1 2 9 0

UNIVERSIDADE Đ COIMBRA

Rafael Oliveira Henriques

Mathematical model to reconstruct the mechanical properties of an elastic medium

VOLUME 1

Dissertação no âmbito do Mestrado em Matemática, Ramo Estatística, Otimização e Matemática Financeira orientado pelo Professor Doutor José Luis Esteves dos Santos e pela Professora Doutora Sílvia Alexandra Alves Barbeiro.

Julho de 2020

Mathematical model to reconstruct the mechanical properties of an elastic medium

Rafael Oliveira Henriques

1 2 9 0

UNIVERSIDADE Đ
COIMBRA

Master in Mathematics
Mestrado em Matemática

MSc Dissertation | Dissertação de Mestrado

Julho 2020

Acknowledgements

I would like to thank my two advisors Professor Sílvia Alexandra Alves Barbeiro and Professor José Luis Esteves dos Santos, for their collaboration in this work.

I also want to thank the ElastoOCT project team, in particular to Professor Miguel Morgado for the suggestion of the real application for this work.

This work was developed under the ElastoOCT project, "Optical Coherence Elastography for imaging of the mechanical properties of the retina" PTDC/EMD-EMD/32162/2017.

Este trabalho é financiado por Fundos FEDER através do Programa Operacional Factores de Competitividade – COMPETE e por Fundos Nacionais através da FCT – Fundação para a Ciência e a Tecnologia no âmbito do projeto PTDC/EMD-EMD/32162/2017;

Cofinanciado por:

COMPETE 2020

PORTUGAL 2020

UNIÃO EUROPEIA
Fundo Europeu
de Desenvolvimento Regional

FCT Fundação para a Ciência e a Tecnologia

Abstract

In this work we investigate a mathematical model to reconstruct the mechanical properties of an elastic medium. To this end, we develop of a mathematical model for the mechanical deformation assuming that the parameters that define the mechanical properties of the medium are known. The model is based on time-harmonic equations of linear elasticity. The numerical solution is obtained using a finite element discretization in a three-dimentional domain. The performance of the method is illustrated with numerical examples. The mathematical model for solving this direct problem is the computational basis to address the inverse problem which consists of determining the set of parameters that characterize the mechanical properties of the medium knowing the displacement fields for a given excitation. We consider different optimization methods to solve the inverse problem and discuss their performance. We report several computational results which illustrate their behavior in terms of accuracy and efficiency.

The study reported in this thesis was developed in the scope of a broader project with the objective of developing a new imaging technique that allows to map in vivo the mechanical properties of the retina by Optical Coherence Elastography, which is an elastography technique based on Optical Coherence Tomography. The final goal is the discovery of biomarkers based on the mechanical properties of the retina that allow the early detection, before clinical manifestations, of neurodegenerative processes.

Resumo

Neste trabalho investigamos o modelo matemático para a reconstrução das propriedades mecânicas de um meio elástico. Com esse objectivo, nós desenvolvemos o modelo matemático para as deformações mecânicas assumindo que os parâmetros que definem as propriedades mecânicas de o meio são conhecidos. O modelo é baseado em equações de elasticidade linear em tempo harmónico. A solução numérica é obtida usando a discretização de elementos finitos em um domínio tridimensional. O desempenho do método é ilustrado com exemplos numéricos. O modelo matemático para resolver este problema direto é a base computacional para abordar o problema inverso que consiste em determinar o conjunto de parâmetros que caracterizam as propriedades mecânicas de um meio conhecendo o campo de deslocamentos para uma dada excitação. Nós consideramos diferentes métodos de otimização para resolver o problema inverso e discutimos o seu desempenho. Nós exibimos vários resultados computacionais que ilustram seu comportamento em termos de precisão e eficiência.

O estudo descrito nesta tese foi desenvolvido no âmbito de um projecto mais amplo com o objectivo do desenvolvimento de uma nova técnica de imagiologia que permite mapear in vivo as propriedades mecânicas da retina por Elastografia de Coerência Óptica, que é uma técnica de elastografia baseada na Tomografia de Coerência Óptica. O objectivo final é a descoberta de biomarcadores baseados nas propriedades mecânicas da retina que permitam a deteção precoce, anterior às manifestações clínicas, de processos neurodegenerativos.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

One of the major challenges that currently arises in the area of health is in prevention. It is very important to identify early stages of diseases while they still are on the asymptomatic phase allowing to mitigate the undesirable consequences of their progression. Neurodegenerative diseases, which affect millions of people worldwide, start to develop 10-20 years before their clinical manifestations [1]. The late stages are characterized by massive neuronal death, leading to an extremely unfavourable prognosis for treatment, making the early diagnosis is of uppermost importance.

This thesis arises in the context of a real application associated with the investigation of neurodegenerative processes in which there is a pressing need for techniques with enough sensitivity to detect early signs of neurodegeneration and to define biomarkers of neurodegenerative diseases. The retina is the visible part of the central nervous system and has been explored for signs of neurodegeneration. Moreover, findings of the scientific community support the idea that imaging the mechanical properties of the retina can detect changes before volumetric changes or neuronal losses become detectable.

Optical coherence elastography (OCE) is an emerging biomedical imaging technique based in the optical coherence tomography (OCT) imaging modality to form pictures of biological tissue and map its biomechanical properties. OCE combines mechanical excitation with OCT for measuring the corresponding elastic displacement [2–5]. Applications of this technique vary from skin to the retina, being the latter the one of most interest to this work. An acoustic excitation system can be used for inducing the mechanical load to the retina. In this case, an ultrasound source is coupled with an OCT device.

The work in this thesis was developed in the framework of a wider project, which gathers a multidisciplinary team, whose objective is to develop an OCE technique for measuring in vivo the mechanical properties of the retina on animal models, with the purpose of detecting early signs of neurodegeneration [6].

One of the tasks of the project is the development of a numerical model for the mechanical deformation of the retina induced by the propagation of acoustic waves, given the parameters that define the mechanical properties of the medium. This mathematical model will be used to numerically simulate the displacement field within the retina, given a known excitation, and considering different sets of values for the parameters of the model. In this thesis, we start by considering this direct problem. We propose a model based on time-harmonic equations of linear elasticity. To compute the numerical solution of this model we use a finite element method in a three-dimensional domain. We discretize the computational domain with tetrahedral elements and we derive the numerical solution using piecewise linear basis functions.

Another task of the project is to investigate a process for obtaining the mechanical properties of the retina given the displacement field, that is, to solve the inverse problem of elastography. In our approach, we formulate the inverse problem as an optimization program using mathematical model for solving the direct problem as the computational basis. The inverse problem consists of determining the set of parameters that characterize the mechanical properties of the medium knowing the displacement field for a given excitation. In practice, it is intended to infer the parameters that characterize the mechanical properties so that the difference between simulated displacements obtained with the mathematical model for the direct problem with those parameters and the data are minimized.

In the direct and the inverse problems, the retina is treated as a material with linear isotropic mechanical behavior, purely elastic.

The thesis is organized as follows. In Chapter 2 we describe the mathematical model for the direct problem. We present the deduction of the linear elasticity model which is a system of partial differential equations together with boundary conditions. We deduce its weak formulation and discuss results of existence and uniqueness of solution. In Chapter 3 we deduce in detail the linear finite element method for the problem and display some simulation results, in two different geometries, in order to illustrate the performance of this numerical method. We present an estimation of the convergence order based on simulation results. In Chapter 4 we investigate the inverse problem in which we intend to infer the mechanical properties of the medium knowing the mechanical deformations. We start by presenting the optimization problem that we need to solve and analyze its properties. Then we consider different optimization methods. We summarize the key ideas of each of them and present the corresponding implementation of the algorithms. Finally, we present several computational results and discuss the performance of the methods. We show the results of our approach using fabricated data which is obtained using the numerical solution of the direct problem. We include experiments with noise free data and noisy data. Introducing noise in the fabricated data we intend to mimic the application scenario where we plan, in future work, to use real data from the retina.

Chapter 2

Elasticity equation

In this chapter we derive the linear elasticity equation and the respective boundary conditions. Next we determine the weak formulation of this equation and we address the issue of existence and uniqueness of solution.

Let's start by introducing some notation needed to write the linear elasticity equation and the boundary conditions. Let p be a scalar function, $\underset{\sim}{v} = (v_i)_{1\leq i\leq 3}$ a vector function and $\underset{\approx}{a} = (a_{ij})_{1\leq i,j\leq 3}$ a matrix of functions of three variables. We will use the following notation for v, $\underset{\sim}{v}$ and $\underset{\approx}{a}$: given a space D, $v \in D$, $\underset{\sim}{v} \in \underset{\sim}{D} = D^3$ and $\underset{\approx}{v} \in \underset{\approx}{D} = D^{3\times 3}$.

We define $\underset{\sim}{\operatorname{grad}}\, p = \left(\frac{\partial p}{\partial x_1}\, \frac{\partial p}{\partial x_2}\, \frac{\partial p}{\partial x_3}\right)^T$, $\operatorname{div}\underset{\sim}{v} = \sum_{i=1}^{3} \frac{\partial v_i}{\partial x_i}$,

$$\underset{\approx}{\operatorname{grad}}\underset{\sim}{v} = \begin{pmatrix} \frac{\partial v_1}{\partial x_1} & \frac{\partial v_1}{\partial x_2} & \frac{\partial v_1}{\partial x_3} \\ \frac{\partial v_2}{\partial x_1} & \frac{\partial v_2}{\partial x_2} & \frac{\partial v_2}{\partial x_3} \\ \frac{\partial v_3}{\partial x_1} & \frac{\partial v_3}{\partial x_2} & \frac{\partial v_3}{\partial x_3} \end{pmatrix},\ \underset{\sim}{\operatorname{div}}\underset{\approx}{a} = \begin{pmatrix} \sum_{i=1}^{3} \frac{\partial a_{1i}}{\partial x_i} \\ \sum_{i=1}^{3} \frac{\partial a_{2i}}{\partial x_i} \\ \sum_{i=1}^{3} \frac{\partial a_{3i}}{\partial x_i} \end{pmatrix},\ \triangle\underset{\sim}{v} = \begin{pmatrix} \sum_{i=1}^{3} \frac{\partial^2 v_1}{\partial x_i^2} \\ \sum_{i=1}^{3} \frac{\partial^2 v_2}{\partial x_i^2} \\ \sum_{i=1}^{3} \frac{\partial^2 v_3}{\partial x_i^2} \end{pmatrix},\ \mathrm{I}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and yet

$$\underset{\approx}{\varepsilon}(\underset{\sim}{v}) = \frac{1}{2}\left(\underset{\approx}{\operatorname{grad}}\underset{\sim}{v} + (\underset{\approx}{\operatorname{grad}}\underset{\sim}{v})^T\right).$$

2.1 Deduction of the linear elasticity equation

This section is dedicated to deduce the equation of linear elasticity from the knowledge of physic laws, more specifically by the law of conservation of the momentum.

Let us consider an isotropic elastic material which occupies a bounded region $\Omega \subseteq \mathbb{R}^3$, with boundary $\partial\Omega$, subject a force acting on it. The aim is to characterize the field of induced displacements $\underset{\sim}{u}(x,t)$ with $x = (x_1, x_2, x_3) \in \Omega$ and $t \in \mathbb{R}_0^+$ in the form of a set of equations for the displacements.

We will denote by $\frac{\partial \underset{\sim}{u}}{\partial t}(x,t)$ the velocity field of such displacements and by ρ the material density. Let $P(t)$ be the momentum of the system that corresponds to

$$P(t) = \int_\Omega \rho \frac{\partial \underset{\sim}{u}}{\partial t}(x,t)\, dx.$$

By laws of physics it is known that the variation of the momentum of the system is equal to the resulting forces that are acting on it [7]. Assuming sufficient regularity, we determine the variation of the momentum that is given by

$$P'(t) = \int_\Omega \rho \frac{\partial^2 \underset{\sim}{u}}{\partial t^2}(x,t)\,dx.$$

Concerning the forces acting on Ω, we have the surface force, which interacts at the boundary of the set, and a force that acts on the volume of the body [8], i.e.,

$$\int_{\partial\Omega} \underset{\sim}{t}\,ds \text{ and } \int_\Omega \underset{\sim}{f}(x)\,dx.$$

respectively. The first integral is restricted to the boundary of the set Ω and $\underset{\sim}{t}$ is called the traction vector. By Cauchy's stress theorem, $\underset{\sim}{t}$ is the same as $\underset{\approx}{\sigma}(\underset{\sim}{u}(x,t))\underset{\sim}{n}$ [9] where $\underset{\approx}{\sigma}(\underset{\sim}{u}(x,t))$ is the stress tensor and it is given by [10]

$$\underset{\approx}{\sigma}(\underset{\sim}{u}(x,t)) = 2\mu\underset{\approx}{\varepsilon}(\underset{\sim}{u}(x,t)) + \lambda tr(\underset{\approx}{\varepsilon}(\underset{\sim}{u}(x,t)))\mathrm{I}_3$$

and $\underset{\sim}{n} = (n_1, n_2, n_3)$ is the unit outward normal to Ω. Summing the two forces results the following

$$\int_\Omega \rho \frac{\partial^2 \underset{\sim}{u}}{\partial t^2}(x,t)\,dx = \int_{\partial\Omega} \underset{\approx}{\sigma}(\underset{\sim}{u}(x,t))\underset{\sim}{n}\,ds + \int_\Omega \underset{\sim}{f}(x)\,dx. \tag{2.1}$$

Applying the divergence theorem to (2.1), we obtain

$$\int_\Omega \rho \frac{\partial^2 \underset{\sim}{u}}{\partial t^2}(x,t) - \mathrm{div}(\underset{\approx}{\sigma}(\underset{\sim}{u}(x,t))) - \underset{\sim}{f}(x)\,dx = 0.$$

The arbitrariness of the set Ω in terms of integration and the continuity of the integrand function allows to obtain the equality [11, Proposition 1.39]

$$\rho \frac{\partial^2 \underset{\sim}{u}}{\partial t^2}(x,t) = \underset{\sim}{\mathrm{div}}\underset{\approx}{\sigma}(\underset{\sim}{u}(x,t)) + \underset{\sim}{f}(x).$$

Assuming that $\underset{\sim}{u} \in C^2$, by the Schwarz's theorem, we have [12]

$$\frac{\partial}{\partial x_i}\left(\frac{\partial u_k}{\partial x_j}\right) = \frac{\partial}{\partial x_j}\left(\frac{\partial u_k}{\partial x_i}\right), i, j, k \in \{1,2,3\} \tag{2.2}$$

and using the definition of $\underset{\sim}{\mathrm{div}}\underset{\approx}{\sigma}(\underset{\sim}{u}(x,t))$ leads the linear elasticity equation for $\underset{\sim}{u}(x,t)$ that is [13, 14]

$$\rho \frac{\partial^2 \underset{\sim}{u}}{\partial t^2}(x,t) = \mu \triangle \underset{\sim}{u}(x,t) + (\lambda + \mu)\,\underset{\sim}{\mathrm{grad}}\left(\mathrm{div}\underset{\sim}{u}(x,t)\right) + \underset{\sim}{f}(x),\ (x,t) \in \Omega \times \mathbb{R}_0^+, \tag{2.3}$$

with the Lamé constants, μ and λ, given, respectively, by

$$\mu = \frac{E}{2(1+\upsilon)} \text{ and } \lambda = \frac{\upsilon E}{(1+\upsilon)(1-2\upsilon)}, \tag{2.4}$$

where E is the Young's Modulus and υ is the Poisson's ratio.

Let's assume that $(\mu,\lambda) \in [\mu_1,\mu_2] \times]0,\infty[$ where $0 < \mu_1 < \mu_2$. Consider that the applied impulse allows to admit that the displacement field has a harmonic time shape [14], i.e.,

$$\underset{\sim}{u}(x,t) = \Re\left(\underset{\sim}{u}(x)\, e^{i\omega t}\right), \tag{2.5}$$

where $\Re$ is the real part of a complex and ω is the angular frequency. By Euler formula we have $e^{i\omega t} = \cos(\omega t) + i\,\mathrm{sen}(\omega t)$, so (2.5) can be written in the form

$$\underset{\sim}{u}(x,t) = \Re\left[\underset{\sim}{u}(x)\left(\cos(\omega t) + i\,\mathrm{sen}(\omega t)\right)\right] = \underset{\sim}{u}(x)\cos(\omega t). \tag{2.6}$$

Next, we present each term of (2.3) considering the transformation (2.6):

$$\frac{\partial \underset{\sim}{u}}{\partial t}(x,t) = -\omega\,\mathrm{sen}(\omega t)\,\underset{\sim}{u}(x), \qquad \frac{\partial^2 \underset{\sim}{u}}{\partial t^2}(x,t) = -\omega^2\cos(\omega t)\,\underset{\sim}{u}(x),$$

$$\triangle \underset{\sim}{u}(x,t) = \triangle \underset{\sim}{u}(x)\cos(\omega t), \quad \underset{\sim}{\mathrm{grad}}\left(\mathrm{div}\underset{\sim}{u}(x,t)\right) = \underset{\sim}{\mathrm{grad}}\left(\mathrm{div}\underset{\sim}{u}(x)\right)\cos(\omega t).$$

So, from (2.3), through the previous simplifications, we obtain

$$\mu \triangle \underset{\sim}{u}(x) + (\lambda+\mu)\,\underset{\sim}{\mathrm{grad}}\left(\mathrm{div}\underset{\sim}{u}(x)\right) + \omega^2\rho\,\underset{\sim}{u}(x) + \underset{\sim}{f}(x) = 0 \;\text{ or }\; \cos(\omega t) = 0.$$

If $\cos(\omega t) = 0$ then $\underset{\sim}{u}(x,t) = \underset{\sim}{u}(x)\cos(\omega t) = 0$. In this case the solution is always null and consequently not captivating. We are interested only in the equation

$$\mu \triangle \underset{\sim}{u}(x) + (\lambda+\mu)\,\underset{\sim}{\mathrm{grad}}\left(\mathrm{div}\underset{\sim}{u}(x)\right) + \omega^2\rho\,\underset{\sim}{u}(x) + \underset{\sim}{f}(x) = 0,\; x \in \Omega. \tag{2.7}$$

For the boundary of the set Ω, let us consider $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ and $\partial\Omega_1 \cap \partial\Omega_2 = \emptyset$ where $\partial\Omega_1$,$\partial\Omega_2$ are two subsets of $\partial\Omega$. Throughout the work it is assumed that $\partial\Omega_1$ is a flat surface. Considering that exists a tension in the normal direction to $\partial\Omega$ on $\partial\Omega_1$, the corresponding boundary condition is such that

$$\underset{\approx}{\sigma}(\underset{\sim}{u})\underset{\sim}{n} = \underset{\sim}{g} \text{ on } \partial\Omega_1, \tag{2.8}$$

where $\underset{\sim}{g}$ is a vector function, which represents the force exercised in $\partial\Omega_1$. It should also be considered that the displacements are null in the boundary $\partial\Omega_2$, i.e.,

$$\underset{\sim}{u} = 0 \text{ in } \partial\Omega_2. \tag{2.9}$$

The model (2.7), (2.8) and (2.9) can be summarized as the following system of equations

$$\left\{\begin{array}{rcl}
\mu\left(\frac{\partial^2 u_1}{\partial x_1^2}+\frac{\partial^2 u_1}{\partial x_2^2}+\frac{\partial^2 u_1}{\partial x_3^2}\right)+(\lambda+\mu)\left[\frac{\partial}{\partial x_1}\left(\frac{\partial u_1}{\partial x_1}+\frac{\partial u_2}{\partial x_2}+\frac{\partial u_3}{\partial x_3}\right)\right]+\omega^2\rho u_1+f_1 &=& 0,\ x\in\Omega,\\
\mu\left(\frac{\partial^2 u_2}{\partial x_1^2}+\frac{\partial^2 u_2}{\partial x_2^2}+\frac{\partial^2 u_2}{\partial x_3^2}\right)+(\lambda+\mu)\left[\frac{\partial}{\partial x_2}\left(\frac{\partial u_1}{\partial x_1}+\frac{\partial u_2}{\partial x_2}+\frac{\partial u_3}{\partial x_3}\right)\right]+\omega^2\rho u_2+f_2 &=& 0,\ x\in\Omega,\\
\mu\left(\frac{\partial^2 u_3}{\partial x_1^2}+\frac{\partial^2 u_3}{\partial x_2^2}+\frac{\partial^2 u_3}{\partial x_3^2}\right)+(\lambda+\mu)\left[\frac{\partial}{\partial x_3}\left(\frac{\partial u_1}{\partial x_1}+\frac{\partial u_2}{\partial x_2}+\frac{\partial u_3}{\partial x_3}\right)\right]+\omega^2\rho u_3+f_3 &=& 0,\ x\in\Omega,\\
\underset{\approx}{\sigma}(\underset{\sim}{u})\underset{\sim}{n} &=& \underset{\sim}{g},\ x\in\partial\Omega_1,\\
\underset{\sim}{u} &=& 0,\ x\in\partial\Omega_2.
\end{array}\right. \tag{2.10}$$

2.2 Weak Formulation

In this section we intend to present the weak formulation of the model (2.7), (2.8) and (2.9). For this, let's start by defining the functional spaces, inner products and their associated norms which will be needed.

