf{x}}_1 &= (\mathbf{A}_{11})^{-1}\left(\dot{\mathbf{b}}_1 - \mathbf{A}_{12}\dot{\mathbf{x}}_2\right) \end{aligned} \tag{6-24}$$

Based on Algorithm 1, the following pseudo-code may be applied for the solution of (6-24).

**Algorithm 3** Partitioned system solution algorithm: sensitivity analysis

```
Perform the LU factorization of $\mathbf{A}_{11} = \mathbf{LU}$
for i=1,2,3... do
    Solve $\mathbf{B}_{12} := \mathbf{A}_{12}{}^T\mathbf{U}^{-1}$ and $\mathbf{B}_{21} := \mathbf{L}^{-1}\mathbf{A}_{12}$ by multiple back substitution
    Solve $\mathbf{c}_1 := \mathbf{L}^{-1}\mathbf{b}_1$ by back substitution
    Perform the LU factorization of $(\mathbf{A}_{22} - \mathbf{B}_{12}\mathbf{B}_{21}) = \mathbf{L}_i\mathbf{U}_i$
    for k=1,2,3... do
        Solve for $\mathbf{x}_2 = (\mathbf{L}_i\mathbf{U}_i)^{-1}(\mathbf{b}_2 - \mathbf{B}_{12}\mathbf{c}_1)$ by double back substitution
        Solve $\mathbf{x}_1 = \mathbf{U}^{-1}(\mathbf{c}_1 - \mathbf{B}_{12}\mathbf{x}_2)$ by back substitution
    end for
end for
```

## 6.2 Shouldered plate example

To show the performance gain due of the proposed approach, the shape optimization of a shoulder fillet is investigated. The problem consists in optimizing the fillet's shape of a shouldered plate subjected to axial loading, as depicted in Figure 6.1, in order to minimize the stress concentrations effects.

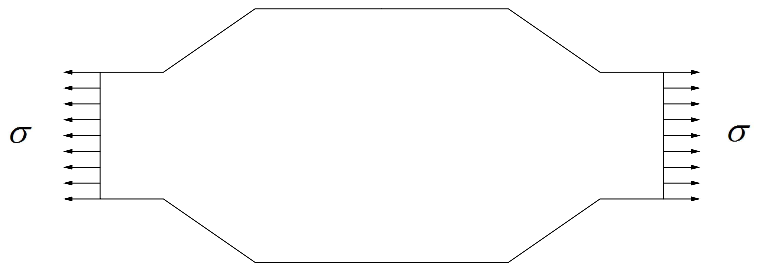


Figure 6.1: Shouldered plate under axial loading.

A numerical example of such problem, depicted in Figure 6 is investigated. Given the double symmetry of the problem, only a quarter is modeled.

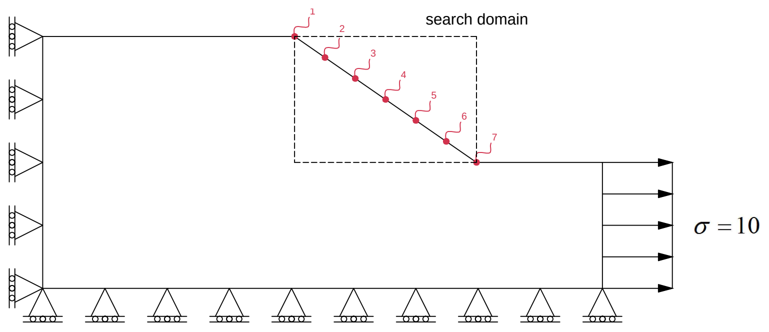
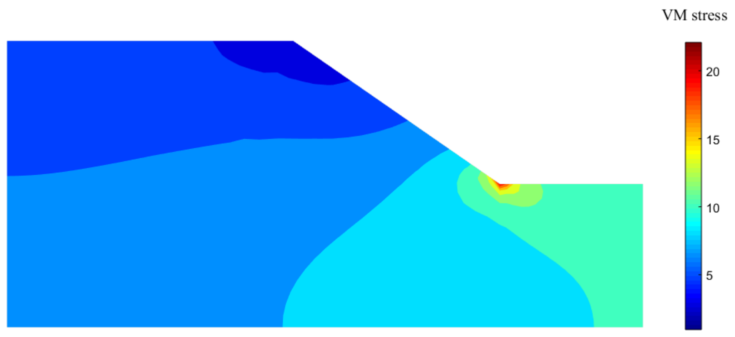
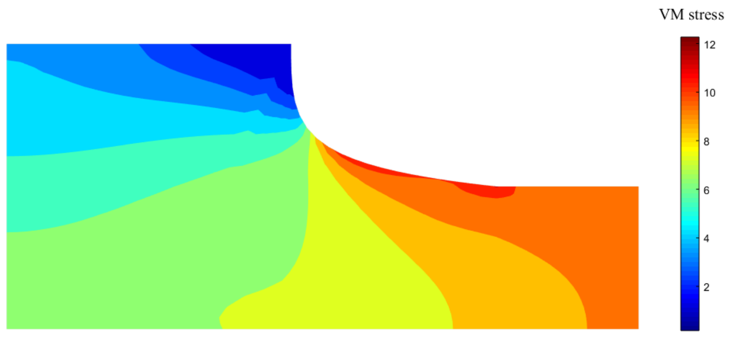