Consider $\underset{\sim}{u}$ and $\underset{\approx}{a}$ functions defined in the domain Ω and the Lebesgue spaces

$$\underset{\sim}{L}^2(\Omega)=\left\{\underset{\sim}{u}:||\underset{\sim}{u}||_{\underset{\sim}{L}^2(\Omega)}<\infty\right\},\quad \underset{\approx}{L}^2(\Omega)=\left\{\underset{\approx}{a}:||\underset{\approx}{a}||_{\underset{\approx}{L}^2(\Omega)}<\infty\right\}$$

where

$$||\underset{\sim}{u}||_{\underset{\sim}{L}^2(\Omega)}=(\underset{\sim}{u},\underset{\sim}{u})^{1/2}_{\underset{\sim}{L}^2(\Omega)}=\left(\int_\Omega|\underset{\sim}{u}(x)|^2dx\right)^{1/2},\quad ||\underset{\approx}{a}||_{\underset{\approx}{L}^2(\Omega)}=(\underset{\approx}{a}:\underset{\approx}{a})^{1/2}_{\underset{\approx}{L}^2(\Omega)}=\left(\int_\Omega|\underset{\approx}{a}(x)|^2dx\right)^{1/2}.$$

The spaces $\underset{\sim}{L}^2(\Omega)$ and $\underset{\approx}{L}^2(\Omega)$ are endowed with the inner products

$$(\underset{\sim}{u},\underset{\sim}{v})_{\underset{\sim}{L}^2(\Omega)}=\int_\Omega\underset{\sim}{u}\cdot\underset{\sim}{v}\,dx=\sum_{i=1}^{3}\int_\Omega u_iv_i\,dx,\qquad (\underset{\approx}{a}:\underset{\approx}{b})_{\underset{\approx}{L}^2(\Omega)}=\int_\Omega\underset{\approx}{a}:\underset{\approx}{b}\,dx=\sum_{i=1}^{3}\sum_{j=1}^{3}\int_\Omega a_{ij}b_{ij}\,dx,$$

respectively, which induce the norms defined before. In space

$$\underset{\sim}{H}^1(\Omega)=\left\{\underset{\sim}{u}\in\underset{\sim}{L}^2(\Omega),\partial\underset{\sim}{u}/\partial x_i\in\underset{\sim}{L}^2(\Omega),i=1,...,3\right\}$$

is equipped with the following inner product

$$(\underset{\sim}{u},\underset{\sim}{v})_{\underset{\sim}{H}^1(\Omega)}=(\underset{\approx}{\mathrm{grad}}\underset{\sim}{u}:\underset{\approx}{\mathrm{grad}}\underset{\sim}{v})_{\underset{\approx}{L}^2(\Omega)}+(\underset{\sim}{u},\underset{\sim}{v})_{\underset{\sim}{L}^2(\Omega)}.$$

Let $\underset{\sim}{V}=\left\{\underset{\sim}{v}\in\underset{\sim}{H}^1(\Omega):\underset{\sim}{v}|_{\partial\Omega_2}=0\right\}$. To obtain the weak formulation, the equation (2.7) is multiplied on both sides by a test function $\underset{\sim}{v}\in\underset{\sim}{V}$ and integrated over the domain Ω. In this way, we have

$$\int_\Omega\mu\triangle\underset{\sim}{u}\cdot\underset{\sim}{v}+(\lambda+\mu)\underset{\sim}{\mathrm{grad}}\left(\mathrm{div}\underset{\sim}{u}\right)\cdot\underset{\sim}{v}+\omega^2\rho\underset{\sim}{u}\cdot\underset{\sim}{v}+\underset{\sim}{f}\cdot\underset{\sim}{v}\,dx=0. \tag{2.11}$$

Before integration by parts, we reorganize the above equality applying the Schwarz's theorem [12] to the partial derivatives that are multiplied by the coefficient μ, for $i \neq j$. We have,

$$\begin{aligned}
(\lambda+\mu)\sum_{i,j=1}^{3}\left(\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_i}\right)\right)v_j &= (\lambda+\mu)\left(\sum_{i=1}^{3}\left(\frac{\partial}{\partial x_i}\left(\frac{\partial u_i}{\partial x_i}\right)\right)v_i+\sum_{i\neq j}^{3}\left(\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_i}\right)\right)v_j\right)\\
&= (\lambda+\mu)\sum_{i=1}^{3}\left(\frac{\partial}{\partial x_i}\left(\frac{\partial u_i}{\partial x_i}\right)\right)v_i+\sum_{i\neq j}^{3}\lambda\left(\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_i}\right)\right)v_j+\mu\left(\frac{\partial}{\partial x_i}\left(\frac{\partial u_i}{\partial x_j}\right)\right)v_j\\
&= \lambda\sum_{i,j=1}^{3}\left(\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_i}\right)\right)v_j+\mu\left(\sum_{i=1}^{3}\left(\frac{\partial}{\partial x_i}\left(\frac{\partial u_i}{\partial x_i}\right)\right)v_i+\sum_{i\neq j}^{3}\left(\frac{\partial}{\partial x_i}\left(\frac{\partial u_i}{\partial x_j}\right)\right)v_j\right).
\end{aligned}$$

To obtain the variational formulation in the desired form, we note that the equation (2.11) is equivalent to

$$\begin{aligned}
&\sum_{i,j=1}^{3}\int_\Omega \mu\left(\frac{\partial}{\partial x_i}\left(\frac{\partial u_j}{\partial x_i}\right)\right)v_j+\lambda\left(\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_i}\right)\right)v_j\,dx\\
&\quad+\mu\left(\sum_{i=1}^{3}\int_\Omega\left(\frac{\partial}{\partial x_i}\left(\frac{\partial u_i}{\partial x_i}\right)\right)v_i\,dx+\sum_{i\neq j}^{3}\int_\Omega\left(\frac{\partial}{\partial x_i}\left(\frac{\partial u_i}{\partial x_j}\right)\right)v_j\,dx\right)\\
&\quad+\sum_{i=1}^{3}\int_\Omega \omega^2\rho u_i v_i+f_i v_i\,dx=0.
\end{aligned}$$

Next we will present the integration by parts of the last equality

$$\begin{aligned}
&\sum_{i,j=1}^{3}\int_\Omega \mu\left(\frac{\partial u_j}{\partial x_i}\right)\left(\frac{\partial v_j}{\partial x_i}\right)+\lambda\left(\frac{\partial u_i}{\partial x_i}\right)\left(\frac{\partial v_j}{\partial x_j}\right)dx+\mu\sum_{i=1}^{3}\int_\Omega\left(\frac{\partial u_i}{\partial x_i}\right)\left(\frac{\partial v_i}{\partial x_i}\right)dx\\
&\quad+\mu\sum_{i\neq j}^{3}\int_\Omega\left(\frac{\partial u_i}{\partial x_j}\right)\left(\frac{\partial v_j}{\partial x_i}\right)dx-\sum_{i=1}^{3}\int_\Omega \omega^2\rho u_i v_i+f_i v_i\,dx \qquad (2.12)\\
&= \sum_{i,j=1}^{3}\int_{\partial\Omega}\mu\left(\frac{\partial u_j}{\partial x_i}\right)n_i v_j+\lambda\left(\frac{\partial u_i}{\partial x_i}\right)n_j v_j\,ds+\mu\sum_{i=1}^{3}\int_{\partial\Omega}\left(\frac{\partial u_i}{\partial x_i}\right)n_i v_i\,ds\\
&\quad+\mu\sum_{i\neq j}^{3}\int_{\partial\Omega}\left(\frac{\partial u_i}{\partial x_j}\right)n_i v_j\,ds,
\end{aligned}$$

where $\underset{\sim}{n}=(n_1,n_2,n_3)$ is the unit outward normal. The following lemmas will be used to simplify the equality (2.12).

Lemma 1. For $\underset{\sim}{u},\underset{\sim}{v}\in\underset{\sim}{V}$ and any vector $\underset{\sim}{n}\in\mathbb{R}^3$ holds

$$\underset{\approx}{\sigma}(\underset{\sim}{u})\underset{\sim}{n}\cdot\underset{\sim}{v}=\sum_{i,j=1}^{3}\mu\left(\frac{\partial u_j}{\partial x_i}\right)n_i v_j+\lambda\sum_{i,j=1}^{3}\left(\frac{\partial u_i}{\partial x_i}\right)n_j v_j+\mu\left(\sum_{i=1}^{3}\left(\frac{\partial u_i}{\partial x_i}\right)n_i v_i+\sum_{i\neq j}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)n_i v_j\right).$$

Proof. We mentioned that $\underset{\approx}{\sigma}(\underset{\sim}{u}) = 2\mu\underset{\approx}{\varepsilon}(\underset{\sim}{u}) + \lambda tr(\underset{\approx}{\varepsilon}(\underset{\sim}{u}))\mathrm{I}_3$. Therefore, according the previously notation, we have that $\underset{\approx}{\sigma}(\underset{\sim}{u})$ is given by

$$2\mu\begin{pmatrix} \frac{\partial u_1}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2}+\frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3}+\frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2}+\frac{\partial u_2}{\partial x_1}\right) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3}+\frac{\partial u_3}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2}\right) & \frac{\partial u_3}{\partial x_3} \end{pmatrix} + \lambda\sum_{i=1}^{3}\frac{\partial u_i}{\partial x_i}\mathrm{I}_3$$

i.e.,

$$2\mu\begin{pmatrix} \frac{\partial u_1}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial u_2}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial u_3}{\partial x_3} \end{pmatrix} + \mu\begin{pmatrix} 0 & \frac{\partial u_1}{\partial x_2}+\frac{\partial u_2}{\partial x_1} & \frac{\partial u_1}{\partial x_3}+\frac{\partial u_3}{\partial x_1} \\ \frac{\partial u_1}{\partial x_2}+\frac{\partial u_2}{\partial x_1} & 0 & \frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2} \\ \frac{\partial u_1}{\partial x_3}+\frac{\partial u_3}{\partial x_1} & \frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2} & 0 \end{pmatrix} + \lambda\textstyle\sum_{i=1}^{3}\frac{\partial u_i}{\partial x_i}\mathrm{I}_3.$$

When multiplying the matrices with the vector $\underset{\sim}{n}$ and making the internal product of the result with the test function $\underset{\sim}{v}$, we obtain that $\underset{\approx}{\sigma}(\underset{\sim}{u})\underset{\sim}{n}\cdot\underset{\sim}{v}$ is given by

$$2\mu\begin{pmatrix} \frac{\partial u_1}{\partial x_1}n_1 \\ \frac{\partial u_2}{\partial x_2}n_2 \\ \frac{\partial u_3}{\partial x_3}n_3 \end{pmatrix}\cdot\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} + \lambda\sum_{i=1}^{3}\frac{\partial u_i}{\partial x_i}\begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}\cdot\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} + \mu\begin{pmatrix} \left(\frac{\partial u_1}{\partial x_2}+\frac{\partial u_2}{\partial x_1}\right)n_2 + \left(\frac{\partial u_1}{\partial x_3}+\frac{\partial u_3}{\partial x_1}\right)n_3 \\ \left(\frac{\partial u_1}{\partial x_2}+\frac{\partial u_2}{\partial x_1}\right)n_1 + \left(\frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2}\right)n_3 \\ \left(\frac{\partial u_1}{\partial x_3}+\frac{\partial u_3}{\partial x_1}\right)n_1 + \left(\frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2}\right)n_2 \end{pmatrix}\cdot\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

$$\begin{aligned} &= 2\mu\sum_{i=1}^{3}\left(\frac{\partial u_i}{\partial x_i}\right)n_iv_i + \lambda\sum_{j=1}^{3}\sum_{i=1}^{3}\left(\frac{\partial u_i}{\partial x_i}\right)n_jv_j + \mu\sum_{i\neq j}^{3}\left(\frac{\partial u_j}{\partial x_i}+\frac{\partial u_i}{\partial x_j}\right)n_iv_j \\ &= \mu\sum_{i=j}^{3}\left(\frac{\partial u_j}{\partial x_i}\right)n_iv_j + \mu\sum_{i=1}^{3}\left(\frac{\partial u_i}{\partial x_i}\right)n_iv_i + \lambda\sum_{i,j=1}^{3}\left(\frac{\partial u_i}{\partial x_i}\right)n_jv_j + \mu\sum_{i\neq j}^{3}\left(\frac{\partial u_j}{\partial x_i}\right)n_iv_j \\ &\quad +\mu\sum_{i\neq j}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)n_iv_j \\ &= \mu\sum_{i,j=1}^{3}\left(\frac{\partial u_j}{\partial x_i}\right)n_iv_j + \lambda\sum_{i,j=1}^{3}\left(\frac{\partial u_i}{\partial x_i}\right)n_jv_j + \mu\left(\sum_{i=1}^{3}\left(\frac{\partial u_i}{\partial x_i}\right)n_iv_i + \sum_{i\neq j}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)n_iv_j\right). \end{aligned}$$

□

Lemma 2. Let $\underset{\approx}{\mathrm{grad}}\underset{\sim}{u} = \left(\frac{\partial u_i}{\partial x_j}\right)_{1\leq i,j\leq 3}$ and $\underset{\approx}{\varepsilon}(\underset{\sim}{u}) = \frac{1}{2}\left(\underset{\approx}{\mathrm{grad}}\underset{\sim}{u} + (\underset{\approx}{\mathrm{grad}}\underset{\sim}{u})^T\right)$. Then

$$2\mu\underset{\approx}{\varepsilon}(\underset{\sim}{u}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v}) = \mu\underset{\approx}{\mathrm{grad}}\underset{\sim}{u}\,\underset{\approx}{\mathrm{grad}}\underset{\sim}{v} + \mu\sum_{i=1}^{3}\left(\frac{\partial u_i}{\partial x_i}\right)\left(\frac{\partial v_i}{\partial x_i}\right) + \mu\sum_{i\neq j}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)\left(\frac{\partial v_j}{\partial x_i}\right).$$

Proof. Starting with the first term, we observe that

$$2\mu\underset{\approx}{\varepsilon}(\underset{\sim}{u}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v}) = \frac{\mu}{2}\left(\underset{\approx}{\mathrm{grad}}\underset{\sim}{u} + (\underset{\approx}{\mathrm{grad}}\underset{\sim}{u})^T\right) : \left(\underset{\approx}{\mathrm{grad}}\underset{\sim}{v} + (\underset{\approx}{\mathrm{grad}}\underset{\sim}{v})^T\right)$$

$$= \frac{\mu}{2}\left(2\underset{\approx}{\mathrm{grad}}\underset{\sim}{u} : \underset{\approx}{\mathrm{grad}}\underset{\sim}{v} + 2\underset{\approx}{\mathrm{grad}}\underset{\sim}{u} : (\underset{\approx}{\mathrm{grad}}\underset{\sim}{v})^T\right) = \mu\underset{\approx}{\mathrm{grad}}\underset{\sim}{u} : \underset{\approx}{\mathrm{grad}}\underset{\sim}{v} + \mu\underset{\approx}{\mathrm{grad}}\underset{\sim}{u} : (\underset{\approx}{\mathrm{grad}}\underset{\sim}{v})^T$$

because $\underset{\approx}{\mathrm{grad}}\underset{\sim}{u} : (\underset{\approx}{\mathrm{grad}}\underset{\sim}{v})^T = (\underset{\approx}{\mathrm{grad}}\underset{\sim}{u})^T : \underset{\approx}{\mathrm{grad}}\underset{\sim}{v}$ and $\underset{\approx}{\mathrm{grad}}\underset{\sim}{u} : \underset{\approx}{\mathrm{grad}}\underset{\sim}{v} = (\underset{\approx}{\mathrm{grad}}\underset{\sim}{u})^T : (\underset{\approx}{\mathrm{grad}}\underset{\sim}{v})^T$. It remains to prove that $\sum_{i=1}^{3}\left(\frac{\partial u_i}{\partial x_i}\right)\left(\frac{\partial v_i}{\partial x_i}\right) + \sum_{i\neq j}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)\left(\frac{\partial v_j}{\partial x_i}\right) = \underset{\approx}{\mathrm{grad}}\underset{\sim}{u} : (\underset{\approx}{\mathrm{grad}}\underset{\sim}{v})^T$. As

$$\underset{\approx}{\text{grad}}\underset{\sim}{u}:(\underset{\approx}{\text{grad}}\underset{\sim}{v})^T=\begin{pmatrix}\frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3}\\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3}\\ \frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3}\end{pmatrix}:\begin{pmatrix}\frac{\partial v_1}{\partial x_1} & \frac{\partial v_2}{\partial x_1} & \frac{\partial v_3}{\partial x_1}\\ \frac{\partial v_1}{\partial x_2} & \frac{\partial v_2}{\partial x_2} & \frac{\partial v_3}{\partial x_2}\\ \frac{\partial v_1}{\partial x_3} & \frac{\partial v_2}{\partial x_3} & \frac{\partial v_3}{\partial x_3}\end{pmatrix}=\Sigma_{i=1}^3\left(\frac{\partial u_i}{\partial x_i}\right)\left(\frac{\partial v_i}{\partial x_i}\right)+\Sigma_{i\neq j}^3\left(\frac{\partial u_i}{\partial x_j}\right)\left(\frac{\partial v_j}{\partial x_i}\right),$$

we obtain the desired equality. □

Applying Lemma 1 to (2.12), we conclude that the equation (2.11) is equivalent to:

$$\sum_{i,j=1}^{3}\int_\Omega \mu\left(\frac{\partial u_j}{\partial x_i}\right)\left(\frac{\partial v_j}{\partial x_i}\right)dx+\lambda\sum_{i,j=1}^{3}\int_\Omega\left(\frac{\partial u_i}{\partial x_i}\right)\left(\frac{\partial v_j}{\partial x_j}\right)dx+\mu\sum_{i=1}^{3}\int_\Omega\left(\frac{\partial u_i}{\partial x_i}\right)\left(\frac{\partial v_i}{\partial x_i}\right)dx$$
$$+\mu\sum_{i\neq j}^{3}\int_\Omega\left(\frac{\partial u_i}{\partial x_j}\right)\left(\frac{\partial v_j}{\partial x_i}\right)dx-\sum_{i=1}^{3}\int_\Omega \omega^2\rho u_iv_i+f_iv_i\,dx=\int_{\partial\Omega}\underset{\approx}{\sigma}(\underset{\sim}{u})\underset{\sim}{n}\cdot\underset{\sim}{v}\,ds. \tag{2.13}$$

The expression (2.13) can be represented in a more condensed way:

$$\int_\Omega \mu\underset{\approx}{\text{grad}}\underset{\sim}{u}:\underset{\approx}{\text{grad}}\underset{\sim}{v}+\lambda\text{div}\underset{\sim}{u}\,\text{div}\underset{\sim}{v}+\mu\left(\sum_{i=1}^{3}\left(\frac{\partial u_i}{\partial x_i}\right)\left(\frac{\partial v_i}{\partial x_i}\right)+\sum_{i\neq j}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)\left(\frac{\partial v_j}{\partial x_i}\right)\right)$$
$$-\omega^2\rho\underset{\sim}{u}\cdot\underset{\sim}{v}-\underset{\sim}{f}\cdot\underset{\sim}{v}\,dx=\int_{\partial\Omega}\underset{\approx}{\sigma}(\underset{\sim}{u})\underset{\sim}{n}\cdot\underset{\sim}{v}\,ds. \tag{2.14}$$

By applying Lemma 2, we obtain

$$\int_\Omega 2\mu\underset{\approx}{\varepsilon}(\underset{\sim}{u}):\underset{\approx}{\varepsilon}(\underset{\sim}{v})+\lambda\text{div}\underset{\sim}{u}\,\text{div}\underset{\sim}{v}-\omega^2\rho\underset{\sim}{u}\cdot\underset{\sim}{v}\,dx=\int_{\partial\Omega}\underset{\approx}{\sigma}(\underset{\sim}{u})\underset{\sim}{n}\cdot\underset{\sim}{v}\,ds+\int_\Omega\underset{\sim}{f}\cdot\underset{\sim}{v}\,dx.$$

In the boundary, given the conditions (2.8), (2.9) and taking into account that $\underset{\sim}{v}\in\underset{\sim}{V}$, so $v=0$ in $\partial\Omega_2$, we have that

$$\int_{\partial\Omega}\underset{\approx}{\sigma}(\underset{\sim}{u})\underset{\sim}{n}\cdot\underset{\sim}{v}\,ds=\int_{\partial\Omega_1}\underset{\approx}{\sigma}(\underset{\sim}{u})\underset{\sim}{n}\cdot\underset{\sim}{v}\,ds+\int_{\partial\Omega_2}\underset{\approx}{\sigma}(\underset{\sim}{u})\underset{\sim}{n}\cdot\underset{\sim}{v}\,ds=\int_{\partial\Omega_1}\underset{\sim}{g}\cdot\underset{\sim}{v}\,ds.$$

In this way, the weak formulation of (2.10) takes the form: find $\underset{\sim}{u}\in\underset{\sim}{V}$ such that

$$a(\underset{\sim}{u},\underset{\sim}{v})=l(\underset{\sim}{v}),\ \forall\underset{\sim}{v}\in\underset{\sim}{V}, \tag{2.15}$$

where

$$a(\underset{\sim}{u},\underset{\sim}{v})=\int_\Omega 2\mu\underset{\approx}{\varepsilon}(\underset{\sim}{u}):\underset{\approx}{\varepsilon}(\underset{\sim}{v})+\lambda\text{div}\underset{\sim}{u}\,\text{div}\underset{\sim}{v}-\omega^2\rho\underset{\sim}{u}\cdot\underset{\sim}{v}\,dx \tag{2.16}$$

and

$$l(\underset{\sim}{v})=\int_{\partial\Omega_1}\underset{\sim}{g}\cdot\underset{\sim}{v}ds+\int_\Omega\underset{\sim}{f}\cdot\underset{\sim}{v}\,dx.$$

2.3 Well Posedness

The objective of this section is study the existence and uniqueness of solutions of the problem (2.15). For this purpose, we will consider the following definitions and theorems:

Definition 1. [15, $Section$ 2.4.3] Let Ω be a limited domain and let's consider the Hilbert's space $\underset{\sim}{V} = \underset{\sim}{H}^1(\Omega)$. A bilinear form $a : \underset{\sim}{V} \times \underset{\sim}{V} \to \mathbb{C}$ is said to be $\underset{\sim}{V}$- coercive if it satisfies, for all $\underset{\sim}{u} \in \underset{\sim}{V}$, the inequality of Gårding

$$\left| a(\underset{\sim}{u}, \underset{\sim}{u}) + C||\underset{\sim}{u}||^2_{\underset{\sim}{L}^2(\Omega)} \right| \geq \alpha ||\underset{\sim}{u}||^2_{\underset{\sim}{H}^1(\Omega)},$$

where C and α are positive constants.