Figure 6.2: Shouldered plate under axial loading.

In order to seek for the optimal shape, the fillet edge is modeled by a cubic *B-spline* with control points 1 to 7. The optimization model is defined by letting points 2 to 6 to move inside the search domain box, depicted in Figure 6.2, in either directions. The initial and optimized shapes are shown in the stress plot of Figure 6.2, while Figure 6.3 depicts the von Mises stresses along the optimized fillet.



6.3(a): Initial shape stress distribution.



6.3(b): Final shape stress distribution.

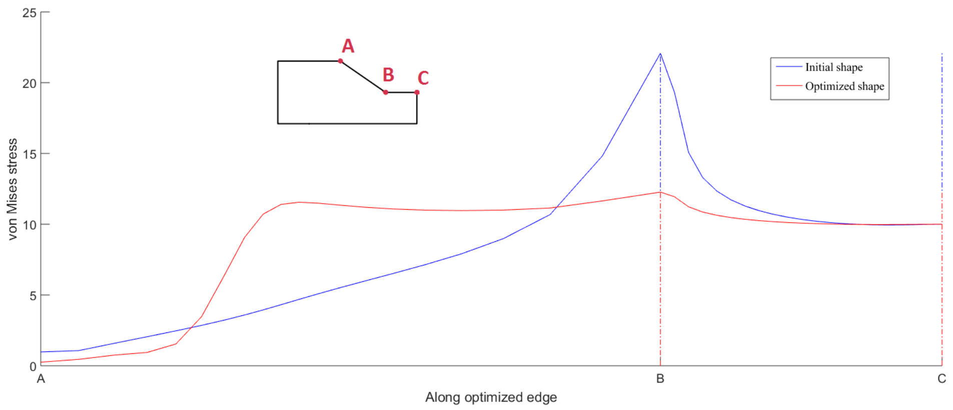


Figure 6.3: von Mises stress along optimized fillet.

In order to assess the efficiency boost of such approach, the shoulder fillet problem was initially carried out by using all the SGBEM equations for every step of the optimization, namely the *full* approach and then, afterwards, the proposed partition approach was employed. Figure 6.4 shows the time comparison between these two methods.

As expected, the partition approach reduces the computational time associated with the structural analysis of each step. In this particular example, partition approach managed to reduce by almost half of the total time,

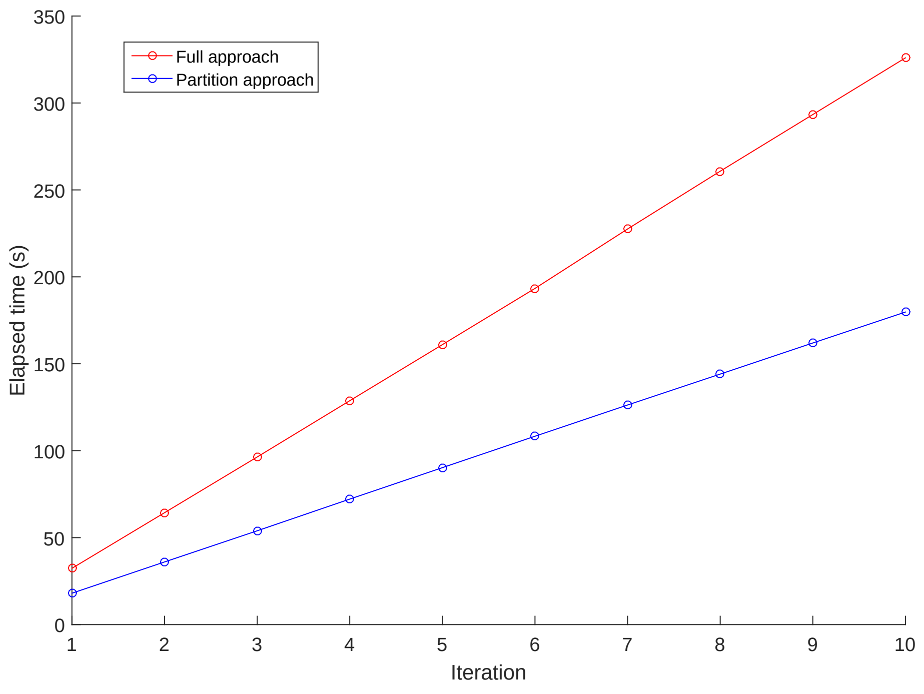


Figure 6.4: Time comparison: full vs partition approach.

specifically 44,85%, thus presenting itself as a promising tool for reducing the computation effort due to the strucural and sensitivity analysis with SGBEM.

# 7
# Conclusions and Future Work

This work has introduced the main aspects regarding the shape optimization for the minimization of stress concentrations. This subject is of great practical interest since the safety and life-span of structures may be significantly extended by avoiding such effects. Despite of the vast literature on this subject, the problem is far from being completely resolved. Actually, the minimization of stress concentrations is an on-going research field by both academy and industry. Hopefully, given the innovative features, this research may contribute to future works in this area.

## 7.1
## Conclusion remarks

The innovative sequential SOCP approach has been proven to be an extremely efficient alternative to the most commonly employed numerical optimization methods. Besides requiring less iterations, the approach has also been shown to be faster then the SQP approach, which, according to [62], is considered one of the most effective methods for solving nonlinearly constrained optimization problems. Furthermore, the trust-region method has been naturally incorporated in the optimization procedure, thus alleviating much of the burden of the structural analysis step.

Another main contribution of this research is due to the SGBEM block partition approach. Such concept has been shown to reduce the computational effort associated with the application of SGBEM as the analysis tool for the optimization. Moreover, the positive definiteness of the matrices involved in the approach, due to the SGBEM, resulted in a stable and efficient algorithm for solving the required system of equations in each step.

Finally, the quadratic function for evaluating the volume in terms of the design variables, besides providing economy in the computational aspect, is of great interest regarding the optimization formulation. Although applying the Green's theorem is not an innovative idea, to the best of the author's knowledge there is no work in the literature which proposes such quadratic function.

## 7.2 Future works

The following topics are suggested for further improving the proposed numerical implementation.

- Isogeometric analysis [64] is a modern computational approach which provides a direct integration between analysis and geometric modeling. In opposition to the traditional approach, in which the CAD data must be appropriately converted for analysis, the isogeometric formulation bypass this cumbersome step by incorporating the NURBS basis functions into the analysis. The application of such formulation within the framework of shape optimization is particularly attractive and has been proposed by several authors [65, 22, 21]. Isogeometric SGBEM has been recently investigated by [66], in the context of 2D boundary value problems for the Laplace equation, where the singular integrals are handled by means of suitable numerical schemes. However, to the best of the author's knowledge, no work involving elasticity problems have been published. The application of such formulation in the context of this work looks very promising and it is suggested for future works.
- Although the sequential SOCP formulation has been proven to be a reliable and efficient procedure towards the minimization of concentration effects, the approach is limited to second-order cone representable yield criteria. In order to extend such formulation for other resistance criteria, the investigation of a sequential semidefinite programming [67] procedure is suggested for future works.

# Bibliography

[1] BENDSOE, M. P.; SIGMUND, O.. **Topology Optimization**. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[2] HAFTKA, R. T.; GRANDHI, R. V.. **Structural shape optimization-A survey**. Computer Methods in Applied Mechanics and Engineering, 57(1):91–106, 1986.

[3] ANANTHASURESH, G. K.. **Topology and Size Optimization of Modular Ribs in Aircraft Wings**. In: 11TH WORLD CONGRESS ON STRUCTURAL AND MULTIDISCIPLINARY OPTIMISATION, p. 1–6, Sydney, 2015.

[4] HAGISHITA, T.; OHSAKI, M.. **Topology optimization of trusses by growing ground structure method**. Structural and Multidisciplinary Optimization, 37(4):377–393, 2009.