Definition 2. [15, $Section$ 2.4.1]
A bilinear form $b : \underset{\sim}{V} \times \underset{\sim}{V} \to \mathbb{C}$ being $\underset{\sim}{V}$ an Hilbert's space is said to be $\underset{\sim}{V}$- elliptic, if there exists $\alpha > 0$ such that

$$\left| b(\underset{\sim}{u}, \underset{\sim}{u}) \right| \geq \alpha ||\underset{\sim}{u}||^2_{\underset{\sim}{V}}$$

for all $\underset{\sim}{u} \in \underset{\sim}{V}$.

Definition 3. [15, $Section$ 2.4.1]
A bilinear form $a : \underset{\sim}{V} \times \underset{\sim}{V} \to \mathbb{C}$ defined on the Hilbert's space $\underset{\sim}{V}$, is said to be continuous if $\exists\, M > 0$ such that

$$\left| a(\underset{\sim}{u}, \underset{\sim}{v}) \right| \leq M ||\underset{\sim}{u}||_{\underset{\sim}{V}} ||\underset{\sim}{v}||_{\underset{\sim}{V}},$$

for all $\underset{\sim}{u}, \underset{\sim}{v} \in \underset{\sim}{V}$.

Applying the Riesz-Schauder theory [16, Theorem 6.5.15] we define the following result.

Theorem 1. Let $a(\cdot,\cdot)$ be a coercive bilinear form such that $a(\underset{\sim}{u}, \underset{\sim}{v}) = b(\underset{\sim}{u}, \underset{\sim}{v}) + \beta_1 (\underset{\sim}{u}, \underset{\sim}{v})_{\underset{\sim}{L}^2}$, being $b(\cdot,\cdot)$ a $\underset{\sim}{V}$ – elliptic bilinear form. For each $\beta_1 \in \mathbb{C}$ we have one of the following alternatives:

1. The problem $a(\underset{\sim}{u}, \underset{\sim}{v}) = l(\underset{\sim}{v})$ has a unique solution;

2. β_1 is an eigenvalue of the problem.

Theorem 2. [17, $Second\,Korn\,inequality$]
There exists a positive constant C such that

$$||\underset{\approx}{\varepsilon}(\underset{\sim}{v})||_{\underset{\approx}{L}^2(\Omega)} \geq C ||\underset{\sim}{v}||_{\underset{\sim}{H}^1(\Omega)}, \underset{\sim}{v} \in \underset{\sim}{V}.$$

where $\underset{\sim}{V} = \left\{ \underset{\sim}{v} \in \underset{\sim}{H}^1(\Omega) : \underset{\sim}{v}|_{\partial\Omega_2} = 0 \right\}$.

The goal is to apply the Theorem 1 to study under what conditions (2.15) has a unique solution. For this purpose, it is intended to check the conditions of this theorem. Let

$$b(\underset{\sim}{v}, \underset{\sim}{v}) = \int_\Omega 2\mu \underset{\approx}{\varepsilon}(\underset{\sim}{v}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v}) + \lambda \operatorname{div} \underset{\sim}{v} \operatorname{div} \underset{\sim}{v}\, dx = 2\mu ||\underset{\approx}{\varepsilon}(\underset{\sim}{v})||^2_{\underset{\approx}{L}^2(\Omega)} + \lambda ||\operatorname{div} \underset{\sim}{v}||^2_{L^2(\Omega)}.$$

As $\lambda > 0$, from Theorem 2 we have

$$\begin{aligned} b(\underset{\sim}{v}, \underset{\sim}{v}) &\geq 2\mu ||\underset{\approx}{\varepsilon}(\underset{\sim}{v})||^2_{\underset{\approx}{L}^2(\Omega)} \\ &\geq 2\mu C^2 ||\underset{\sim}{v}||^2_{\underset{\sim}{H}^1(\Omega)}, \end{aligned}$$

so $b(\cdot,\cdot)$ is $\underset{\sim}{H}^1(\Omega)-$elliptic.

Consequently for the bilinear form $a(\cdot,\cdot)$ defined in (2.16), we have that

$$a(\underset{\sim}{v},\underset{\sim}{v}) \geq 2\mu C^2 ||\underset{\sim}{v}||^2_{\underset{\sim}{H}^1(\Omega)} - \omega^2\rho ||\underset{\sim}{v}||^2_{\underset{\sim}{L}^2(\Omega)},$$

so $a(\cdot,\cdot)$ is coercive. Because of Theorem 1 and assuming that β_1 is not an eigenvalue value of the problem, we have that (2.15) has a unique solution.

Let $\underset{\sim}{u}^*$ be this solution and let's consider a new problem: find $\underset{\sim}{u} \in \underset{\sim}{V}$ such that

$$b(\underset{\sim}{u},\underset{\sim}{v}) = l_1(\underset{\sim}{v}),\ \forall \underset{\sim}{v} \in \underset{\sim}{V}, \tag{2.17}$$

where $l_1(\underset{\sim}{v}) = l(\underset{\sim}{v}) + \omega^2\rho(\underset{\sim}{u}^*,\underset{\sim}{v})$.

To obtain the solution of this problem, we will consider the following approach. Let's define a functional $J : \underset{\sim}{V} \to \mathbb{R}$

$$J(\underset{\sim}{v}) = \frac{1}{2} b(\underset{\sim}{v},\underset{\sim}{v}) - l_1(\underset{\sim}{v}).$$

In this way it is possible to obtain a relationship between a minimization problem for this functional and the problem (2.17) from the following lemmas.

Lemma 3. [18, $Section$2.2]
Let $\underset{\sim}{u}$ be the unique solution of the weak formulation (2.17) and suppose that $b(\cdot,\cdot)$ is a symmetric bilinear functional on $\underset{\sim}{V}$. Under these conditions $\underset{\sim}{u}$ is the unique minimizer of $J(\cdot)$ on $\underset{\sim}{V}$.

Lemma 4. [18, $Section$2.2]
If $\underset{\sim}{u} \in \underset{\sim}{V}$ is the unique minimizer of $J(\cdot)$ then $\underset{\sim}{u}$ is the unique solution of the problem (2.17).

With these two lemmas it is possible to conclude the equivalence between the weak formulation (2.17) and the minimization problem defined by the functional J, provided that the weak formulation has unique solution and that $b(\cdot,\cdot)$ is a symmetric elliptic bilinear functional on $\underset{\sim}{V}$. The proof of these conditions is then presented in what follows.

The problem (2.17) has unique solution because $b(\cdot,\cdot)$ is $\underset{\sim}{H}^1(\Omega)-$elliptic [16, Theorem 6.5.9]. In addition, $b(\cdot,\cdot)$ is a symmetric bilinear form on $\underset{\sim}{H}^1(\Omega)$ as

$$b(\underset{\sim}{u},\underset{\sim}{v}) = \int_\Omega 2\mu\underset{\approx}{\varepsilon}(\underset{\sim}{u}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v}) + \lambda \operatorname{div}\underset{\sim}{u}\operatorname{div}\underset{\sim}{v}\,dx = \int_\Omega 2\mu\underset{\approx}{\varepsilon}(\underset{\sim}{v}) : \underset{\approx}{\varepsilon}(\underset{\sim}{u}) + \lambda \operatorname{div}\underset{\sim}{v}\operatorname{div}\underset{\sim}{u}\,dx = b(\underset{\sim}{v},\underset{\sim}{u}),\ \ \forall \underset{\sim}{u},\underset{\sim}{v} \in \underset{\sim}{V}.$$

Let $\underset{\sim}{\bar{u}}$ the unique solution of problem (2.17) so

$$b(\underset{\sim}{\bar{u}},\underset{\sim}{v}) = l(\underset{\sim}{v}) + \beta_1(\underset{\sim}{u}^*,\underset{\sim}{v}), \forall \underset{\sim}{v} \in \underset{\sim}{V}.$$

On the other hand, as $\underset{\sim}{u}^*$ is the unique solution of problem (2.15), we have that

$$b(\underset{\sim}{u}^*,\underset{\sim}{v}) = l(\underset{\sim}{v}) + \beta_1(\underset{\sim}{u}^*,\underset{\sim}{v}), \forall \underset{\sim}{v} \in \underset{\sim}{V}.$$

Consequently $\underset{\sim}{u}^* = \underset{\sim}{\bar{u}}$.

We conclude that $\underset{\sim}{u}^*$ is the only minimizer of $J(\cdot)$.

Chapter 3

Finite Element Method

3.1 Description of the method

Given the system of equations (2.10) resulting from the time-harmonic linear elasticity equation together with boundary conditions, we will discuss how to obtain its solution. In this context, we will approximate the exact solution by continuous piecewise linear functions considering a partition of the domain using the finite element method. The procedure for applying this method will be presented in detail in this chapter.

Let us consider the weak formulation of the problem stated in equation (2.15). The characterization of the finite element solution to obtain the approximation $\underset{\sim}{u}_h$ of $\underset{\sim}{u}$ consists in considering a finite dimensional subspace $\underset{\sim}{V}_h \subset \underset{\sim}{V}$, which consists of continuous piecewise polynomial functions of a fixed degree associated with each element of a partition of the domain. In this contexts, h represent the diameter of the partition. Therefore, the finite element formulation of the problem (2.15) is: find $\underset{\sim}{u}_h \in \underset{\sim}{V}_h$ such that

$$a(\underset{\sim}{u}_h, \underset{\sim}{v}_h) = l(\underset{\sim}{v}_h),\ \forall \underset{\sim}{v}_h \in \underset{\sim}{V}_h. \tag{3.1}$$

Assuming the conditions of existence and uniqueness of solution of the problem (2.15) derived in Chapter 2 and for h small enough by [16, Theorem 8.2.8] we know that the discrete problem (3.1) has unique solution.

In this work, the finite element space $\underset{\sim}{V}_h$ consists of continuous piecewise linear functions. We start considering a partition of Ω into M tetrahedrons K_j, $j \in \{1,...,M\}$ so that

$$\Omega = \bigcup_{j=1}^{M} K_j \text{ and, } \operatorname{int}(K_i) \cap \operatorname{int}(K_j) = \emptyset, \forall i,j \in \{1,...,M\},\ i \neq j. \tag{3.2}$$

The resulting subdivision (or mesh) is denoted by Ω_h. To each tetrahedron there are associated four vertices that can be vertices of the interior or the border of Ω. For any pair of tetrahedrons from the partition of Ω either they don't intersect or they have in common only vertices or edges.

Assuming that N is the total number of vertices in Ω_h then $\dim \underset{\sim}{V}_h = 3N$ because each function in $\underset{\sim}{V}_h$ is a three component vector function. So we associate to each vertex a scalar basis function for each of the three components of the vector function.

Let us consider $\underset{\sim}{V_h} = \text{span}\left\{\phi_{11},...,\phi_{N1},\phi_{12},...,\phi_{N2},\phi_{13},...,\phi_{N3}\right\}$, where ϕ_{ji}, $i=1,2,3$, $j=1,...,N$ are the linearly independent basis functions. In this way, each component of the approximate solution $\underset{\sim}{u_h}$ will be written as a linear combination of the basis functions ϕ_{ji}

$$u_h = (u_{1h}, u_{2h}, u_{3h})$$

with

$$u_{ih}(x,y,z) = \sum_{j=1}^{N} U_{ji}\phi_{ji}(x,y,z)\,,\; i=1,2,3, \tag{3.3}$$

where U_{ji}, $i=1,2,3$, $j=1,...,N$ are the constants that we want to calculate.

As already mentioned, each vertex j of the partition of Ω in tetrahedrons is associated with three base functions ϕ_{ji}, $i=1,2,3$, $j=1,...,N$, one function for each component. These functions are continuous in $\overline{\Omega}$ and linear in each tetrahedron.

In this way, we can define the approximation of finite elements, from the weak formulation of elasticity equation. The purpose is to find $\underset{\sim}{u_h} \in \underset{\sim}{V_h}$ such that

$$\int_\Omega 2\mu\underset{\approx}{\varepsilon}(\underset{\sim}{u_h}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) + \lambda\text{div}\underset{\sim}{u_h}\,\text{div}\underset{\sim}{v_h} - \omega^2\rho\underset{\sim}{u_h}\cdot\underset{\sim}{v_h}\,dx = \int_{\partial\Omega_1}\underset{\sim}{g}\cdot\underset{\sim}{v_h}\,ds + \int_\Omega\underset{\sim}{f}\cdot\underset{\sim}{v_h}\,dx,\ \forall\underset{\sim}{v_h}\in\underset{\sim}{V_h}. \tag{3.4}$$

It should be noted that, if $\underset{\sim}{u_h}^*$ is the solution of the problem (3.4) then $\underset{\sim}{u_h}^*$ is the minimizer of the problem

$$J_h(\underset{\sim}{v_h}) = \frac{1}{2}b(\underset{\sim}{v_h},\underset{\sim}{v_h}) - l_{1,h}(\underset{\sim}{v_h}), \underset{\sim}{v_h}\in\underset{\sim}{V_h}, \tag{3.5}$$

with $l_{1,h}(\underset{\sim}{v_h}) = l(\underset{\sim}{v_h}) + \omega^2\rho(\underset{\sim}{u}_h^*,\underset{\sim}{v_h})$, this is, the solution of finite elements $\underset{\sim}{u_h}^*$ satisfies

$$J(\underset{\sim}{u_h}^*) = \min_{\underset{\sim}{v_h}\in\underset{\sim}{V_h}} J(\underset{\sim}{v_h}). \tag{3.6}$$

Next, it is intended to show that the problem (3.6) is equivalent to the following problem: find $V\in\mathbb{R}^{3N}$ such that

$$\frac{1}{2}V^TB^*V - V^TF_1^* \text{ is minimum,} \tag{3.7}$$

where $V = [V_{11},...,V_{N1},V_{12},...,V_{N2},V_{13},...,V_{N3}]^T$, with B^* being a $3N\times3N$ matrix and F_1^* being a vector of dimension $3N\times1$.

3.2 Deriving the numerical scheme

In this section we will present the way of calculating the global stiffness matrix B^* and the vector F_1^* of the problem (3.7). We can write the bilinear form $b(\cdot,\cdot)$ in (3.5) in the following way

$$b(\underset{\sim}{v_h},\underset{\sim}{v_h}) = \sum_{K\in\Omega_h}\int_K 2\mu\underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) + \lambda\text{div}\underset{\sim}{v_h}\,\text{div}\underset{\sim}{v_h}\,dx.$$

3.2.1 Reference tetrahedron

An important question to be resolved is how to obtain the value of the integrals over the tetrahedrons. The tetrahedrons that form the partition of the domain have different configurations. To make the calculations easier, let's use a reference tetrahedron.

For each tetrahedron K of the partition of Ω_h, consider $r_i = (x_i, y_i, z_i)$, $i = 1, ..., 4$, representing the coordinates of their vertices in a global cartesian system. Consider also the local coordinates (ξ, η, τ) where the vertices of the tetrahedron are the local coordinate axes as represented in Figure 3.1.

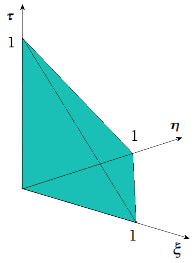

Fig. 3.1 Tetrahedron represented in local coordinates.

In this way, we can write the coordinates of each point $r = (x, y, z)$ of K as a convex combination of the local coordinates

$$r = (x,y,z) = r_1\psi_1(\xi,\eta,\tau) + r_2\psi_2(\xi,\eta,\tau) + r_3\psi_3(\xi,\eta,\tau) + r_4\psi_4(\xi,\eta,\tau) \tag{3.8}$$

where

$$\psi_1(\xi,\eta,\tau) = 1-\xi-\eta-\tau, \;\; \psi_2(\xi,\eta,\tau) = \xi,$$
$$\psi_3(\xi,\eta,\tau) = \eta, \;\; \psi_4(\xi,\eta,\tau) = \tau.$$

The elements of $\{\psi_i, i = 1, ..., 4\}$ are called the nodal basis of the set of linear polynomials in relation to local coordinates. This transformation is associated with the Jacobi matrix which is given by

$$J = \frac{\partial(x,y,z)}{\partial(\xi,\eta,\tau)} = \begin{pmatrix} x_2-x_1 & x_3-x_1 & x_4-x_1 \\ y_2-y_1 & y_3-y_1 & y_4-y_1 \\ z_2-z_1 & z_3-z_1 & z_4-z_1 \end{pmatrix}.$$

The Jacobian can be written as

$$|J| = \det\begin{pmatrix} x_2-x_1 & x_3-x_1 & x_4-x_1 \\ y_2-y_1 & y_3-y_1 & y_4-y_1 \\ z_2-z_1 & z_3-z_1 & z_4-z_1 \end{pmatrix} = \det\begin{pmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{pmatrix}.$$

Note that $|J| = 6V_{1234}$, where V_{1234} is the volume of the tetrahedron K induced by $r_1,...,r_4$. So, for any function $\underset{\sim}{v_h} \in \underset{\sim}{V_h}$,

$$v_{ih}(x,y,z) = v_{ih}(r(\xi,\eta,\tau)) = \sum_{j=1}^{4} V_{ji}\psi_j(\xi,\eta,\tau), i = 1,...,3, \tag{3.9}$$

where V_{ji} is the value of the function v_{ih} in the vertex of the tetrahedron K with position r_j, $i = 1,...,3$, $j = 1,...,4$. Realizing, in the vector form, we have

$$\begin{pmatrix} v_{1h}(x,y,z) \\ v_{2h}(x,y,z) \\ v_{3h}(x,y,z) \end{pmatrix} = \begin{pmatrix} V_{11} \\ V_{12} \\ V_{13} \end{pmatrix} \psi_1(\xi,\eta,\tau) + \begin{pmatrix} V_{21} \\ V_{22} \\ V_{23} \end{pmatrix} \psi_2(\xi,\eta,\tau) + \begin{pmatrix} V_{31} \\ V_{32} \\ V_{33} \end{pmatrix} \psi_3(\xi,\eta,\tau) + \begin{pmatrix} V_{41} \\ V_{42} \\ V_{43} \end{pmatrix} \psi_4(\xi,\eta,\tau).$$

Note that from the previous transformation and by the chain rule it is possible to obtain a relationship between the partial derivatives

$$\begin{aligned} \frac{\partial v_{jh}}{\partial \xi} &= \frac{\partial v_{jh}}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial v_{jh}}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial v_{jh}}{\partial z}\frac{\partial z}{\partial \xi}, \\ \frac{\partial v_{jh}}{\partial \eta} &= \frac{\partial v_{jh}}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial v_{jh}}{\partial y}\frac{\partial y}{\partial \eta} + \frac{\partial v_{jh}}{\partial z}\frac{\partial z}{\partial \eta}, \\ \frac{\partial v_{jh}}{\partial \tau} &= \frac{\partial v_{jh}}{\partial x}\frac{\partial x}{\partial \tau} + \frac{\partial v_{jh}}{\partial y}\frac{\partial y}{\partial \tau} + \frac{\partial v_{jh}}{\partial z}\frac{\partial z}{\partial \tau}, \end{aligned}$$

that can be written as follows:

$$\begin{pmatrix} \frac{\partial v_{jh}}{\partial \xi} & \frac{\partial v_{jh}}{\partial \eta} & \frac{\partial v_{jh}}{\partial \tau} \end{pmatrix}^T = J^T \begin{pmatrix} \frac{\partial v_{jh}}{\partial x} & \frac{\partial v_{jh}}{\partial y} & \frac{\partial v_{jh}}{\partial z} \end{pmatrix}^T, j = 1,2,3. \tag{3.10}$$

The inverse $\left(J^T\right)^{-1}$ exists, and $\begin{pmatrix} \frac{\partial v_{jh}}{\partial x} & \frac{\partial v_{jh}}{\partial y} & \frac{\partial v_{jh}}{\partial z} \end{pmatrix}^T = \left(J^{-1}\right)^T \begin{pmatrix} \frac{\partial v_{jh}}{\partial \xi} & \frac{\partial v_{jh}}{\partial \eta} & \frac{\partial v_{jh}}{\partial \tau} \end{pmatrix}^T$ with

$$\left(J^{-1}\right)^T = \frac{(J^*)^T}{|J|} = \frac{1}{|J|}\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

where J^* is the adjugate matrix of J and

$$\begin{aligned} a_{11} &= (y_3-y_1)(z_4-z_1)-(z_3-z_1)(y_4-y_1), & a_{12} &= (z_2-z_1)(y_4-y_1)-(y_2-y_1)(z_4-z_1), \\ a_{13} &= (y_2-y_1)(z_3-z_1)-(z_2-z_1)(y_3-y_1), & a_{21} &= (z_3-z_1)(x_4-x_1)-(x_3-x_1)(z_4-z_1), \\ a_{22} &= (x_2-x_1)(z_4-z_1)-(z_2-z_1)(x_4-x_1), & a_{23} &= (z_2-z_1)(x_3-x_1)-(x_2-x_1)(z_3-z_1), \\ a_{31} &= (x_3-x_1)(y_4-y_1)-(y_3-y_1)(x_4-x_1), & a_{32} &= (y_2-y_1)(x_4-x_1)-(x_2-x_1)(y_4-y_1), \\ a_{33} &= (x_2-x_1)(y_3-y_1)-(y_2-y_1)(x_3-x_1), \end{aligned}$$

with $j = 1,2,3$.

In this way, multiplying both members of (3.10) by $\left(J^T\right)^{-1}$, we obtain

$$\begin{cases} \frac{\partial v_{jh}}{\partial x}|J| = a_{11}\frac{\partial v_{jh}}{\partial \xi} + a_{12}\frac{\partial v_{jh}}{\partial \eta} + a_{13}\frac{\partial v_{jh}}{\partial \tau}, & j = 1,2,3, \\ \frac{\partial v_{jh}}{\partial y}|J| = a_{21}\frac{\partial v_{jh}}{\partial \xi} + a_{22}\frac{\partial v_{jh}}{\partial \eta} + a_{23}\frac{\partial v_{jh}}{\partial \tau}, & j = 1,2,3, \\ \frac{\partial v_{jh}}{\partial z}|J| = a_{31}\frac{\partial v_{jh}}{\partial \xi} + a_{32}\frac{\partial v_{jh}}{\partial \eta} + a_{33}\frac{\partial v_{jh}}{\partial \tau}, & j = 1,2,3, \end{cases} \tag{3.11}$$

and we can write the partial derivatives given on the coordinates (x,y,z) in terms of the partial derivatives on the local coordinates.