[5] ELSABBAGH, A.. **Size Optimization of Stiffeners in Bending Plates**. Mechanics of Advanced Materials and Structures, 20(9):764–773, 2013.

[6] BENDSOE, M. P.; KIKUCHI, N.. **Generating optimal topologies in structural design using a homogenisation method**. Computer Methods in Applied Mechanics and Engineering, 71(2):197–224, 1988.

[7] ZEGARD, T.; PAULINO, G. H.. **Bridging topology optimization and additive manufacturing**. Structural and Multidisciplinary Optimization, 53(1):175–192, 2016.

[8] HSU, Y.-L.; HSU, M.-S. ; CHEN, C.-T.. **Interpreting results from topology optimization using density contours**. Computers & Structures, 79(10):1049–1058, 2001.

[9] HSU, M.-H.; HSU, Y.-L.. **Interpreting three-dimensional structural topology optimization results**. Computers & Structures, 83(4):327–337, 2005.

[10] CAPPELLO, F.; MANCUSO, A.. **A genetic algorithm for combined topology and shape optimisations**. Computer-Aided Design, 35(8):761–769, 2003.

[11] YOUN, S.-K.; PARK, S.-H.. **A study on the shape extraction process in the structural topology optimization using homogenized material**. Computers & Structures, 62(3):527–538, 1997.

[12] BHAVIKATTI, S.; RAMAKRISHNAN, C.. **Optimum shape design of shoulder fillets in tension bars and T-heads**. International Journal of Mechanical Sciences, 21(1):29–39, 1979.

[13] FRANCAVILLA, A.; RAMAKRISHNAN, C. V. ; ZIENKIEWICZ, O. C.. **Optimization of shape to minimize stress concentration**. The Journal of Strain Analysis for Engineering Design, 10(2):63–70, 1975.

[14] MENG, L.; ZHANG, W.-H.; ZHU, J.-H. ; XIA, L.. **A biarc-based shape optimization approach to reduce stress concentration effects**. Acta Mechanica Sinica, 30(3):370–382, 2014.

[15] HILDING, D.; TORSTENFELT, B. ; KLARBRING, A.. **A computational methodology for shape optimization of structures in frictionless contact**. Computer Methods in Applied Mechanics and Engineering, 190(31):4043–4060, 2001.

[16] SAMAREH, J. A.. **A Survey of Shape Parameterization Techniques**. In: CEAS/AIAA/ICASE/NASA LANGLEY INTERNATIONAL FORUM ON AEROELASTICITY AND STRUCTURAL DYNAMICS, p. 333–343, Hampton, 1999.

[17] VAN MIEGROET, L.; DUYSINX, P.. **Stress concentration minimization of 2D filets using X-FEM and level set description**. Structural and Multidisciplinary Optimization, 33(4-5):425–438, 2007.

[18] BREBBIA, C.; DOMINQUEZ, J.. **Boundary elements: an introductory course**, 1992.

[19] CANELAS, A.; HERSKOVITS, J. ; TELLES, J.. **Shape optimization using the boundary element method and a SAND interior point algorithm for constrained optimization**. Computers & Structures, 86:1517–1526, 2008.

[20] CERROLAZA, M.; ANNICCHIARICO, W. ; MARTINEZ, M.. **Optimization of 2D boundary element models using $\beta$-splines and genetic**

**algorithms**. Engineering Analysis with Boundary Elements, 24(5):427–440, 2000.

[21] LI, K.; QIAN, X.. **Isogeometric analysis and shape optimization via boundary integral**. Computer-Aided Design, 43(11):1427–1437, 2011.

[22] LIAN, H.; KERFRIDEN, P. ; BORDAS, S.. **Shape optimization directly from CAD: An isogeometric boundary element approach using T-splines**. PhD thesis, Cardiff University, 2016.

[23] MARCZAK, R. J.. **Optimization of elastic structures using boundary elements and a topological-shape sensitivity formulation**. Latin American Journal of Solids and Structures, 5(2):99–117, 2008.

[24] ARORA, J. S.; WANG, Q.. **Review of formulations for structural and mechanical system optimization**. Structural and Multidisciplinary Optimization, 30(4):251–272, 2005.

[25] WILCZYNSKI, B.. **Shape optimization of elastic 2D bodies by the Fictitious Stress Method**. Transactions on Modelling and Simulation, 12(1996):301–310, 1993.

[26] CARLOS EDUARDO KUBRUSLY DA, S.. **Otimização de Forma de Modelos Bidimensionais de Elementos Finitos com Comportamento Elasto-Plástico.** PhD thesis, PUC-Rio, 2000.

[27] PARENTE JUNIOR, E.; VAZ, L. E. ; SILVA, R. R. E.. **Análise de sensibilidade e otimização de forma de estruturas geometricamente não-lineares.** PhD thesis, PUC-Rio, 2000.

[28] SOUSA JUNIOR, J. B. M. D.; VAZ, L. E. ; HINTON, E. E.. **Auto-adaptação e otimização de forma em cascas.** PhD thesis, PUC-Rio, 2000.

[29] KARMARKAR, N.. **A new polynomial-time algorithm for linear programming**. In: PROCEEDINGS OF THE SIXTEENTH ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING - STOC '84, p. 302–311, New York, New York, USA, 1984. ACM Press.

[30] MCKEOWN, J.. **Optimal composite structures by deflection-variable programming**. Computer Methods in Applied Mechanics and Engineering, 12(2):155–179, oct 1977.

[31] PARVIZIAN, J.; FENNER, R.. **Shape optimisation by the boundary element method: a comparison between mathematical programming and normal movement approaches**. Engineering Analysis with Boundary Elements, 19:137–145, 1997.

[32] CHAUDOUET-MIRANDA, A.; EL YAFI, F.. **Boundary Element Method Applied to 3D Optimum Design**. In: Cruse, T. A., editor, ADVANCED BOUNDARY ELEMENT METHODS: PROCEEDINGS OF THE IUTAM SYMPOSIUM, SAN ANTONIO, TEXAS, APRIL 13–16, 1987, p. 101–108. Springer Berlin Heidelberg, Berlin, Heidelberg, 1988.

[33] ZIENKIEWICZ, O.; CAMPBELL, J.. **Shape optimization and sequential linear programming**. In: OPTIMUM STRUCTURAL DESIGN : THEORY AND APPLICATIONS, p. 109–126. London ; New York : Wiley, 1973.

[34] WANG, M. Y.; WANG, X. ; GUO, D.. **A level set method for structural topology optimization**. Computer Methods in Applied Mechanics and Engineering, 192(1-2):227–246, 2003.

[35] S.FLOATER, M.. **Splines and B-Splines**. Technical report, University of Oslo, Oslo, 2007.

[36] PRAUTZSCH, H.; BOEHM, W. ; PALUSZNY, M.. **Bézier and B-Spline Techniques**. Mathematics and Visualization. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[37] SONMEZ, F. O.. **Optimal shape design of shoulder fillets for flat and round bars under various loadings**. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 223(8):1741–1754, 2009.

[38] SEO, Y. D.; KIM, H. J. ; YOUN, S. K.. **Shape optimization and its extension to topological design based on isogeometric analysis**. International Journal of Solids and Structures, 47(11-12):1618–1640, 2010.

[39] BECKER, A. A.. **The Boundary Element Methods in Engineering**, 1992.

[40] BONNET, M.; MAIER, G. ; POLIZZOTTO, C.. **Symmetric Galerkin Boundary Element Methods**. Applied Mechanics Reviews, 51(11):669, 1998.

[41] FRANGI, A.; NOVATI, G.. **Symmetric BE method in two-dimensional elasticity: evaluation of double integrals for curved elements**. Computational Mechanics, 19:58–68, 1996.

[42] CARINI, A.; DILIGENTI, M.; MARANESI, P. ; ZANELLA, M.. **Analytical integrations for two-dimensional elastic analysis by the symmetric Galerkin boundary element method**. Computational Mechanics, 23(4):308–323, 1999.

[43] BALAKRISHNA, C.; GRAY, L. J. ; KANE, J. H.. **Efficient analytical integration of symmetric Galerkin boundary integrals over curved elements: elasticity formulation**. Computer Methods in Applied Mechanics and Engineering, 111(1):335–355, 1994.

[44] SUTRADHAR, A.; PAULINO, G. H. ; GRAY, L. J.. **Symmetric galerkin boundary element method**. Springer-Verlag, Berlin, 1st ed edition, 2008.

[45] GRAY, L. J.. **Evaluation of singular and hypersingular Galerkin integrals: direct limits and symbolic computation**. Singular Integrals in Boundary Element Methods, Comp. Mech. Publ., Southampton, p. 45–84, 1998.