The relations in (3.11) will be very helpful to obtain the matrix B^* and the vector F_1^* in (3.7). The calculations of the matrix B^* and the vector F_1^* are very extensive and can be found in the Appendix A.

3.2.2 Linear system

In this section we will discuss a way to obtain the solution of the problem (3.7). Let's consider the function χ, defined in the following form

$$\begin{array}{rcl} \chi : \mathbb{R}^{3N} & \to & \mathbb{R} \\ V & \mapsto & \frac{1}{2}V^T B^* V - V^T F_1^*. \end{array}$$

Note that, by Section 2.3, the function χ has a unique minimizer on $\mathbb{R}^3$.

We will denote the gradient of the function χ by, $\nabla\chi = \left(\frac{\partial \chi}{\partial V_{11}}, \dots, \frac{\partial \chi}{\partial V_{N1}}, \frac{\partial \chi}{\partial V_{12}}, \dots, \frac{\partial \chi}{\partial V_{N2}}, \frac{\partial \chi}{\partial V_{13}}, \dots, \frac{\partial \chi}{\partial V_{N3}}\right)^T$. The minimizer V of χ is such that $\nabla\chi(V) = 0$. As the matrix B^* is symmetric, $\nabla\chi(V) = B^*V - F_1^*$. In this way, the minimizer V satisfies

$$B^*V = F_1^*.$$

In this way, we calculate the coefficients U_{ji}^*, $i = 1,2,3$, $j = 1,...,N$ of (3.3) that allow to obtain the solution $\underset{\sim}{u_h}^*$ of the problem (3.4) which satisfy $B^*U^* = F_1^*$. This problem is equivalent to

$$AU = F, \tag{3.12}$$

where $A = B^* - \beta_1 B''$ is a $3N \times 3N$ matrix and $F = F_1^* - \beta_1 B'' U^*$ is a $3N \times 1$ vector with $B'' = \sum_{k=1}^{M} L_1^k B' \left(L_1^k\right)^T$.

Note that the system (3.12) is simplified since we have Dirichlet boundary conditions, $\underset{\sim}{u} = 0$, in $x \in \partial\Omega_2$. So we will consider another system where we eliminate the rows and columns of matrix A and the entries of vector F that correspond to the vertices of this boundary.

Next, we will illustrate the performance of the method with numerical experiments.

3.3 Numerical results

In this section we present some numerical results. Having in mind the problem that motivated this work, we use real parameters that characterize the retina to define the mathematical model. The implementation of the method was made in *Matlab*. We present two groups of test problems. In the first group, for simplicity, we will consider a cubic domain. In the second group, the domain is taken to be a cylinder, aiming to mimic possible configurations of the domain of interest in the project application [20].

Let Ω be a cube, $\Omega = [-2,2]^3 \subset \mathbb{R}^3$, with $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ where $\partial\Omega_1$ corresponds to the face of the cube contained in the plane $z = -2$ and $\partial\Omega_2$ corresponds to the remaining faces of the cube. Let's consider a partition of Ω into a set of M tetrahedrons.

For meshing we used the *Matlab* function $DelaunayTriangulation(x,y,z)$. In Figure 3.2 we represent a partition of the domain into a set of ten tetrahedrons.

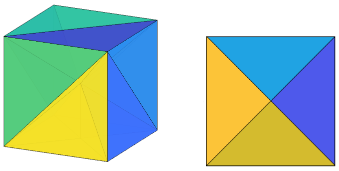

Fig. 3.2 Partition of the cube into a set of ten tetrahedrons (left); Triangulation on the boundary $\partial\Omega_1$ (right).

The equation was defined using the parameters taken from [21]: Young's modulus E and Poisson's ratio υ given by

$$\begin{aligned} E &= 2\times 10^4 Pa, \\ \upsilon &= 0.498. \end{aligned}$$

The Lamé constants are calculated from the above values using (2.4). In addition, we set the angular frequency ω to be $2\pi \times 10^6$ rad/s and the density ρ to be $1 \text{g}/cm^3$. For the vector function $\underset{\sim}{g}$ we considered

$$g_1 = 10^3, \quad g_2 = 10^3, \quad g_3 = 3\cos(z)\times 10^6. \tag{3.13}$$

In Figure 3.3, we present the numerical solution of the elasticity equation using the mesh in Figure 3.2.

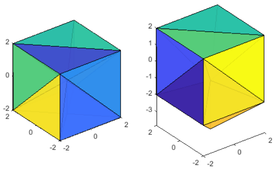

Fig. 3.3 The initial cube (left) and its deformation by the action of the force defined by (3.13) (right).

As can be seen from the Figure 3.3, the only vertex inside the lower face suffered a significant downward displacement (the third component of the point decreased) and it is still possible to observe that the inner point of the cube moved slightly in the same direction. In Figure 3.4 we present another example this time considering the same Lamé constants and the function $\underset{\sim}{g}$ defined by

$$g_1 = 10^3, \quad g_2 = 10^3, \quad g_3 = -3\cos(z) \times 10^6, \tag{3.14}$$

i.e., only by changing g_3 making it symmetrical to the function of the previous example. For the function $\underset{\sim}{g}$ defined in this way, it is expected that the vertices that suffered displacements in the previous situation, would move in the opposite direction. The results are illustrated in Figure 3.4 and in Figure 3.5.

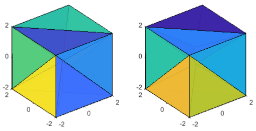

Fig. 3.4 The initial cube (left) and its deformation by the action of the force defined by (3.14) (right).

Changes are not easily visible in the two figures. For that reason, we use the Table 3.1 to show the values of the displacements in two points, one inside the cube and other located in the bottom face in the cube. In Table 3.1 we show the new coordinates of two points that result from the sum of the original coordinates with the displacements calculated.

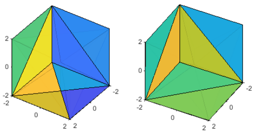

Fig. 3.5 An other point of view of the initial cube (left) and its deformation by action of the force defined by (3.14) (right).

	Original			After displacement		
Coordinates	x	y	z	x	y	z
Interior point	0	0	0	3.64×10^{-7}	-0.000917	0.911
Boundary point	0	0	-2	0.000147	-0.00211	-0.163

Table 3.1 Coordinates of original vertices and coordinates after displacement.

Let us now consider the second group of numerical results where the domain is a cylinder.

Let $\Omega = \left\{(x,y,z) : x^2 + y^2 \leq 0.25, 0 \leq z \leq 1\right\}$ with $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ where $\partial\Omega_1$ corresponds the face of the cylinder contained in the plane $z = 1$ and $\partial\Omega_2$ corresponds the remaining faces of the solid. In this domain we have some additional issues because of the curved boundary on the side face of the cylinder. The way to generate the mesh points that correspond to the vertices of the tetrahedron of the partition of the domain has to be done carefully in this situation. We will consider several planes of the form $z = \beta$, $\beta \in [0,1]$ and points equally spaced in circumferences of the shape

$$\left\{(x,y) : x^2 + y^2 = r^2\right\}, r \leq 0.5$$

at each plane. This way, we construct a mesh on the cylinder, so if we use more points, better will be the approximation of the solid. Let's consider a partition of Ω into M tetrahedrons and a triangulation of $\partial\Omega_1$ into M_1 triangles.

Let $\underset{\sim}{u} = (0,0,u_3)$, with $u_3 = -((x-0.5)^2 + (y-0.5)^2 - 0.25)z$, be the exact a solution that satisfies the boundary conditions previously imposed. To define the problem, we calculate $\underset{\sim}{f}$ in (2.7) and $\underset{\sim}{g}$ in (2.8) from the exact solution. The purpose of using a fabricate solution is to make possible to control if the numerical method works well.

The cylinder as well as the numerical displacements are shown in the Figure 3.6. We can observe that the method presents the desired behaviour. The gray shades, from the figure on the right, quantify the displacements where the darker tones correspond to greater displacements.

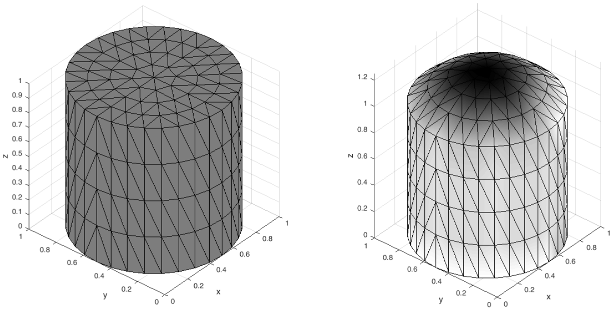

Fig. 3.6 Partition of the cylinder in to a set of tetrahedrons (left) and the corresponding displacements obtained by the numerical method (right).

Only from the perception of the shade colors in Figure 3.6, it isn't easy to have an idea of the magnitude of the displacements obtained inside the solid. To solve this issues, we plotted the results in Figure 3.7 using the software *Paraview*, combined with *Matlab*.

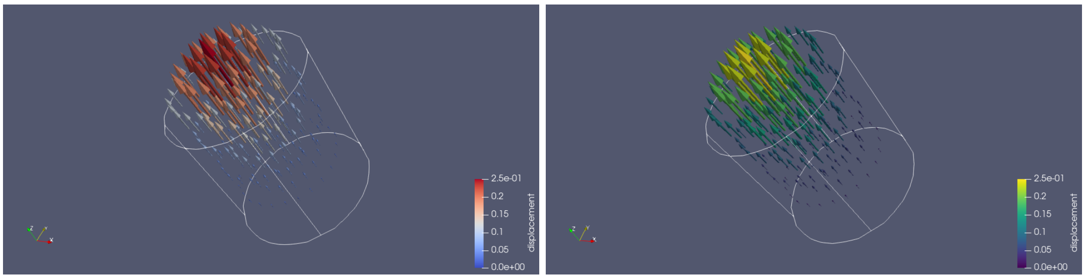

Fig. 3.7 Displacements quantified by color arrows. Numerical solution (left) and exact solution $\underset{\sim}{u}$ (right).

We can observe the displacement field looking to the colors and size of the arrows associated to the points. We conclude by, Figure 3.7, that the numerical solution and the exact solution behave in the same qualitative behaviour.

Now we will determine experimentally the order of convergence of our implementation of the finite element method for the linear elasticity equation. Here, we will consider the cubic domain $\Omega = [-2,2]^3$.

Let's consider $\underset{\sim}{u}^{exact} = (u_1^{exact}, u_2^{exact}, u_3^{exact})$ given by

$$u_i^{exact}(x_1,x_2,x_3) = \text{sen}\left(\frac{\pi}{4}(x_1+2)\right)\text{sen}\left(\frac{\pi}{4}(x_2+2)\right)\cos\left(\frac{\pi}{8}(x_3+2)\right),\ i=1,2,3, \tag{3.15}$$

which is the exact solution of (2.10), for the case $w=0$. We define $\underset{\sim}{f}$ and $\underset{\sim}{g}$ such that $\underset{\sim}{u}^{exact}$ satisfies

$$\begin{cases} -\mu \triangle \underset{\sim}{u}^{exact} - (\lambda+\mu)\,\underset{\sim}{\text{grad}}\left(\text{div}\underset{\sim}{u}^{exact}\right) &= \underset{\sim}{f},\ x\in\Omega, \\ \underset{\approx}{\sigma}(\underset{\sim}{u}^{exact})\underset{\sim}{n} &= \underset{\sim}{g},\ x\in\partial\Omega_1, \\ \underset{\sim}{u}^{exact} &= 0,\ x\in\partial\Omega_2, \end{cases}$$

where $\partial\Omega_1$ corresponds to the face of the cube contained in the plane $z=-2$ and $\partial\Omega_2$ corresponds to the remaining faces of the cube. The Lamé constants are defined by $\mu=\lambda=10$. Numerically the solution is obtained by solving the linear system defined in Section (3.2.2), that is, the solution is a $3N\times 1$ vector. We will denote the approximation solution obtained by the finite element method (3.1) as U_{approx}. We denote by U_{exact} the $3N\times 1$ vector whose entries are the exact solution defined in (3.15) at the mesh points. It is possible to quantify the error e of the numerical method using the discrete L_h^2 norm $||e||_{L_h^2(\Omega)} = ||U_{exact}-U_{approx}||_{L_h^2(\Omega)}$ where

$$||y||^2_{L_h^2(\Omega)} = \sum_{K\in\Omega_h} ||y||^2_{L_h^2(K)}$$

with

$$||y||^2_{L_h^2(K)} = \frac{V_{1234}}{4}\left[\sum_{i=1}^{4}\sum_{j=0}^{2} y^2_{t(r_i)+jN}\right], y\in\mathbb{R}^{3N}.$$

K represents each tetrahedron of the partition of Ω_h with vertices $r_i, i\in\{1,\ldots,4\}$ defined in Section 3.2.1 with volume V_{1234}. The expression $y^2_{t(r_i)+jN}$, $j\in\{0,1,2\}$ denotes the $(t(r_i)+jN)$-th squared component of the vector y. The function t is defined by

$$\begin{aligned} t:\mathbb{R}^3 &\to \{1,\ldots,N\} \\ r_i &\mapsto t(r_i), \end{aligned}$$

where $t(r_i)$ is the index that corresponds to vertex of r_i in the global numbering.

In Tables 3.2 and 3.3 we present the error for different choices of the mesh size. We denote by h the size of the subdivision of the edges of the cube to form the partition of the domain.

In the experiments, we start with an initial uniform partition of the domain and then we refine the mesh.

h	2	1	0.5
$\|\|e\|\|_{L^2_h(\Omega)}$	2.4859	1.0718	0.3416

Table 3.2 Norm of the error.

h	$4/3$	$4/9$
$\|\|e\|\|_{L^2_h(\Omega)}$	1.6028	0.2766

Table 3.3 Norm of the error.

Let us denote by e_1 and e_2 the approximation errors corresponding to two partitions of the domain with diameter h_1 and h_2, respectively. We are assuming that the norm of the errors can be written in the form

$$\|e_1\|_{L^2_{h_1}} \leq Ch_1^p \quad \text{and} \quad \|e_2\|_{L^2_{h_2}} \leq Ch_2^p.$$

Then,

$$\frac{\|e_1\|_{L^2_{h_1}}}{\|e_2\|_{L^2_{h_2}}} \leq \left(\frac{h_1}{h_2}\right)^p \text{ and consequently } p \geq \log_{\left(\frac{h_1}{h_2}\right)}\left(\frac{\|e_1\|_{L^2_{h_1}}}{\|e_2\|_{L^2_{h_2}}}\right).$$

With the data of the previous tables, it would be interesting to estimate the convergence order of this numerical method. Collecting the results of Table 3.2 and Table 3.3 it's possible to estimate p. We consider cases $h_1 = 2$, $h_2 = 1$ (case 1), $h_1 = 1$, $h_2 = 0.5$ (case 2), $h_1 = 2$, $h_2 = 0.5$ (case 3) and $h_1 = 4/3$, $h_2 = 4/9$ (case 4).

We estimate the values of p by computing

$$\log_{\left(\frac{h_1}{h_2}\right)}\left(\frac{\|e_1\|_{L^2_{h_1}}}{\|e_2\|_{L^2_{h_2}}}\right).$$

The results are presented in Table 3.4.

case	1	2	3	4
estimate for p	1.21	1.65	1.43	1.60

Table 3.4 Estimation of the convergence order p.

From the results obtained, we estimate that the convergence order is about 1.5.

Chapter 4

Inverse problem

In Chapter 3, a numerical method was developed to solve the equation of linear elasticity. This method can be used to determine the displacements induced in the retina by ultrasounds, being known the elasticity constants which characterize the medium. In other words, knowing the Lamé constants μ and λ that define the elastic properties of the retina we can compute the elastic deformations. In this chapter, it will be addressed the inverse of the previous problem, that is, knowing the displacements, the goal is to infer the values of the Lamé constants characterizing the medium under investigation. In terms of the application which is in the scope of the ElastoOCT project, the goal is to have a prior knowledge of the health status of the retina through the properties of tissues in order to avoid neurodegenerative processes and analyze the evolution of certain disease. In our approach to solve this problem, it will be necessary to consider the mathematical model that was developed for the direct problem. It is intended to find μ and λ so that the difference between the displacements predicted by the mathematical model and the observed displacements is minimized.

This chapter will be divided in three sections where we start to study the description of the inverse problem in Section 4.1. Several optimization methods were considered to solve this problem and we present a brief description of them in Section 4.2. Finally, in Section 4.3, we report the computational results about the performance of the algorithm in terms of the number of iterations and accuracy.

4.1 Description of problem

In this section we will analyze the inverse problem, which can be described by the following minimization program:

$$\begin{array}{ll} \min\limits_{\mu,\ \lambda} & \|U - U_{obs}\|^2_{L^2_h(\Omega)} \,/\, \|U_{obs}\|^2_{L^2_h(\Omega)} \\ s.t. & \qquad AU = F \\ & \qquad \mu \in [\mu_1, \mu_2] \\ & \qquad \lambda \in [\lambda_1, \lambda_2] \end{array} \tag{4.1}$$

where Ω is the domain defined in Section 2.1 and A and F are the matrix and the vector which define the linear system to solve the direct problem defined in Section 3.2.2. U_{obs} and U are the vectors with the observed displacements and the solution of the system (3.12), respectively. We are assuming that μ and λ have range values compatible with the biological structures so it makes sense working with limited domains for them. The objective function uses the discrete L_h^2 norm defined in Section 3.3. Although computational results are done with Ω being a cube, the algorithm is prepared to work for any domain. Since A is a non-singular matrix (see Section 3.2.2), the problem (4.1) can be rewritten as follows:

$$\begin{array}{lc} \min\limits_{\mu,\lambda} & \|A^{-1}F - U_{obs}\|^2_{L_h^2(\Omega)} \,/\, \|U_{obs}\|^2_{L_h^2(\Omega)} \\ s.t. & \mu \in [\mu_1, \mu_2] \\ & \lambda \in [\lambda_1, \lambda_2] \end{array} \tag{4.2}$$

The matrix A depends on μ and λ and, consequently, $U = A^{-1}F$ depends also on μ and λ. For convenience, we denote the objective function by $l(\mu,\lambda)$, that is, $l(\mu,\lambda) = \|A^{-1}F - U_{obs}\|^2_{L_h^2(\Omega)} \,/\, \|U_{obs}\|^2_{L_h^2(\Omega)}$.

Despite the fact we weren't able to obtain an explicit expression of A^{-1} in terms of these parameters, we could deduce implicit expressions for the partial derivatives of the objective function l. These expressions will be useful to characterize the convexity of the objective function and for the development of optimization methods to solve this problem. The study of the convexity of the objective function is an important aspect for this problem, since if it is convex in a convex set, all the local minimizers in that set will be global.

In what follows, our goal is to present expression for the first and second order partial derivatives of l. We will only present the deduction of the expression for the first derivative in order to μ since the derivative in order to λ follows the same reasoning. He have that

$$\begin{aligned} \frac{\partial l}{\partial \mu}(\mu,\lambda) &= \frac{\partial}{\partial \mu}\left(\|A^{-1}F - U_{obs}\|^2_{L_h^2(\Omega)}\right) / \|U_{obs}\|^2_{L_h^2(\Omega)} \\ &= \sum_{K\in\Omega_h} \frac{V_{1234}}{4\,\|U_{obs}\|^2_{L_h^2(\Omega)}} \times \sum_{i=1}^{4}\sum_{j=0}^{2}\left(\frac{\partial}{\partial \mu}\left[(A^{-1}F - U_{obs})^2_{t(r_i)+jN}\right]\right). \end{aligned} \tag{4.3}$$

Using matrix differentiation we can write that ([22, 23])

$$\frac{\partial}{\partial \mu}\left[(A^{-1}F - U_{obs})^2_{t(r_i)+jN}\right] = 2(A^{-1}F - U_{obs})_{t(r_i)+jN}\left(\left[\frac{\partial A^{-1}}{\partial \mu}\right]F\right)_{t(r_i)+jN},$$

which is the same as

$$-2(A^{-1}F - U_{obs})_{t(r_i)+jN}\left(A^{-1}\frac{\partial A}{\partial \mu}A^{-1}F\right)_{t(r_i)+jN}, \; j \in \{0,1,2\}.$$

Defining the $3N \times 1$ vectors $q = A^{-1}F - U_{obs}$ and $q_\mu = A^{-1}\frac{\partial A}{\partial \mu}A^{-1}F$ we obtain:

$$\frac{\partial}{\partial \mu}\left[(A^{-1}F - U_{obs})^2_{t(r_i)+jN}\right] = -2q(t(r_i)+jN)q_\mu(t(r_i)+jN), \; j \in \{0,1,2\}.$$

So the expression (4.3) can be written in the form

$$\frac{\partial l}{\partial \mu}(\mu,\lambda) = -\sum_{K\in\Omega_h} \frac{V_{1234}}{2\,\|U_{obs}\|^2_{L_h^2(\Omega)}}\left[\sum_{i=1}^{4}\sum_{j=0}^{2} q(t(r_i)+jN)q_\mu(t(r_i)+jN)\right].$$

In a similar way, considering that $q_\lambda = A^{-1}\frac{\partial A}{\partial \lambda}A^{-1}F$ we obtain the expression for the first derivative in order to λ:

$$\frac{\partial l}{\partial \lambda}(\mu,\lambda) = -\sum_{K\in\Omega_h}\frac{V_{1234}}{2\,\|U_{obs}\|^2_{L_h^2(\Omega)}}\left[\sum_{i=1}^{4}\sum_{j=0}^{2} q(t(r_i)+jN)q_\lambda(t(r_i)+jN)\right].$$

The derivative of A is deduced easily from the expression of B^* (defined in the Appendix), obtaining:

$$\frac{\partial A}{\partial \mu} = \frac{1}{36V_{1234}}\sum_{k=1}^{M} L_1^k\left(R^k\right)^T C_1 R^k\left(L_1^k\right)^T, \; \frac{\partial A}{\partial \lambda} = \frac{1}{36V_{1234}}\sum_{k=1}^{M} L_1^k\left(R^k\right)^T C_2 R^k\left(L_1^k\right)^T,$$

with

$$C_1 = \begin{pmatrix} 2I_3 & 0_3 \\ 0_3 & I_3 \end{pmatrix} \text{ and } C_2 = \begin{pmatrix} 1_3 & 0_3 \\ 0_3 & 0_3 \end{pmatrix},$$

where 0_3, 1_3 are square matrices with all components being zeros and ones respectively.