[46] PARREIRA, P.; GUIGGIANI, M.. **On the implementation of the galerkin approach in the boundary element method**. Computers and Structures, 33(1):269–279, 1989.

[47] KIELHORN, L.. **A Time-Domain Symmetric Galerkin BEM for Viscoelastodynamics**. Monographic series TU Graz : Computation in engineering and science ; 5. Verl. der Techn. Univ. Graz, Graz, 2009.

[48] TROUTMAN, J. L.. **Variational Calculus and Optimal Control**. Undergraduate Texts in Mathematics. Springer New York, New York, NY, 1996.

[49] BOYD, S.; VANDENBERGHE, L.. **Convex Optimization Theory**, volumen 25. Cambridge University Press 2004, 2010.

[50] BEN-TAL, A.; NEMIROVSKII, A.. **An Interior Point Algorithm for Truss Topology Design**, p. 55–69. Springer Netherlands, Dordrecht, 1993.

[51] LOBO, M. S.; VANDENBERGHE, L.; BOYD, S. ; LEBRET, H.. **Applications of second-order cone programming**. Linear Algebra and its Applications, 284(1-3):193–228, 1998.

[52] LEBRET, H.; BOYD, S.. **Antenna array pattern synthesis via convex optimization**. IEEE Transactions on Signal Processing, 45(3):526–532, 1997.

[53] HSIEH, H.; BALARIN, F.; LAVAGNO, L. ; VINCENTELLI, A. S.. **Efficient methods for embedded system design space exploration**. Proceedings of the 37th Annual Design Automation Conference, p. 607–612, 2000.

[54] MAKRODIMOPOULOS, A.; MARTIN, C.. **Limit analysis using large-scale socp optimization**. In: PROC. 13TH NAT. CONF. OF UK ASSOCIATION FOR COMPUTATIONAL MECHANICS IN ENGINEERING, SHEFFIELD, p. 21–24, 2005.

[55] ANDERSEN, E. D.; ROOS, C. ; TERLAKY, T.. **On implementing a primal-dual interior-point method for conic quadratic optimization**. Mathematical Programming, Series B, 95(2):249–277, 2003.

[56] NESTEROV, Y. E.; TODD, M. J.. **Self-Scaled Barriers and Interior-Point Methods for Convex Programming**. Mathematics of Operations Research, 22(1):1–42, 1997.

[57] BISBOS, C. D.; PARDALOS, P. M.. **Second-order cone and semidefinite representations of material failure criteria**. Journal of Optimization Theory and Applications, 134(2):275–301, 2007.

[58] CARBONARI, R.; MUÑOZ-ROJAS, P.; ANDRADE, E.; PAULINO, G.; NISHIMOTO, K. ; SILVA, E.. **Design of pressure vessels using shape optimization: An integrated approach**. International Journal of Pressure Vessels and Piping, 88(5-7):198–212, 2011.

[59] CHANGWEN, X.; MINGHUA, Y.. **Shape optimization of structures to minimize stress concentration**. Computers & Structures, 36(3):491–497, 1990.

[60] SONMEZ, F. O.. **Shape optimization of 2D structures using simulated annealing**. Computer Methods in Applied Mechanics and Engineering, 196(35-36):3279–3299, 2007.

[61] APS, M.. **The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).**, 2015.

[62] NOCEDAL, J.; WRIGHT, S. J.. **Numerical Optimization**. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.

[63] KRISTENSEN, E. S.; MADSEN, N. F.. **On the optimum shape of fillets in plates subjected to multiple in-plane loading cases**. International Journal for Numerical Methods in Engineering, 10(5):1007–1019, 1976.

[64] COTTRELL, J. A.; HUGHES, T. J. R. ; BAZILEVS, Y.. **Isogeometric Analysis: Toward Integration of CAD and FEA**. Isogeometric Analysis: Toward Integration of CAD and FEA, p. 1–335, 2009.

[65] QIAN, X.. **Full analytical sensitivities in NURBS based isogeometric shape optimization**. Computer Methods in Applied Mechanics and Engineering, 199(29-32):2059–2071, 2010.

[66] AIMI, A.; DILIGENTI, M.; SAMPOLI, M. L. ; SESTINI, A.. **Isogemetric analysis and symmetric Galerkin BEM: A 2D numerical study**. Applied Mathematics and Computation, 272:173–186, 2016.

[67] VANDENBERGHE, L.; BOYD, S.. **Semidefinite Programming**. SIAM Review, 38(1):49–95, 1996.