Next, we will present the deduction for the second order derivative in order to μ following the same steps:

$$\frac{\partial^2 l}{\partial \mu^2}(\mu,\lambda) \;=\; \sum_{K\in\Omega_h}\frac{V_{1234}}{4\,\|U_{obs}\|^2_{L_h^2(\Omega)}} \times \sum_{i=1}^{4}\sum_{j=0}^{2}\left(\frac{\partial^2}{\partial \mu^2}\left[\left(A^{-1}F - U_{obs}\right)^2_{t(r_i)+jN}\right]\right). \tag{4.4}$$

We can write the second order derivative of the term inside the parentheses in the following form:

$$2(A^{-1}\frac{\partial A}{\partial \mu}A^{-1}F)^2_{t(r_i)+jN} - 2(A^{-1}F - U_{obs})_{t(r_i)+jN}\left(\frac{\partial}{\partial \mu}\left[A^{-1}\frac{\partial A}{\partial \mu}A^{-1}\right]F\right)_{t(r_i)+jN},$$

for $j \in \{0,1,2\}$. Simplifying the derivative in order to μ of the matrix product we obtain

$$\begin{aligned}\frac{\partial}{\partial \mu}\left[A^{-1}\frac{\partial A}{\partial \mu}A^{-1}\right] &= \left[\frac{\partial A^{-1}}{\partial \mu}\right]\left[\frac{\partial A}{\partial \mu}A^{-1}\right] + A^{-1}\frac{\partial}{\partial \mu}\left[\frac{\partial A}{\partial \mu}A^{-1}\right] \\ &= -2A^{-1}\frac{\partial A}{\partial \mu}A^{-1}\frac{\partial A}{\partial \mu}A^{-1}.\end{aligned}$$

In the previous expressions note that the second order derivative of the matrix A in order to μ is zero. So we obtain the following expression:

$$2(A^{-1}\frac{\partial A}{\partial \mu}A^{-1}F)^2_{t(r_i)+jN} + 4(A^{-1}F - U_{obs})_{t(r_i)+jN}\left(A^{-1}\frac{\partial A}{\partial \mu}A^{-1}\frac{\partial A}{\partial \mu}A^{-1}F\right)_{t(r_i)+jN},$$

$j \in \{0,1,2\}$. So the expression (4.4) is the same as

$$\frac{\partial l^2}{\partial \mu^2}(\mu,\lambda) \;=\; \sum_{K\in\Omega_h}\frac{V_{1234}}{2\,\|U_{obs}\|^2_{L_h^2(\Omega)}}\left[\sum_{i=1}^{4}\sum_{j=0}^{2}\left(q_\mu(t(r_i)+jN)\right)^2 + 2q(t(r_i)+jN)w_\mu(t(r_i)+jN)\right],$$

where $w_\mu = A^{-1}\frac{\partial A}{\partial \mu}A^{-1}\frac{\partial A}{\partial \mu}A^{-1}F$ is a $3N \times 1$ vector.

In a similar way, considering the $3N \times 1$ vectors $w_\lambda = A^{-1}\frac{\partial A}{\partial \lambda}A^{-1}\frac{\partial A}{\partial \lambda}A^{-1}F$ and $w_{\mu\lambda} = A^{-1}\left(\frac{\partial A}{\partial \mu}A^{-1}\frac{\partial A}{\partial \lambda} + \frac{\partial A}{\partial \lambda}A^{-1}\frac{\partial A}{\partial \mu}\right)A^{-1}F$, we obtain the expression for the second order derivative in order to λ and also the mixed derivatives:

$$\frac{\partial l^2}{\partial \lambda^2}(\mu,\lambda) \;=\; \sum_{K\in\Omega_h} \frac{V_{1234}}{2\,\|U_{obs}\|^2_{L^2_h(\Omega)}} \left[\sum_{i=1}^{4}\sum_{j=0}^{2}(q_\lambda(t(r_i)+jN))^2+2q(t(r_i)+jN)w_\lambda(t(r_i)+jN)\right]$$

and

$$\begin{aligned}\frac{\partial l^2}{\partial \lambda \partial \mu}(\mu,\lambda) \;=\; & \sum_{K\in\Omega_h} \frac{V_{1234}}{2\,\|U_{obs}\|^2_{L^2_h(\Omega)}} \left[\sum_{i=1}^{4}\sum_{j=0}^{2} q_\lambda(t(r_i)+jN)q_\mu(t(r_i)+jN)\right.\\ & \left. +q(t(r_i)+jN)w_{\mu\lambda}(t(r_i)+jN)\right].\end{aligned}$$

The expressions previously obtained are difficult to manipulate to find out a region where the function l is convex. A way to have an hint about the convexity of l is to plot the function. As an example, we consider that U_{obs} corresponds to the solution of the direct problem with the parameters $(10,10)$, which will be designated by $(\mu_{obs},\lambda_{obs})$ and will correspond to the optimal solution of the inverse problem. Figure 4.1 (left), shows us the shape of l in the region $[5,25]^2$. Figure 4.1 (right), shows a zoom in the region $[9,11]^2$ where the optimal solution is found.

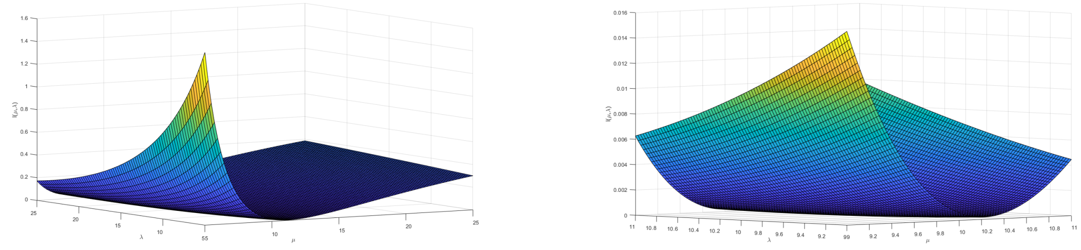

Fig. 4.1 Graph of the objective function in $[5,25]^2$ (left) and in $[9,11]^2$ (right).

From this figures it seems that the objective function l seems to be convex in a certain region. To get more information about this behavior, we compute the eigenvalues of the hessian $\nabla^2 l(\mu,\lambda)$. We consider uniformly distributed grid points in the domain $[5,25]^2$ and we calculate the signal of the corresponding eigenvalues on those points.

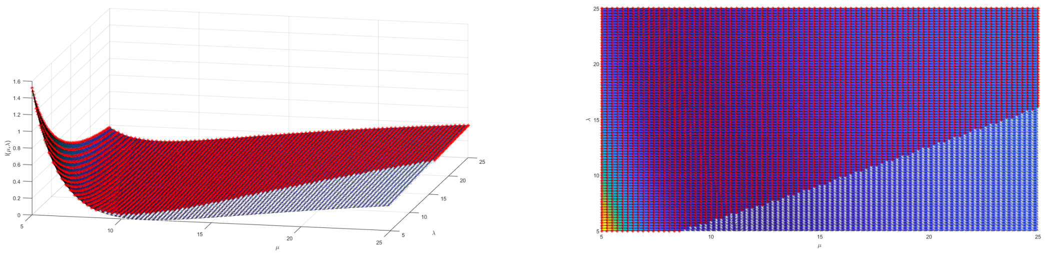

Fig. 4.2 A different perspective of the graph of the objective function in $[5,25]^2$ (left); the red points on the surface correspond to a region where the objective functions seems to be convex. A two-dimensional image with the same domain (right).

In the surface of Figure 4.2 we have red and white points where the corresponding hessian have both eigenvalues positive (the red points) and eigenvalues with different signals (the white points). We conclude that function l isn't convex.

From Figure 4.2 (right), we infer the region where the objective function l seems to be convex is given by

$$\{(\mu,\lambda) : 5 \leq \mu \leq 25 \wedge 0.681\mu - 0.836 \leq \lambda \leq 25\}.$$

Thus, in view of our assumption, if the methods converge to a stationary point inside this region, so this point is a local minimizer and, therefore, global in that region. This is what happens in our numerical results, as will be shown in Section 4.3.

4.2 Optimization methods

To solve the problem (4.2) we will present some optimization methods. It should be noted that we don't have an analytical expression for the objective function l, which initially prevented us from obtaining its derivatives. In this way, the first methods analysed were methods without derivatives, namely the coordinated search method (procedure usually used in these situations) and the trust-region method using a quadratic model obtained by the least squares. The first one is among of the most popular methods which only require the use of the function values. The other one aims to determine the minimizer of l from successive optimization sub-problems. At a later stage of this study, we were able to obtain the expressions of the partial derivatives of l presented in Section 4.1, so we also implemented the steepest-descent method and compared it with the previous ones.

We will describe the optimization methods for a general unconstrained minimization problem of a function $\vartheta(x)$, $x \in \mathbb{R}^n$. In the problem analyzed in this work, $x = (\mu,\lambda)^T$ and along the text we will use the notation x or $(\mu,\lambda)^T$ depending on which one is more convenient. For each one of these methods, we will present an example of how the algorithm works in the following academic case: a cubic domain being $\Omega = [-2,2]^3$ with $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ where $\partial\Omega_1$ corresponds to the face of the cube contained in the plane $z = -2$ and $\partial\Omega_2$ corresponds to the remaining faces of the cube. In Figure 4.3 we represent a partition of the domain Ω into a set of 48 tetrahedrons. Note that the non-null displacements calculated using this mesh correspond to two points, one inside the cube with coordinates $(0,0,0)$ and the other located in the bottom face in the cube $(0,0,-2)$.

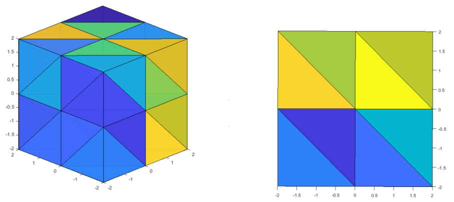

Fig. 4.3 Partition of the cube into a set of 48 tetrahedrons (left); Triangulation on the boundary $\partial\Omega_1$ (right).

Let $I = [9,11]^2$ and $(\mu_{obs}, \lambda_{obs}) = (10,10)$ the Lamé constants used to generate the observed displacement U_{obs}. The direct problem uses the following data: the function $\underset{\sim}{g}$ is defined as $g_i = 5.86 \times 10^{-3}, \forall i \in 1,2,3$, the angular frequency being $\omega = 2$ rad/s and the density of material is defined by $\rho = 1\text{g}/cm^3$. In these concrete example we know the exact solution which is $(10,10)$ so we can consider the following stop condition in the three algorithms: $||(10,10) - x_k||_2 < 0.1$, where x_k is the approximation obtained in the iteration k of the optimization methods. Here $||\cdot||_2$ denotes the euclidean norm defined by $\|y\|_2^2 = \sum_{i=1}^2 y_i^2$, $y \in \mathbb{R}^2$.

The methods presented here will be compared computationally in Section 4.3.

4.2.1 Coordinated search method

In this section we intend to apply an iterative method to find a possible local/global minimum. We choose the coordinated search method that is a direct search method of directional type that successively minimizes along coordinate directions. This option arises because it is an optimization method without derivatives which has properties of global convergence, i.e., the convergence doesn't depend on the chosen initial point [24]. We can also mention that it evaluates the objective function for a finite number of points at each iteration. In the iteration k, the method evaluates the function in a set of points near to the point x_k. If it reduces the value of the objective function we update the approximation; otherwise, the method executes a new iteration with a new set of points. The points must satisfy a certain geometry in order to achieve the convergence of the method. This geometry is established by a positive spanning set (PSS) of $\mathbb{R}^n$. The set positively generated by $\{v_1, ..., v_s\} \subset R^n$ is the convex cone

$$\left\{ v \in \mathbb{R}^n : v = \sum_{i=1}^{s} \alpha_i v_i, \alpha_i \geq 0, i = 1, ..., s \right\},$$

that is, the set of vectors which are a linear combination of vectors where all coefficients are non negative scalars. A PSS in $\mathbb{R}^n$ is a set of vectors in $\mathbb{R}^n$ that spans it positively [24]. The use of a PSS avoids to calculate the objective function in a random set of points and to each of them is possible to find a vector that is a descent direction [25]. This means that for a given PSS D and $\forall w \neq 0, \exists \bar{d} \in D : w^T \bar{d} > 0$ [24, lemma 3.2]. In other words, assuming that ϑ is continuously differentiable and $\nabla \vartheta(x) \neq 0$, there is always a descent direction for the function ϑ in x_k. If we don't find a point where the objective function reduces its value in a given iteration, the step is reduced. Since one of the directions is of descent, then by reducing the step successively we will reach a successful iteration. In general it is necessary more than a simple decrease of the function ϑ to guarantee the convergence of the method. In this sense, we can use a force function, $\rho : \mathbb{R}^+ \to \mathbb{R}^+$, that is typically continuous and satisfies [24, 26]

$$\lim_{t \to 0^+} \frac{\rho(t)}{t} = 0 \tag{4.5}$$

and $\rho(t_1) \leq \rho(t_2)$ if $t_1 < t_2$. To initialize the method we have to define the initial point x_0 and the step length α_0, which is a positive constant.

Initially the algorithm consist in a search step. For each iteration k, we compare $\vartheta(x_k)$ with the value of the function ϑ in $P_k = \{x_k + \alpha_k d : d \in D\}$, where D a given PSS of $\mathbb{R}^n$, to find a direction d_k

and the step length α_k such that the following inequality holds

$$\vartheta(x_k + \alpha_k d_k) \leq \vartheta(x_k) - \rho(\alpha_k). \tag{4.6}$$

The geometry of a PSS D (with non-zero vectors) is often evaluated using the cosine measure [24]

$$\text{cm}(D) = \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D} \frac{v^T d}{\|v\| \, \|d\|},$$

where $\|\cdot\|$ could be any norm. It can be proved that cosine measure is positive for a PSS [24]. This result will be important to support the proof of global convergence.

There are several ways to implement the algorithm but we will describe the one used in this work (Algorithm 1). We will consider that the first direction of D that satisfies the inequality (4.6) is the one chosen because it is costly to evaluate the objective function. As a consequence the order of directions in a PSS has influence in the choise of the trajectory for a given initial point. If there is a direction d_k and step α_k that satisfies (4.6), the iteration k is declared successfully and then the next point is given by $x_{k+1} = x_k + \alpha_k d_k$. Otherwise the iteration k is declared to be unsuccessful and $x_{k+1} = x_k$. To update α_{k+1} we consider $\alpha_{k+1} = \alpha_k$ when the iteration is successful; otherwise, $\alpha_{k+1} = \alpha_k/2$. Next the sketch of this method will be presented.

Algorithm 1: Coordinated search method

```
Initialization Choose $x_0$ satisfying $\vartheta(x_0) < +\infty$ and $\alpha_0 > 0$.
for $k = 0, 1, 2, \ldots$ do
    Order the set $P_k = \{x_k + \alpha_k d : d \in D\}$ for a PSS $D$. Calculate the value of the function $\vartheta$
     in $P_k$ following the order of $D$.
    if $\exists d_k \in D : \vartheta(x_k + \alpha_k d_k) \leq \vartheta(x_k) - \rho(\alpha_k)$ then
        $x_{k+1} = x_k + \alpha_k d_k$;
        $\alpha_{k+1} = \alpha_k$;
    else
        $x_{k+1} = x_k$;
        $\alpha_{k+1} = \alpha_k/2$;
```

Next, we will present the results to prove the global convergence of this optimization method, [24, 27]. The first condition ensures that the function ϑ should be bounded from below to avoid a infinity of successful iterations.

Hypothesis 1. Let ϑ be bounded from below in the set $L(x_0) = \{x \in \mathbb{R}^n : \vartheta(x) \leq \vartheta(x_0)\}$.

Hypothesis 2. The gradient $\nabla\vartheta(x)$ is Lipschitz continuous with Lipschitz constant $\delta > 0$ in a open set that contains $L(x_0)$.

In order to have convergence we have to guarantee the existence of a subsequence S of unsuccessful iterations that converges to zero:

$$\lim_{k \in S} \alpha_k = 0. \tag{4.7}$$

To continue the study, the next result should be imposed. In case of having unsuccessful iterations the following inequality holds [24]:

$$\|\nabla\vartheta(x)\| < \left(\frac{\delta}{2}\mathrm{cm}(D)^{-1}\max_{d\in D}\|d\|\right)\alpha + \left(\frac{\mathrm{cm}(D)^{-1}}{\min\limits_{d\in D}\|d\|}\right)\frac{\rho(\alpha)}{\alpha}.$$

This result relates the gradient norm of the function ϑ with the step length α as well the ratio $\rho(\alpha)/\alpha$.

From Hypothesis 1 and 2, we can easily conclude the global convergence to a stationary point.

Theorem 3. Suppose Hypothesis 1 and Hypothesis 2, then there is a subsequence S of unsuccessful iterations such that

$$\lim_{k\in S}||\nabla\vartheta(x_k)|| = 0.$$

Under the same hypothesis it is proved in [28] that

$$\min_{0\leq j\leq k}\|\nabla\vartheta(x_j)\|$$

converges sublinearly to zero with rate $1/\sqrt{k}$. If the function is convex it can be proved a sublinear global rate of $1/k$. When it's strongly convex we have a linear global rate for this method [29].

For the concrete case of function l, it wasn't possible verify the Hypothesis 2. However, the numerical results presented in Section 4.3 show that the method seems to converge in all tests performed.

We will apply the Algorithm 1 with the initial point $x_0 = (11,11)$ and the initial step $\alpha_0 = 0.1$ to show how this method works. We will consider the force function $\rho(t) = t^2/5000$, and the PSS given by

$$D = \left\{\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}-1\\0\end{bmatrix}, \begin{bmatrix}0\\-1\end{bmatrix}\right\}.$$

Initially the algorithm found the direction $d_3 = (-1,0)^T$ to decrease the function value. The approximation obtained was $x_1 = (10.9,11)$ and $\alpha_1 = 0.1$. This process repeats the same value of the step length until the iteration 20. At iteration 20, there is no point in P_{20} that decreases the value of the function, so the iteration is unsuccessful. In the next iteration the algorithm kept the same approximation but decreased the step length value for half of it. In iteration 21 there was a decrease in the value of the function. The process continues until the iteration 28 where $x_{28} = (9.975, 10.075)$ checks the stop condition $\|(10,10) - x_{28}\|_2 < 0.1$. In Figure 4.4 (left), we have the approximations obtained with the method (red points) and in white the final solution x_*. In Figure 4.4 (right) we have the trajectories for four initial points with different colors (the trajectory starting from $(11,11)$ is also included). Other PSS can be considered, for example the set

$$D_1 = \left\{\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\-1\end{bmatrix}, \begin{bmatrix}-1\\1\end{bmatrix}, \begin{bmatrix}-1\\-1\end{bmatrix}\right\}.$$

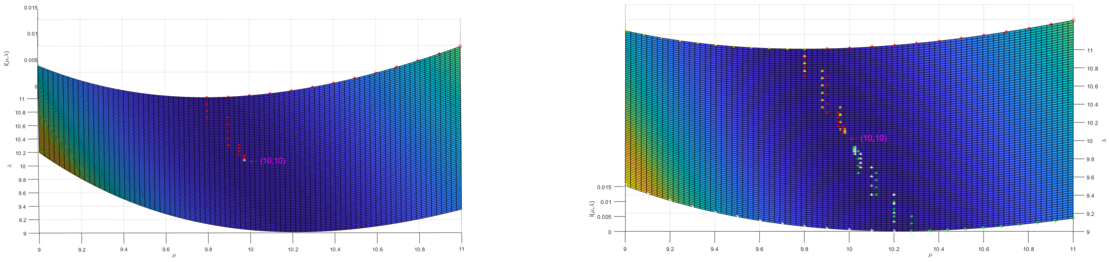

Fig. 4.4 The graphs present the objective function in I where we have successive iterations of Algorithm 1. In the left, it is presented the trajectory starting from $x_0 = (11,11)$ with $\alpha_0 = 0.1$ and in the right it is presented the trajectories starting from different initial points.

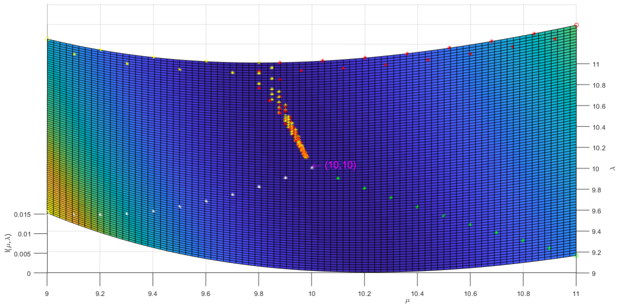

Fig. 4.5 Several trajectories obtained by Algorithm 1 for the four points with the PSS D_1. The surface represent the objective function in I.

In Figure 4.5 we present the trajectories starting in the same initial points shown in the Figure 4.4 (right). We can observe that the trajectories are different, but all seem to converge to the optimal solution. In section 4.3 a comparative study of this method with both PSS will be done.

4.2.2 Derivative free trust-region method

The configuration of the function l seems to be well approximate, locally, by a quadratic function. This fact, motivated us to apply trust region methods using a quadratic model to be an approximation of the objective function. In this section we aim to achieve the minimizer with this technique.

One of the first works about this subject uses the multivariate interpolation technique. It arises in the context of unconstrained optimization and doesn't require the use of derivatives of the objective function [30]. This fact is relevant for our work because initially we hadn't access to derivatives which is a disadvantage compared to many optimization methods. Other problem described is the high cost of obtaining the values of the objective function, something that also arises in our work. The method described in the article belongs to the class of trust region methods where the quadratic model is obtained by interpolation and the trust region is defined by the euclidean norm centered in x_k. In our work, to simplify the calculations, we will consider the infinite norm, defined by $\|y\|_\infty = \max_{1\leq i\leq n} |y_i|$, $y \in \mathbb{R}^n$, to form a square in the neighborhood of the point x_k. In general the article presents the ideas that we want to describe, however, it doesn't present the theoretical results proving the global convergence of the method.

Other studies initially consider interpolation and regression techniques to derive the quadratic models. As we obtained better results with regression, we will only describe this approach to get the quadratic models. In the regression models the number of points used is less restrictive since it can use more points (advantage). The theoretical results can be found in [28] where it is shown global convergence for trust region algorithms without derivatives. It is also presented how the models with regression and interpolation are obtained to have certain properties (depending on the geometry of the set) [31, 32]. Other works have emerged in this area to have a similar role to directional methods [34, 36]. Recently in [35] trust region methods have been applied to non-smooth problems, where the derivatives didn't exist. In these situations, other variants of the method are considered. For this method, the global convergence is proved and an efficient and robust technique for non-smooth unconstrained optimization problems is presented.

We will describe the method for $\vartheta(x)$, $x \in \mathbb{R}^2$, since the quadratic models and domains belong to this space. In this way, we will use the notation $x = (\mu, \lambda)^T$ and for the iteration k, $x_k = (\mu_k, \lambda_k)^T$.

In trust-region methods, a region is defined around the point x_k. For this region, it is defined a quadratic model that is easy to minimize and it should give a good approximation to the objective function. After that, the next candidate will be the minimizer in this region and the reduction of the objective function is compared with the reduction of the quadratic model . If the minimizer isn't accepted, the iteration is unsuccessful and the trust region radius is reduced; otherwise, we have a successful iteration, x_{k+1} is updated and the trust region radius remains the same.

The trust region radius plays an important role to obtain a successful iteration. If it is too small, the algorithm loses the opportunity to get a close approximation near the minimizer of the objective function. If it is too large the quadratic model couldn't be a good approximation of the objective function in all the region and the minimizer of the model may not be close to the minimizer of the objective function.

For iteration k, we will denote the quadratic model by l_k. Usually this model is based on Taylor's series expansion of ϑ around x_k [25] using the first and second order derivatives of the function ϑ. Here we decided to avoid the use of the derivatives of $\vartheta(x)$ so we used the least squares method to get the quadratic model which approximates $\vartheta(x)$. To build it, we use a set P_k with random points where $\vartheta(x)$ will be calculated. We will try, whenever possible, to preserve points from the previous iteration that have already been calculated to decrease the number of times the function is evaluated.

Let $P_k = \{(\mu_{i,k}, \lambda_{i,k}) : i \in \{1, ..., n_k\}\}$ be a set with $n_k \geq 9$ random points where the function ϑ is known. To determine the quadratic model of l_k we use the least squares method through the points of P_k. To deduce these models we will consider that P_k has n_k points in the set $I_k = \left[\mu_{inf,k}, \mu_{sup,k}\right] \times \left[\lambda_{inf,k}, \lambda_{sup,k}\right]$ and this set is divided into nine sub-sets as follows

$$I_{i,j,k} = \begin{cases} \left[\mu_{inf,k} + (i-1)s_k, \mu_{inf,k} + is_k\right] \times \left[\lambda_{inf,k} + (j-1)r_k, \lambda_{inf,k} + jr_k\right], & (i,j) \in \{(1,1)\}, \\ \left[\mu_{inf,k} + (i-1)s_k, \mu_{inf,k} + is_k\right] \times \left]\lambda_{inf,k} + (j-1)r_k, \lambda_{inf,k} + jr_k\right], & (i,j) \in \{(1,2),(1,3)\}, \\ \left]\mu_{inf,k} + (i-1)s_k, \mu_{inf,k} + is_k\right] \times \left[\lambda_{inf,k} + (j-1)r_k, \lambda_{inf,k} + jr_k\right], & (i,j) \in \{(2,1),(3,1)\}, \\ \left]\mu_{inf,k} + (i-1)s_k, \mu_{inf,k} + is_k\right] \times \left]\lambda_{inf,k} + (j-1)r_k, \lambda_{inf,k} + jr_k\right], & (i,j) \in \{(2,2),(2,3),(3,2),(3,3)\}, \end{cases} \tag{4.8}$$

where $s_k = \frac{1}{3}(\mu_{sup,k} - \mu_{inf,k})$, $r_k = \frac{1}{3}(\lambda_{sup,k} - \lambda_{inf,k})$ and for each sub-sets there is at least one point, however, we will assume that $x_k \in I_{2,2,k}$. This condition is imposed to make sure that the points aren't very close to each other and to make possible to have representative points around all the trust region.

So with these distributed n_k points, we want to get the k-th quadratic model l_k, that can be written as follows,

$$l_k(\mu,\lambda) = a_1 + a_2\mu + a_3\lambda + a_4\mu^2 + a_5\lambda^2 + a_6\mu\lambda,\, k \in \mathbb{Z}_0^+, \tag{4.9}$$

for some $a_i \in \mathbb{R}$, $i \in \{1,...,6\}$ that best fits the objective function. These coefficients a_i, $i \in \{1,...,6\}$, are determined in way to minimize the distance between the points considered and the approximate function. Let $z_{i,k} = l(\mu_{i,k},\lambda_{i,k})$ be the i-th value obtained in the objective function l, $i \in \{1,...,n_k\}$. The problem can be solved by minimizing the function

$$L_k(a_1,...,a_6) = \sum_{i=1}^{n_k} \left(z_{i,k} - l_k(\mu_{i,k},\lambda_{i,k})\right)^2.$$

The necessary condition to obtain the solution is that the partial derivatives of L_k need to be zero, i.e.,

$$\frac{\partial L_k}{\partial a_i}(a_1,...,a_6) = 0, \forall i \in \{1,...,6\},$$

that is equivalent to solve the following linear system of six equations

$$\begin{pmatrix} n & \sum_{i=1}^{n_k}\mu_{i,k} & \sum_{i=1}^{k}\lambda_{i,k} & \sum_{i=1}^{n_k}\mu_{i,k}^2 & \sum_{i=1}^{n_k}\lambda_{i,k}^2 & \sum_{i=1}^{n_k}\mu_{i,k}\lambda_{i,k} \\ \sum_{i=1}^{n_k}\mu_{i,k} & \sum_{i=1}^{n_k}\mu_{i,k}^2 & \sum_{i=1}^{n_k}\mu_{i,k}\lambda_{i,k} & \sum_{i=1}^{n_k}\mu_{i,k}^3 & \sum_{i=1}^{n_k}\mu_{i,k}\lambda_{i,k}^2 & \sum_{i=1}^{n_k}\mu_{i,k}^2\lambda_{i,k} \\ \sum_{i=1}^{n_k}\lambda_{i,k} & \sum_{i=1}^{n_k}\mu_{i,k}\lambda_{i,k} & \sum_{i=1}^{n_k}\lambda_{i,k}^2 & \sum_{i=1}^{n_k}\mu_{i,k}^2\lambda_{i,k} & \sum_{i=1}^{n_k}\lambda_{i,k}^3 & \sum_{i=1}^{n_k}\mu_{i,k}\lambda_{i,k}^2 \\ \sum_{i=1}^{n_k}\mu_{i,k}^2 & \sum_{i=1}^{n_k}\mu_{i,k}^3 & \sum_{i=1}^{n_k}\mu_{i,k}^2\lambda_{i,k} & \sum_{i=1}^{n_k}\mu_{i,k}^4 & \sum_{i=1}^{n_k}\mu_{i,k}^2\lambda_{i,k}^2 & \sum_{i=1}^{n_k}\mu_{i,k}^3\lambda_{i,k} \\ \sum_{i=1}^{n_k}\lambda_{i,k}^2 & \sum_{i=1}^{n_k}\mu_{i,k}\lambda_{i,k}^2 & \sum_{i=1}^{n_k}\lambda_{i,k}^3 & \sum_{i=1}^{n_k}\mu_{i,k}^2\lambda_{i,k}^2 & \sum_{i=1}^{n_k}\lambda_{i,k}^4 & \sum_{i=1}^{n_k}\mu_{i,k}\lambda_{i,k}^3 \\ \sum_{i=1}^{n_k}\mu_{i,k}\lambda_{i,k} & \sum_{i=1}^{n_k}\mu_{i,k}^2\lambda_{i,k} & \sum_{i=1}^{n_k}\mu_{i,k}\lambda_{i,k}^2 & \sum_{i=1}^{n_k}\mu_{i,k}^3\lambda_{i,k} & \sum_{i=1}^{n_k}\mu_{i,k}\lambda_{i,k}^3 & \sum_{i=1}^{n_k}\mu_{i,k}^2\lambda_{i,k}^2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n_k} z_{i,k} \\ \sum_{i=1}^{n_k}\mu_{i,k}z_{i,k} \\ \sum_{i=1}^{n_k}\lambda_{i,k}z_{i,k} \\ \sum_{i=1}^{n_k}\mu_{i,k}^2 z_{i,k} \\ \sum_{i=1}^{n_k}\lambda_{i,k}^2 z_{i,k} \\ \sum_{i=1}^{n_k}\mu_{i,k}\lambda_{i,k}z_{i,k} \end{pmatrix}. \tag{4.10}$$

The solution of this system exists and it is unique [28, 33]. So the solution of this linear system allows to obtain the constants and we have the approximation l_k by the least squares method of the function ϑ, $\forall k \in \mathbb{Z}_0^+$.

Note that by this technique, it's possible add, if it is necessary, more points to obtain a better fit and this issue is an advantage over the for quadratic interpolation since it only allows six points [37].

Since we already know how these polynomials are determined, we intend to characterize the minimizer of this successive models through the trust region. In this method, x_0 is the initial point and $\Delta_0 > 0$ is the initial trust region radius. So the initial trust-region is defined by the set I_0 where $x_0 \in I_{2,2,0}$ and for the other sub-sets a random point is generated. Consequently the set P_0 is defined with this points and we are able to calculate the approximation l_0. For the iteration k let us consider the points of P_k to obtain the k-th approximation l_k by the least squares method. To achieve the minimizer, we will obtain the solution of the next sub-problem

$$\begin{array}{cc} \min\limits_{x\in\mathbb{R}^2} & l_k(x) \\ s.t & \|x - x_k\|_\infty \leq \Delta_k. \end{array} \tag{4.11}$$

The solution x_{k+1} will be the minimizer of l_k in the square centered in x_k and radius Δ_k. Its critical point is the solution of $\nabla l_k(\mu, \lambda) = 0$:

$$\begin{bmatrix} 2a_4 & a_6 \\ a_6 & 2a_5 \end{bmatrix} \begin{bmatrix} \mu \\ \lambda \end{bmatrix} = - \begin{bmatrix} a_2 \\ a_3 \end{bmatrix} \tag{4.12}$$

where the matrix of the system is the hessian, $\nabla^2 l_k(\mu, \lambda)$, of the quadratic model. This matrix is definite positive (DP) if the two eigenvalues are positive [38, Theorem 2.27]. The characteristic polynomial of the hessian allows us to obtain the next expressions for eigenvalues:

$$a_4 + a_5 \pm \sqrt{(a_4 - a_5)^2 + a_6^2}.$$

So this values are both positive when $4a_4a_5 - a_6^2 > 0$ and $a_4 > 0$.

We will denote by x_k^* the minimizer of (4.11) in the iteration k. If the hessian is definite positive and the solution obtained by resolution of (4.12) is in the trust region, the minimizer of (4.11) is the solution of (4.12). If one of this conditions isn't satisfied the minimizer will be determined over the boundary. As l_k is continuous in $\|x - x_k\|_\infty \leq \Delta_k$ which is closed and bounded, (i.e. compact of finite dimension), there is a minimizer (and a maximizer) by Weirstrass Theorem [39, Theorem 4.1].

To accept this minimizer, being the iteration successful, it is necessary that the reduction obtained in the objective function be at least one portion of the observed reduction in the quadratic model. In this way, we analyze the ratio:

$$\rho_k = \frac{\vartheta(x_k) - \vartheta(x_k^*)}{l_k(x_k) - l_k(x_k^*)}, \tag{4.13}$$

and we compare it with the value $\eta \in]0, 1[$ initially fixed. If $\rho_k \geq \eta$, the quadratic model fits well to the objective function and the iteration k is successful; otherwise, the trust region is too large and the fit between the function and the model isn't in accordance with what we expected. In this case, we reduce the trust region radius (maintaining the same approximation). The Algorithm 2 presents a brief description of this method.

To update P_{k+1} and I_{k+1}, independent of having success or not, we will have as trust region a new square centered on x_{k+1} with radius Δ_{k+1} and the set I_{k+1} will be given by the trust region. As x_{k+1} is the centre of these region, $x_{k+1} \in I_{2,2,k+1}$ and if there are points of I_k that belong to some $I_{i,j,k+1}$, $(i, j) \in \{1, 2, 3\} \neq (2, 2)$, we will keep those points; otherwise a new point is generated for each sub-set. P_{k+1} will be the set with the new n_{k+1} points obtained by this process and a new quadratic model is obtained by the least squares method. For the next iterations repeat this procedure as it is briefly described in Algorithm 2.

Next we will discuss which are the sufficient conditions to have global convergence of this method based on the work presented in [28].

Let's start by the properties of the quadratic model. For the iteration k, we will consider $\Lambda_k = \max_{1 \leq i,j \leq n_k} \left\| (\mu_{i,k}, \lambda_{j,k}) - x_k \right\|_2$ where these points belong to P_k and they were generated following an uniform random distribution on each sub-interval except the one in $I_{2,2,k}$. The first condition assumes that function ϑ is continuously differentiable in an open domain A containing the ball $B(x_k, \Lambda_k) = \left\{ x \in \mathbb{R}^2 : \|x - x_k\|_2 \leq \Lambda_k \right\}$ and $\nabla^2 \vartheta$ is Lipschitz continuous with constant $\delta_1 > 0$ in A. This result

Algorithm 2: Derivative free trust-region method

```
Initialization Choose $x_0$, $\Delta_0 > 0$ and the constant $\eta \in ]0,1[$ with $\vartheta(x_0) < \infty$. Obtain $I_0$ and
  consequently $P_0$.
for $k = 0,1,2,...$ do
    Construct the model for $l_k(\mu,\lambda)$ by least squares method in (4.10) with the points of $P_k$.
    Obtain the critical point by resolution of the problem (4.12).
    if $\nabla^2 l_k(\mu,\lambda)$ is DP and satisfies the constrain of the sub-problem (4.11) then
        $x_k^*$ is the minimizer;
    else
        Obtain the minimizer $x_k^*$ over the boundary of the square centered in $x_k$ and radius $\Delta_k$.
    Calculate $\rho_k$ in (4.13).
    if $\rho_k \geq \eta$ then
        $\Delta_{k+1} = \Delta_k$;
        $x_{k+1} = x_k^*$;
    else
        $\Delta_{k+1} = \Delta_k/2$;
        $x_{k+1} = x_k^*$;
    Update $I_{k+1}$, $P_{k+1}$ for the new points and obtain $l_{k+1}$.
```

allows us to have a ball over the successive points of P_k so if we consider Λ_k greater than the trust-region, the ball will cover the respective square. If it isn't possible to satisfies such condition, we will generate a new point in order to ensure that it is satisfied.

Then we will present the conditions to prove the global convergence. The function ϑ should be smooth in $L(x_0)$ where this set is the same as defined in Hypothesis1 with $n = 2$. In the trust-region method it's assumed that the hessian of the quadratic model is uniformly bounded.

Hypothesis 3. There exist a positive constant C, such that, for all iterations x_k verifies

$$||\nabla^2 l_k(x_k)|| \leq C.$$

Hypothesis 4. The function ϑ is continuously differentiable and Hypothesis 2 holds for an open domain containing the following set

$$\bigcup_{x \in L(x_0)} B(x, \Delta_{max}),$$

where $\Delta_{max} \geq \Delta_k$, $\forall k \geq 0$ and x_0 are known.

The following theorem shows that all sequences of gradients of ϑ converge to zero.

Theorem 4. The Hypothesis 1,3 and 4 allow to conclude [28] that

$$\lim_{k \to \infty} ||\nabla \vartheta(x_k)|| = 0.$$

Given the expression of l, we can't verify whether the hypotheses of Theorem 4 are valid in our problem, but the results in Section 4.3 appear to indicate that the method converges.

In order to clarify how the Algorithm 2 works, we will present a numerical example. Let's consider $x_0 = (11,11)$, $\Delta_0 = 0.1$ and $\eta = 0.1$. Then, we generate the points of P_0 in order to build l_0, where

$$l_0(\mu,\lambda) = \left(5855 - 997\mu - 227\lambda + 43\mu^2 + 2.53\lambda^2 + 19\mu\lambda\right) \times 10^{-4}.$$

The critical point of l_0 has both eigenvalues positive but since it doesn't belong to the square we will calculate the minimizer over the boundary which is $(10.9,10.9)$. Since $\rho_0 = 0.99 \geq \eta$ we have success in first iteration and this means that the quadratic model is a good approximation of the objective function in the trust region. Thus we obtain the solution x_1 and the radius of the trust region is kept. The next nine approximations were obtained over the boundary where all of them we had success. The last approximation is $x_{10} = (10,10.0005)$ which it has an error smaller than 0.1.

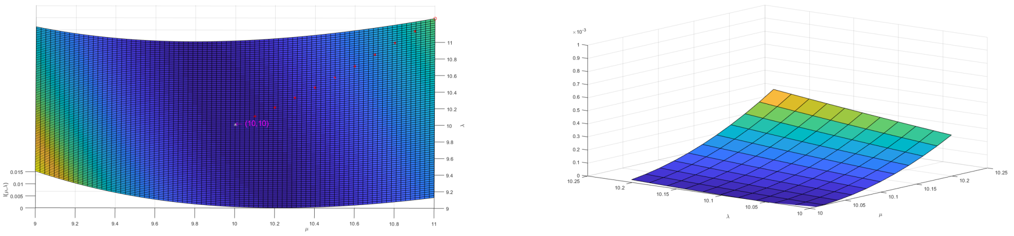

Fig. 4.6 The graph shows the objective function in I where the several approximations are obtained by the Algorithm 2 starting with $x_0 = (11,11)$ and $\Delta_0 = 0.1$ (left). It is also presented the last quadratic model obtained by the least squares (right).

4.2.3 Steepest-descent method

In this section we will apply an optimization method with the derivatives obtained in Section 4.1, called the steepest-descent method also known as the gradient method. It is a line search method where the direction of this method is the one where the function l decrease more quickly and requires only the calculation of the first derivatives. However, it is a slow method in difficult problems and doesn't provide large reductions of function value due to be sensitive to low scales [25].

Let's assume that the function $\vartheta(x)$, $x \in \mathbb{R}^n$, is continuously differentiable. In the steepest-descent method the iterations are given by

$$x_{k+1} = x_k + \alpha_k p_k \tag{4.14}$$

where the descent direction p_k is $-\nabla\vartheta(x_k)$ and α_k is the length of the step in such direction. The step length can be found in an exact form through the next optimization problem of one variable

$$\min_{\alpha>0} \vartheta(x_k + \alpha p_k). \tag{4.15}$$

However, it is expensive to solve (4.15) and it isn't necessary to calculate α_k in the way to obtain global convergence. Usually, it is sufficient to determine α_k in an approximate form satisfying the Wolfe's conditions [40]: given $c_1, c_2 \in]0,1[$ with $c_1 < c_2$, $\vartheta(x_k + \alpha p_k) \leq \vartheta(x_k) + c_1\alpha(\nabla\vartheta(x_k))^T p_k$ (sufficient decrease condition) and $(\nabla\vartheta(x_k + \alpha p_k))^T p_k \geq c_2(\nabla\vartheta(x_k))^T p_k$ (curvature condition) which allows to obtain global convergence of the method. The first condition provides a decrease in ϑ proportional to α while the second avoids very small steps. The last condition can be replaced by the backtracking technique without losing the convergence properties [25]. The idea is to start by considering a given value $\bar{a} > 0$. The backtracking procedure is to find a $\alpha = \bar{a}$ that satisfies the sufficient decrease condition and in the case that $\bar{a}$ doesn't satisfy it, it is reduced by one factor until the condition is satisfied. The first value of $\bar{a}$ that algorithm verifies, it is the value consider for α_k. This procedure to determine α_k is given by Algorithm 3 while Algorithm 4 gives us the sketch of the steepest-descent method.

```
Algorithm 3: Backtracking for steepest-descent method
  Initialization Choose $\bar{a} > 0$, $c_1 \in ]0,1[$ and let $\alpha = \bar{a}$.
  while $\vartheta(x_k - \alpha \nabla \vartheta(x_k)) > \vartheta(x_k) - c_1 \alpha \|\nabla \vartheta(x_k)\|_2^2$ do
      $\alpha = \alpha/2$;
  Terminate $\alpha_k = \alpha$.
```

```
Algorithm 4: Steepest-descent method
  Initialization Choose the initial point $x_0$ where $\vartheta(x_0) < \infty$.
  for $k = 0, 1, 2, \ldots$ do
      Step 1: Calculate the direction $\nabla \vartheta(x_k)$;
      Step 2: Find the length step $\alpha_k$ by the Algorithm 3 ;
      Step 3: Update $x_{k+1} = x_k - \alpha_k \nabla \vartheta(x_k)$;
```

Then we will present the results that guarantee the global convergence of steepest-method [25, 41].

Lemma 5. Let any iteration of the form (4.14) be obtained with the steepest-descent method. Suppose that ϑ is bounded below in $\mathbb{R}^n$ and that it is continuously differentiable in an open domain A containing the set defined in Hypothesis 1. Assuming also Hypothesis 2 then the following expression holds:

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\nabla \vartheta(x_k)\|^2 < \infty.$$

Theorem 5. Assuming the conditions in Lemma 5 and that $\cos \theta_k \geq \gamma > 0$ for all k, then [25]

$$\lim_{k \to \infty} ||\nabla \vartheta(x_k)|| = 0.$$

It can be proved also that under Hypothesis 1 and 2, when the Algorithm 4 is used the gradient decays at a sublinear rate $1/\sqrt{k}$ [28]. If the function is convex or strongly convex the global rate is the same as for the coordinated search method [29].

Since some hypothesis are the same as those presented for the trust region method, we don't know if the function l verifies these conditions. However, the numerical results presented in Section 4.3 seem exhibit the convergence of the method.

We will present a simple example in order to show how Algorithm 4 works. We will consider the initial point $x_0 = (11, 11)$, $\bar{a} = 25$ and $c_1 = 0.01$. Initially the algorithm determines the gradient for the initial point that is given by $(-0.0144, -0.0033)$. With this descent direction and with the step length

$\alpha_0 = 25$ we can decrease the value of the function obtaining the approximation $x_1 = (10.641, 10.916)$. After a some iterations we obtained the approximation $x_{226} = (9.979, 10.097)$ with an error smaller than 0.1.

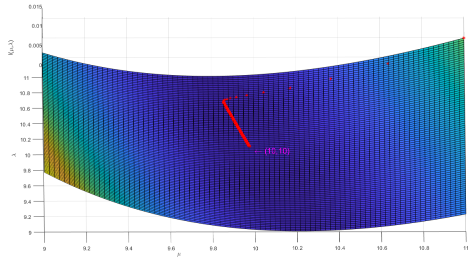

Fig. 4.7 Sucessive approximations obtained by Algorithm 4 starting from $x_0 = (11, 11)$ where $\bar{a} = 25$ and $c_1 = 0.01$. The trajectory is described on the surface of the objective function l defined in $[9, 11]^2$.

4.3 Computational study

In this section we will present a comparative study of the optimization methods described in Section 4.2 applied to the problem analyzed in this work.

In the context of the ElastoOCT project we still have no experimental data that can be used in this work. Thus, in the tests performed we will use as observed displacements the vector U_{obs} which corresponds to the solution obtained by the direct method with $(\mu_{obs}, \lambda_{obs}) = (10, 10)$. In this way, the vector $(\mu_{obs}, \lambda_{obs})$ will be the optimal solution of the inverse problem that we want to approximate. In the context of the project, the experimental data should contain noise so in this section are considered two scenarios: noise free data (Section 4.3.1) and noisy data (Section 4.3.2). In the first scenario, it is intended to identify the method with the best performance, which will be used in the second scenario to assess its sensitivity to the noise level considered. This study will allow to evaluate the applicability of the proposed solution in a real scenario in the future.

For both scenarios we consider the same domain $\Omega = [-2, 2]^3$ and the same boundary conditions presented in previous section. The density of the material is taken to be $\rho = 1 g/cm^3$, the angular frequency is $w = 2$ rad/s and the force function $\underset{\sim}{g}$ is defined by $g_i = 5.86 \times 10^{-3}, \forall i \in 1, 2, 3$. For the optimization problem we will choose the set $I = [5, 100]^2$. The surface of objective function l in I is presented in Figure 4.8.

The performance of the methods will be analyzed in term of the following parameters: the number of iterations, the number of times the objective function is calculated, the absolute error, the function value and the gradient norm. We will do a statistical study where we perform 30 simulations with different starting points and we evaluate the mean, the standard deviation (SD) and the maximum value for the number of iterations. We also present the percentage of successful iterations (% Suc) until the solution converges considering the stop condition. For the second parameter described we will analyze the mean and standard deviation, however, for the remaining parameters we study the behavior from the respectively graphs. In each simulation, the three methods use the same starting point.

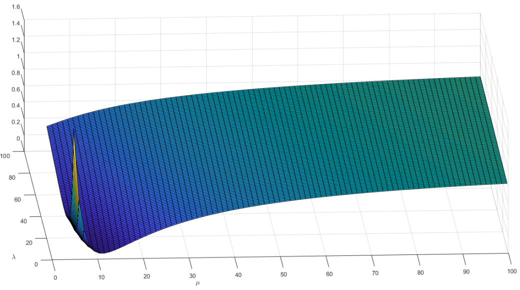

Fig. 4.8 Surface of the objective function l in $[5,100]^2$.

4.3.1 Noise free data

In this section we aim to compare the performance of the three methods described in Section 4.2. We want to verify which of these methods is the most efficient with the goal of choosing the best optimization method to be applied in the problems with noisy data.

For this scenario, since we know that the optimal solution is $(10,10)$, we will consider the stop condition for the three algorithms given by $||(10,10)-x_k||_2 < 0.1$.

Coordinated search method

Since we have the same starting points, the choice of the initial step length has to be consistent since it should be the same value for the three methods. In this method we have chosen $\alpha_0 = 10$, since it is possible to pass through all regions of the set I with few iterations.

For this method we will present a study with the PSS D and D_1 defined in Section 4.2.1 considering the force function $\rho(t) = t^2/5000$. We intend to compare which of this PSS produces better approximations with the Algorithm 1.

Given the domain configuration and the directions considered in D_1, there are situations where it may not be possible to find a descent direction in D_1. For example, if in iteration k we obtain the point $x_k = (5,100)$, only the vector $(1,-1)^T$ of D_1 allows us to keep the next points in I but this direction may be an ascent direction. To overcome this situation, we defined the PSS $D_2 = D_1 \cup D$ where we kept the elements in same order as in the respective sets and the directions of D are checked only if none of the directions in D_1 generate a successful iteration.

As the distribution looks roughly symmetrical (see boxplot in Figure 4.9) and there aren't many outliers, we decided to analyze the mean and standard deviation for the number of iterations and the number of evaluations of the objective function.

In Figure 4.9, we show the boxplot of the number of iterations (left), the average value of the step length for iteration and the average value of the absolute error, using the euclidean norm, for each iteration (right), considering 30 simulations with different starting points.

Figure 4.9 and Table 4.1 present a summary of the results obtained with the 30 simulations. From these results we can conclude that the use of PSS D_2 is, in general, more efficient in the tests performed. In fact, in terms of the number of iterations to achieve the desired precision, the mean and variability is lower when PSS D_2 is used.

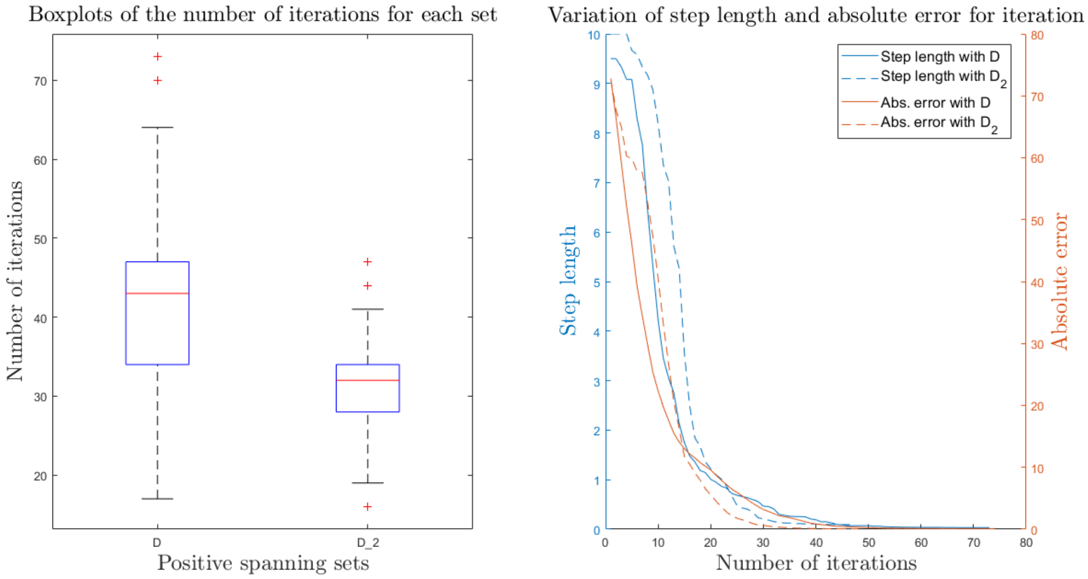

Fig. 4.9 Boxplot of the number of iterations (left); Average values of the step length for each iteration (blue lines) and the variation on average of the absolute error (red lines) measure by euclidean norm in terms of the number of iterations (red lines) corresponding different PSS. The continuous lines and dashed lines correspond to PSS D and D_2, respectively. The values correspond to the mean of 30 simulations.

This decrease in the number of iterations is achieved with an increase in the number of times that the objective function is evaluated. Taking into account the values in Table 4.1, we observe that the average number of times the objective function is evaluated for each iteration is 1.6 in the case of the PSS D and approximately 2.7 in the case of set D_2. Generally, we can then conclude that, on average, in the first case, it uses only the first two vectors of D while in the second case three vectors of D_1 are used. Relatively to the length of the step, it is observed that the use of the set D_2 allows to maintain higher values in the first iterations, since the absolute error is higher, causing the algorithm to approximate the exact solution more quickly (resulting in a reduction in the number of iterations already indicated). When increasing the number of iterations, the PSS D_2 (blue dash lines) obtained on average a value of the step length closer to zero more quickly than when the PSS D is used.. This figure is also in agreement with equation (4.7) which guarantees the existence of a sequence of iterations where the step length tends to zero. We also report the percentage of successful iterations in both cases is similar (Table 4.1) which corresponds to the portion of successful iterations among the total iterations performed until the stop condition is satisfied. From the analysis of this table and Figure 4.9 (right), we conclude that unsuccessful iterations should appear earlier when using the set D, which would be expected because it has less elements, leading to a faster decrease of the step length.

Since the objective function l isn't negative, an indicator of the convergence of successive approximations can be the value of the objective function. In Figure 4.10 we present the variation of the function value for both PSS. We can observe that the behavior of this graph is similar to the variation of step length and absolute error in Figure 4.9 (right) since the decrease in the value of the step length and the absolute error until reaching the optimal solution implies a decrease in the value of the function, which was expected.

We include the average of the gradient norm in Figure 4.10 to verify if, numerically, the average of the 30 simulations we have the convergence to the stationary point stated in Theorem 3. Analyzing this parameter in both PSS, they have a different behavior than we expected since initially the curves increase a little bit. This behavior can be explained by using the surface of the objective function in Figure 4.8. Note that the 30 starting points are random and it is more probable that they belong to set $[40,100]\times[5,100]\subset I$ where the objective function is characterized by having a little slope corresponding to where the norm of the gradient has small values. The algorithm in a few iterations will have approximations close to the region where the function changes its concavity (see Figure 4.8), making that the gradient will gradually increase since the slope is bigger. However it quickly becomes smaller when we are already very close to the optimal solution and converges to a stationary point.

In Figure 4.10 we can conclude that since the PSS D presents higher decreases of the initial function values, the gradient for this PSS is slightly higher when increasing the number of iterations. However, after a few iterations, the PSS D_2 presents higher decreases of the value of function more quickly so in this way as it preforms fewer iterations until it converges, the gradient norm in this PSS tends more quickly to zero as we can see in Figure 4.10.

	Number of iterations				Number of evaluation of l	
PSS/Data	mean	*SD*	maximum	% Suc	mean	*SD*
D	42.87	13.96	73	78%	69.15	15.6
D_2	31.32	7.38	47	75.7%	83.9	14.86

Table 4.1 Statistical summary of the computational results for the coordinated search method.

Derivative free trust-region method

First we will discuss a reasonable value for the initial radius. In Algorithm 1, the directions of the PSS D and D_2 have vectors with unit norms and $\alpha_0 = 10$, making $||x_1 - x_0||_2 \leq 10$.

To be in the same condition when using Algorithm 2, we will use $\Delta_0 = 5\sqrt{2}$ since the trust region is a neighborhood of x_0 with radius Δ_0 defined with the infinite norm, that is, a square with side $10\sqrt{2}$ centered on x_0.

We will consider the same aspects as before to evaluate the performance of this optimization method. In Figure 4.11 we present the boxplot for the number of iterations, a graph with the average value of the trust region radius and the absolute error for each iteration. The statistical summary is presented in Table 4.2.

From the results presented, we observed that this method needs fewer iterations than the coordinated search method to achieve the desired precision, but evaluates the objective function at more points (approximately four evaluations of the function for each iteration). From Figure 4.11 (right), we see in general a significant decrease of the average of trust region radius and the absolute error. The evolution of the trust region radius has three "phases". At first (up to iteration 7) maintains a high radius value allowing absolute error to decrease quickly; in the second (until iteration 20) the radius

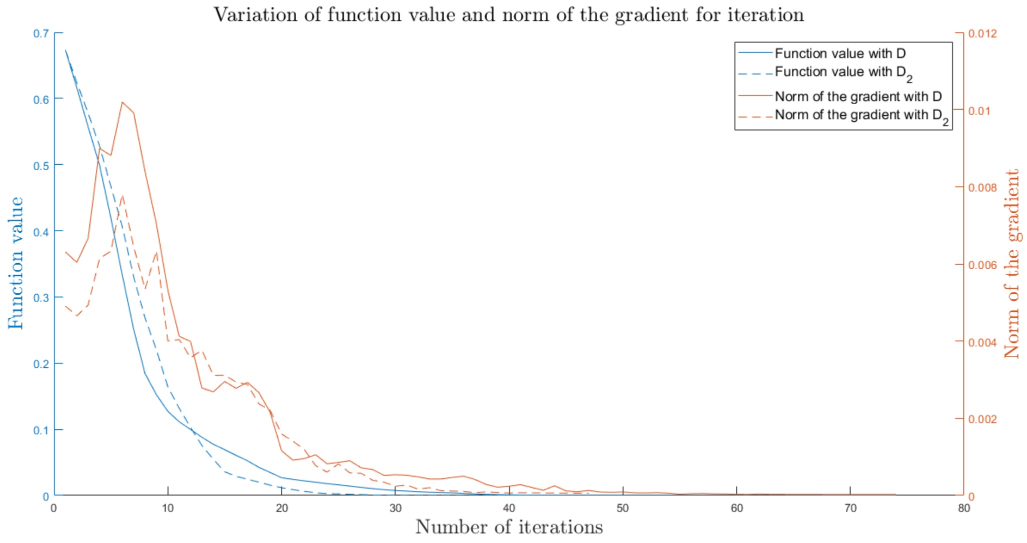

Fig. 4.10 Variation on the average, for each iteration, of the function value (blue lines) and the gradient norm (red lines) for each PSS. The continuous lines and dashed lines correspond to the PSS D and D_2, respectively.

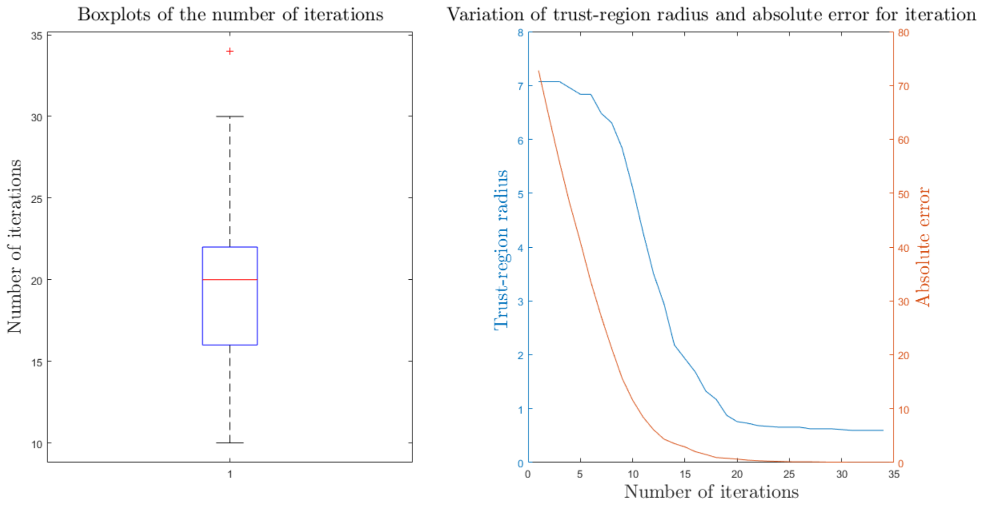

Fig. 4.11 Boxplot of the number of iterations (left); Average values of the trust-region radius (blue) and the absolute error measure by euclidean norm in terms for each iteration (right).

decreases quickly (should be when it gets the iterations without success) and there is a slowdown in the decrease in the error; in the latter (after iteration 20) the two lines stabilize (this happens because most simulations have already converged). We can notice that the percentage of successful iterations for each solution compared to the coordinated search method is slightly higher, which was expected since this method performs less iterations until it satisfies the stopping condition. All the simulations converge to the optimal solution in less than 35 iterations.

By observing Figure 4.11 (right, blue line) and Figure 4.12, we notice that the behavior of the absolute error, the norm of the gradient and value of the function are similar to the previous method. However, these metrics decreased quickly in the trust region method because it needed fewer iterations until converging to the optimal solution.

	Number of iterations				Number of evaluation of l	
Data	mean	*SD*	maximum	% Suc	mean	*SD*
Values	19.6	5.02	34	80.2%	79.5	16.85

Table 4.2 Statistical summary of the computational results for the derivative free trust-region method.

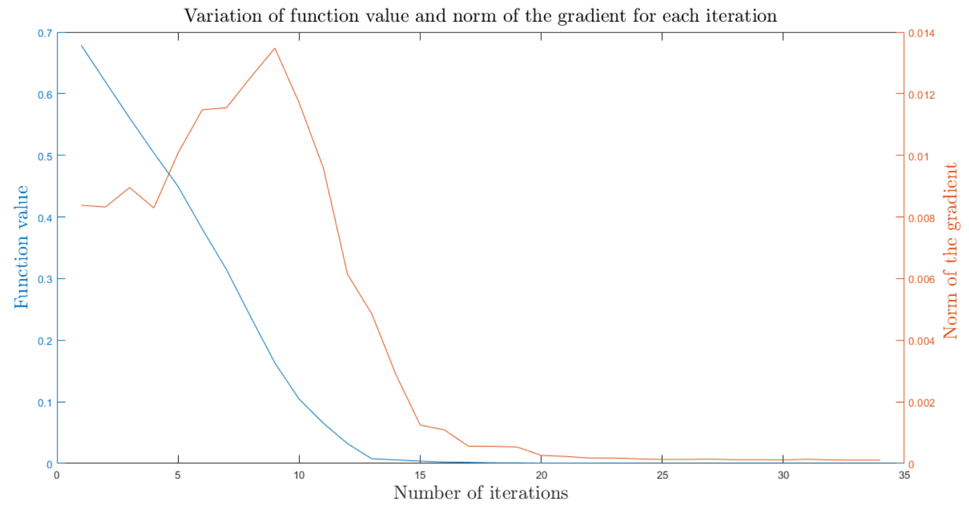

Fig. 4.12 Variation on average of the function value (blue line) and the gradient norm (red line) with the number of iterations.

Steepest-descent method

We will present a way to obtain the step length $\alpha_0 \in]0,\bar{a}]$ for this method in order to be comparable with the other methods. We want the best effective distance through the direction $\nabla l(x_0)$ using a reasonable value for $\bar{a}$. The effective distance, $\bar{a}||\nabla l(x_0)||_2$ must be equal to 10 to be in concordance with the other methods. One approximation for the initial step is $\bar{a} = 10/||\nabla l(x_0)||_2$ and it depends of the initial approximation x_0 which is different of others methods. We will also consider that $c_1 = 0.01$ as described in Section 4.2.3.

In Figure 4.13, we show the boxplot of the number of iterations (left), the average value of the step and the absolute error for each iteration (right). Observing the results of Figure 4.13 and Table 4.3 we conclude that it is necessary, on average, more iterations to reach a solution close to the optimal solution and the data show greater variability in the number of iterations compared to what we obtained in the other two methods. This method determines the step length using the backtracking technique, needing to compute the objective function on average four times for iteration.

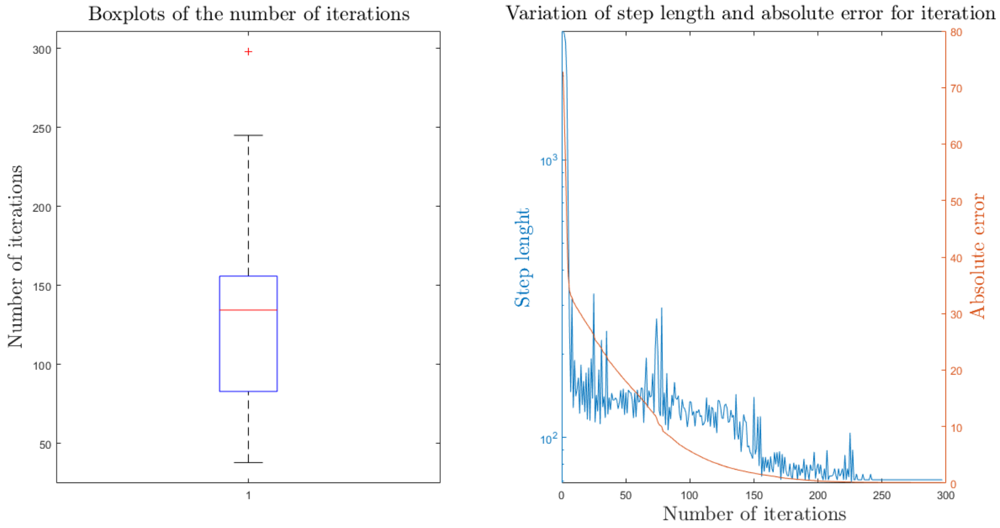

Fig. 4.13 Boxplot of the number of iterations (left); Average values of the step length for each solution (blue line). The line in red represents the variation on average of the absolute error measure by euclidean norm of the number of iterations for the 30 simulations (right).

By Figure 4.13 (right) we can conclude that with the decrease in absolute error (red line), the step length (blue line) tends to zero, which allow to guarantee the convergence of the approximations to a stationary point.

	Number of iterations			Number of evaluation of l	
Data	mean	*SD*	maximum	mean	*SD*
Values	133.6	57.9	298	534.7	264

Table 4.3 Statistical summary of the computational results for the steepest-descent method.

We present in Figure 4.14 a graph with the average value of the Euclidean norm of the gradient and function value for each iteration. We can observe that the behaviour of these graphs are similar to that obtained by the other methods. However we can see that the peak of the value of the norm of the gradient (red lines) is higher than the other methods. In what concerns to the behavior of the value of the objective function, it is similar to the previous method. However their decrease of the objective function was slower since the method need more iterations until converging to the optimal solution. We can conclude that the initial points of our simulations converge to the optimal solution since, on average, the norm of the gradient of them tends to zero (Theorem 5).

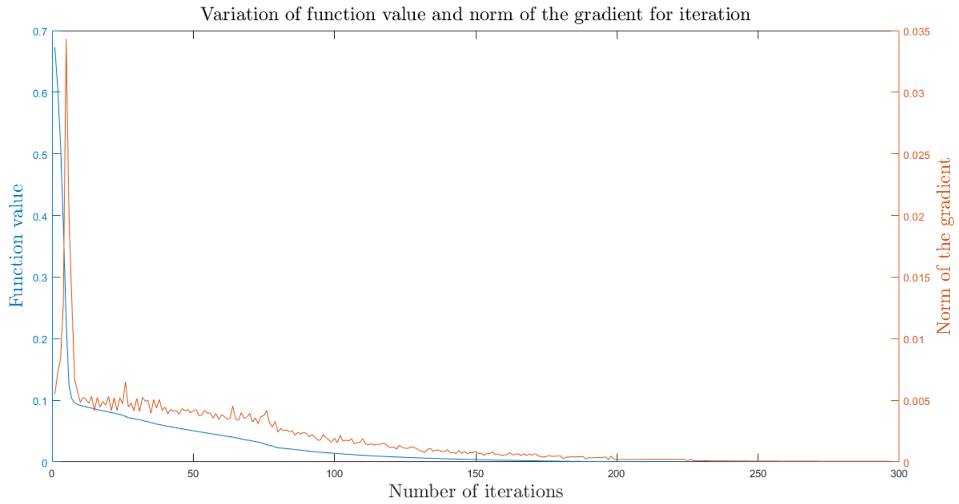

Fig. 4.14 Variation on average of the function value (blue line) and the gradient norm (red line) with the number of iterations.

In this simulations we consider that derivative free trust-region method is the most efficient since it converges very fast to an approximation close to the optimal solution in a few iterations. This method isn't very restrictive when determines a new approximation since it is the minimizer in the trust region and allow to obtain descent directions (advantage). In simulations, the coordinated search method is exhaustive since Algorithm 1 finds the first direction of the PSS that satisfies (4.6) around x_k. It only uses the knowledge of l on a discrete set of points, having no general information about what happens in the neighborhood of x_k (disadvantage), where it can be possible to decrease the value of the objective function even further. In the trust region method, the quadratic model that best fits the objective function gives an idea of the objection function in a whole neighborhood at each iteration. in each iteration. The steepest-descent method in our simulations have on average very low values of the gradient norm and consequently the value of the gradient. Despite the gradient is the direction from where the function decrease quickly, this process takes a lot of iterations to converge (disadvantage). The next section is dedicated to a more realistic scenario which we will solve using consider the derivative free trust-region method.

4.3.2 Noisy data

In this section we intend to evaluate if it is possible to infer the optimal solution with noisy displacements. To put this idea in practice, we will simulate observed displacements by the direct problem where this solution satisfies the system (3.12) and then we will introduce a gaussian noise $R \sim N(0,\sigma)$ where R is a vector of dimension $3N \times 1$ and σ is the standard deviation. So in noisy situations instead of displacements U_{obs}, we will consider as data $\bar{U}_{obs} = (R + 1_{3N\times 1})U_{obs}$, where $1_{3N\times 1}$ is a $3N \times 1$ vector with all components equal to one and the i-th component of the vector $\bar{U}_{obs}$ is given by $(R(i)+1)U_{obs}(i)$, $i \in 1,\ldots,3N$. The value of the standard deviation σ can't be very high, otherwise the disturbed values will be very different from the initial values and it isn't possible to recover the optimal solution. For this reason, we present a study about which variations the values of standard deviation could have.

Application

We will present a study to discuss the sensitivity of the data disturbed by noise. We will consider ten equally spaced variations of σ in the interval $[0,0.02]$ where for each value of it, we will consider simulations with 10 random initial points. We intend to evaluate the absolute error, using the euclidean norm, between the solution $(10,10)$ and the approximation obtained by the derivative free trust-region method.

In this context we can't impose the same stopping condition of the previous section because we don't know if the optimal solution $(10,10)$ disturbed with noise will converges to the optimal solution. We will consider 34 iterations to run the algorithm for each initial solution since in previous section the derivative free trust-region method needed at most this number of iterations to converge considering the stop condition. In Figure 4.15 we present on average of the variation of the absolute error with the variation of the σ for ten simulations.

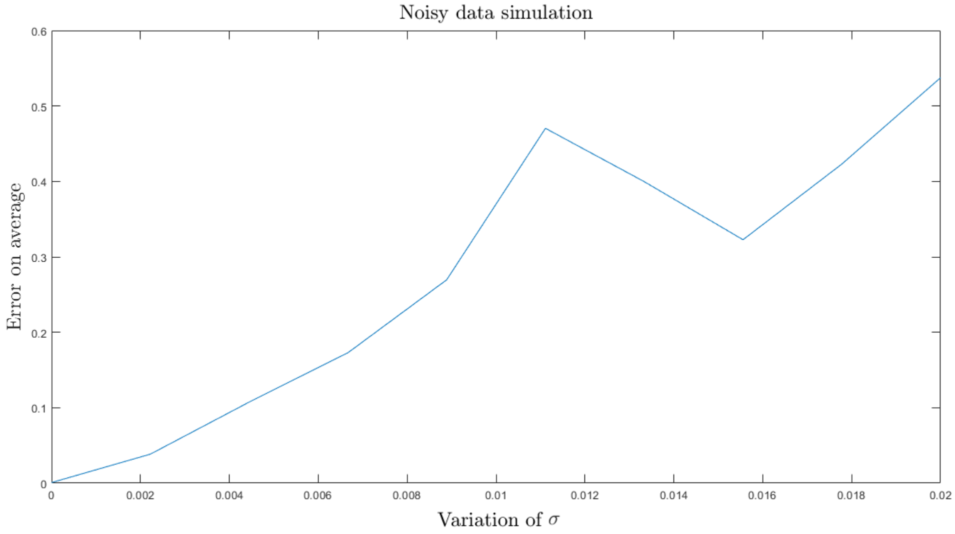

Fig. 4.15 Average of the euclidean norm of the absolute error obtained from ten simulations for each value of σ.

We can conclude that all simulations without noise, $\sigma = 0$, arrived at the optimal solution. Generally increasing the value of σ has the effect of increasing the error except when σ belongs to the interval $[0.011,0.015]$. We weren't expecting this situation, however, the same situation happened in other tests we made. Note that for this range of values of σ, the error is quite big so the approximation is far from the optimal solution.

Generally, from Figure 4.15 we can observe that we obtain a good approximation for the Lamé constants for values of $\sigma \in [0,0.004]$ with absolute error less than 0.1 between the optimal solution and the approximate solution by this optimization method.

Our simulations correspond to the angular frequency, the density material, an the function $\underset{\sim}{g}$ defined in Section 3.3. From the data considered and smaller values of σ ($\sigma \leq 0.004$) was possible to recover the Lamé constants (with the norm of the absolute error less than 0.1).

References

[1] K. S. Sheinerman and S. R. Umansky, Circulating cell-free microRNA as biomarkers for screening, diagnosis and monitoring of neurodegenerative diseases and other neurologic pathologies. Front. Cell. Neurosci, vol. 7, 2013.

[2] D. Claus, M. Mlikota, J. Geibel, T. Reichenbach, G. Pedrini, J. Mischinger, S. Schmauder, and W. Osten. Large-field-of-view optical elastography using digital image correlation for biological soft tissue investigation. Journal of Medical Imaging, 4 (1): 1–14, 2017.

[3] B. F. Kennedy, X. Liang, S. G. Adie, D. K. Gerstmann, B. C. Quirk, S. A. Boppart, and D. D. Sampson. In vivo three-dimensional optical coherence elastography. Opt. Express, 19 (7):6623–6634, 2011.

[4] Y. Qu, Y. He, Y. Zhang, T. Ma, J. Zhu, Y. Miao, C. Dai, M. Humayun, Q. Zhou, and Z. Chen. Quantified elasticity mapping of retinal layers using synchronized acoustic radiation force optical coherence elastography. Biomed.Opt.Express, 9 (9):4054–4063, 2018.

[5] J. Zhu, Y. Miao, L. Qi, Y. Qu, Y. He, Q. Yang, and Z. Chen. Longitudinal shear wave imaging for elasticity mapping using optical coherence elastography. Applied Physics Letters, 110 (20):201101, 2017.

[6] A. M. Morgado, S. Barbeiro, R. Bernardes, J. M. Cardoso, J. Domingues, C. Loureiro, M. Santos, and P. Serranho. Optical coherence elastography for imaging retina mechanical properties, FCT project. http://miguelmorgado.net/research/projects/on-going/elastooct.html.

[7] Sushma Santapuri, Stephen E. Bechtel. Linear Momentum in Fundamentals of Continuum Mechanics. ScienceDirect, 2015.

[8] Richard Fitzpatrick. Theoretical Fluid Mechanics, 2017.

[9] U. Saravanan. Advanced Solid Mechanics. Traction and Stress, 2013.

[10] Susane C. Brenner, L. Ridgway Scott. The Mathematical Theory of Finite Element Methods. Springer, 1997.

[11] Todd Arbogast and Jerry L. Bona. Methods of Applied Mathematics. Department of Mathematics, and Institute for Computational Engineering and Sciences The University of Texas at Austin, 2008.

[12] M. Paula Serra de Oliveira, Cândida de Oliveira Pereira. Análise Infinitesinal em $\mathbb{R}^N$. Coimbra, 2017.

[13] Erich Miersemann. Partial Differential Equations. Lecture Notes, 2012.

[14] M. M. Doyley. Model-based elastography: a survey of approaches to the inverse elasticity problem. Physics in Medicine and Biology, 57 (3):R35–R73, 2012.

[15] Frank Ihlenburg. Finite Element Analysis of Acoustic Scattering. Springer, 1998.

[16] W. Hackbusch. Elliptic Differential Equations. Theory and Numerical Treatment, 1992.

[17] Lie-heng Wang. On Korn inequality. Academic of Mathematics and System Sciences, 2003.

[18] Endre Süli. Lecture Notes on Finite Element Methods for Partial Differential Equations. Mathematical Institute, University of Oxford, 2012.

[19] J. Alberty, C. Cartensen, S. A. Funken, R. Klose, Kiel. Matlab Implementation of the Finite Element Method in Elasticity. Computing, 2002.

[20] S. Barbeiro, P. Serranho. The method of fundamental solutions for the direct elastography problem in the human retina. Proceedings of the 9th Conference on Trefftz Methods and 5th Conference on Method of Fundamental Solutions, Springer, 2020.

[21] I. L. Jones, M. Warner, J. D. Stevens. Mathematical modelling of the elastic properties of retina: A determination of young's modulus. Eye, 6(15):556–559, 1992.

[22] Kaare Brandt Petersen, Michael Syskind Pedersen. The Matrix Cookbook, 2012.

[23] Randal J. Barnes. Matrix Differentiation, Department of Civil Engineering, University of Minnesota Minneapolis, Minnesota, USA. Spring, 2006.

[24] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. SIAM Rev., 45:385–482, 2003.

[25] J. Nocedal and S. J. Wright. Numerical Optimization. Berlin, Springer, segunda edição, 2006.

[26] A. L. Custódio, K. Scheinberg, and L. N. Vicente. Methodologies and Software for Derivative-free Optimization, 2017.

[27] R. Garmanjani, L. N. Vicente. Smoothing and Worst-Case Complexity for Direct-Search, Methods in Nonsmooth Optimization, 2012.

[28] Diogo Júdice. Trust-Region Methods without using Derivatives: Worst-Case Complexity and the Non-Smooth Case. University of Coimbra, 2015.

[29] Nesterov, Y. Introductory Lectures on Convex Optimization. Kluwer Academic Publishers, Dordrecht, 2004.

[30] A. R. Conn, Ph. L. Toint. An algorithm using quadratic interpolation for unconstrained derivative free optimization, 1995.

[31] Conn, A. R., Scheinberg, K., and Vicente, L. N. Global convergence of general derivative-free trust-region algorithms to first and second order critical points. SIAM J. Optim., 20:387–415, 2009.

[32] Conn, A. R., Scheinberg, K., and Vicente, L. N. Introduction to Derivative-Free Optimization. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[33] Andrew R. Conn, Katya Scheinberg, and Luís N.Vicente. Geometry of sample sets in derivative-free optimization: polynomial regression and undertermined interpolation, 2008.

[34] C. Audet and J. E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim., 17:188–217, 2006.

[35] G. Liuzzi, S. Lucidi, F. Rinaldi, L. N. Vicente. Trust-region methods for the derivative-free optimization of nonsmooth black-box functions, 2019.

[36] N. Vicente and A. L. Custódio. Analysis of direct searches for discontinuous functions. Math Program, 133:299–325, 2012.

[37] Peter-Wolfgang Gräber. Interpolation methods, Systems Analysis in Water Management, Chapter 1, Mathematical fundamentals, 2016.

[38] Helmute Graeb. Analog Design Centering and Sizing, Technische University Muenchenn, Germany, Springer, 2007.

[39] Jasbir S. Arora in Introduction to Optimum Design, Third Edition, 2012.

[40] Azam Asl, Michael L. Overton. Analysis of the Gradient Method with an Armijo-Wolfe Line Search on a Class of Nonsmooth Convex Functions, 2018.

[41] Stephen Boyd, Lieven Vandenberghe. Convex Optimization. Cambridge University Press, 2009.

Appendix A

A.1 Global matrix B^*

In this section we will calculate the matrix B^* in (3.7). Using (3.5) we will determine

$$b(\underset{\sim}{v_h},\underset{\sim}{v_h}) = \int_\Omega 2\mu \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) + \lambda \left(\mathrm{div}\underset{\sim}{v_h}\right)^2 dx = \sum_{K\in\Omega_h} \left(2\mu ||\underset{\approx}{\varepsilon}(\underset{\sim}{v_h})||^2_{\underset{\approx}{L}^2(K)} + \int_K \lambda \left(\mathrm{div}\underset{\sim}{v_h}\right)^2 dx\right)$$

in terms of the coefficients V_{ji}.

We will denote the base functions are linear in each tetrahedron K. Given that $\underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v_h})$ and $\left(\mathrm{div}\underset{\sim}{v_h}\right)^2$ depend on first-order derivatives so these functions are constant in each tetrahedron K. Thus, the integrals of these functions are the products of these constants by the measure of the set K which, in this case, is the volume of tetrahedron with the vertices $r_1,...,r_4$, i.e.,

$$2\mu ||\underset{\approx}{\varepsilon}(\underset{\sim}{v_h})||^2_{\underset{\approx}{L}^2(K)} + \int_K \lambda \left(\mathrm{div}\underset{\sim}{v_h}\right)^2 dx = V_{1234}\left(2\mu \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) + \lambda \left(\mathrm{div}\underset{\sim}{v_h}\right)^2\right).$$

As $|J| = 6V_{1234}$, the last expression is equal to

$$\frac{1}{6}|J|\left(2\mu \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) + \lambda \left(\mathrm{div}\underset{\sim}{v_h}\right)^2\right),$$

or equivalently,

$$\frac{1}{36V_{1234}}|J|^2\left(2\mu \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) + \lambda \left(\mathrm{div}\underset{\sim}{v_h}\right)^2\right). \tag{A.1}$$

With the purpose of obtaining a simplification of the previous expression, let's consider the following lemmas.

Lemma 6. [19] *For* $\gamma(\underset{\sim}{v_h}) = [\varepsilon_{11}(v_h)\,\varepsilon_{22}(v_h)\,\varepsilon_{33}(v_h)\,2\varepsilon_{12}(v_h)\,2\varepsilon_{13}(v_h)\,2\varepsilon_{23}(v_h)]^T$ *and* $\varepsilon_{ij}(v_h) = \frac{1}{2}\frac{\partial v_{ih}}{\partial x_j} + \frac{1}{2}\frac{\partial v_{jh}}{\partial x_i}, i,j \in \{1,2,3\}$ *holds*

$$2\mu \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) : \underset{\approx}{\varepsilon}(\underset{\sim}{v_h}) + \lambda \left(div\underset{\sim}{v_h}\right)^2 = \gamma^T(\underset{\sim}{v_h})C\gamma(\underset{\sim}{v_h}),$$

where the matrix C is given by

$$\begin{pmatrix} 2\mu+\lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & 2\mu+\lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & 2\mu+\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix}.$$

Lemma 7. *For* $V^K = [V_{11}V_{21}V_{31}V_{41}V_{12}V_{22}V_{32}V_{42}V_{13}V_{23}V_{33}V_{43}]^T$ *holds* $\gamma(\underset{\sim}{v_h}) = \frac{1}{|J|}R^kV^K$, *where* $\left(R^k\right)^T$ *is a* 12×6 *matrix given by*

$$\left(R^k\right)^T = \begin{pmatrix} -(a_{11}+a_{12}+a_{13}) & 0 & 0 & -(a_{21}+a_{22}+a_{23}) & -(a_{31}+a_{32}+a_{33}) & 0 \\ a_{11} & 0 & 0 & a_{21} & a_{31} & 0 \\ a_{12} & 0 & 0 & a_{22} & a_{32} & 0 \\ a_{13} & 0 & 0 & a_{23} & a_{33} & 0 \\ 0 & -(a_{21}+a_{22}+a_{23}) & 0 & -(a_{11}+a_{12}+a_{13}) & 0 & -(a_{31}+a_{32}+a_{33}) \\ 0 & a_{21} & 0 & a_{11} & 0 & a_{31} \\ 0 & a_{22} & 0 & a_{12} & 0 & a_{32} \\ 0 & a_{23} & 0 & a_{13} & 0 & a_{33} \\ 0 & 0 & -(a_{31}+a_{32}+a_{33}) & 0 & -(a_{11}+a_{12}+a_{13}) & -(a_{21}+a_{22}+a_{23}) \\ 0 & 0 & a_{31} & 0 & a_{11} & a_{21} \\ 0 & 0 & a_{32} & 0 & a_{12} & a_{22} \\ 0 & 0 & a_{33} & 0 & a_{13} & a_{23} \end{pmatrix}.$$

Proof. Given the expression of derivatives in (3.11) and from (3.8) and (3.9) we obtain

$$\frac{\partial v_{jh}}{\partial \xi} = V_{2j} - V_{1j}, \quad \frac{\partial v_{jh}}{\partial \eta} = V_{3j} - V_{1j}, \quad \frac{\partial v_{jh}}{\partial \tau} = V_{4j} - V_{1j}, \; j = 1,2,3,$$

which proves the lemma. □

From lemmas 6 and 7, we obtain (A.1) in the equivalent form

$$\frac{1}{36V_{1234}}|J|^2\gamma^T(\underset{\sim}{v_h})C\gamma(\underset{\sim}{v_h}) = \frac{1}{36V_{1234}}\left(V^K\right)^T\left(R^k\right)^T CR^kV^K = \left(V^K\right)^T B^kV^K, \tag{A.2}$$

where

$$B^k = \frac{1}{36V_{1234}}\left(R^k\right)^T CR^k.$$

Matrix B^k is the local matrix corresponding to the tetrahedron of vertices $r_1,\ldots,r_4$. We will construct B^* from these M local matrices. For the k-th tetrahedron K, let's consider an application L^k that associates the vertices of this tetrahedron with all the vertices of the set Ω_h. We can defined a matrix $N \times 4$ with entries zeros and ones. The number of columns is the number of vertices of the tetrahedron. We will describe the j-th column of L^k, $j \in \{1,\ldots,4\}$. For matrix B^k, if the vertex with position r_j is the i-th vertex in global numbering, $i \in \{1,\ldots,N\}$ so the j-th column of L^k has unit entrie in i-th row, $j \in \{1,\ldots,4\}$. As we are working in $\mathbb{R}^3$ the idea is to do the same for the three components. It should be noted that

$$V^K = \begin{pmatrix} \left(L^k\right)^T & 0 & 0 \\ 0 & \left(L^k\right)^T & 0 \\ 0 & 0 & \left(L^k\right)^T \end{pmatrix} V = \begin{pmatrix} L^k & 0 & 0 \\ 0 & L^k & 0 \\ 0 & 0 & L^k \end{pmatrix}^T V = \left(L_1^k\right)^T V, \tag{A.3}$$

where L_1^k is a block matrix, where each block diagonal is the matrix L^k. With this relationship is possible to pass from $v_{ih}(r_j)\,, i = 1,...3,\ j = 1,...,4$, which are the values of the function $\underset{\sim}{v_h}$ in the local matrix vertices, to $v_{ih}(r_j)\,, i = 1,...3,\ j = 1,...,N$ which are the values of the function for all vertices of the set Ω_h. So, rewriting (A.2) using (A.3), we obtain

$$\left(V^K\right)^T B^k V^K = \left(\left(L_1^k\right)^T V\right)^T B^k \left(L_1^k\right)^T V = V^T L_1^k B^k \left(L_1^k\right)^T V$$

and therefore

$$b(\underset{\sim}{v_h},\underset{\sim}{v_h}) = \sum_{K\in\Omega_h}\left(2\mu||\underset{\approx}{\varepsilon}(\underset{\sim}{v_h})||^2_{\underset{\approx}{L^2}(K)} + \int_K \lambda\left(\mathrm{div}\underset{\sim}{v_h}\right)^2 dx\right)$$

$$= \sum_{K\in\Omega_h}\left(V^K\right)^T B^k V^K = \sum_{K\in\Omega_h} V^T L_1^k B^k \left(L_1^k\right)^T V = V^T\left(\sum_{k=1}^{M} L_1^k B^k \left(L_1^k\right)^T\right) V = V^T B^* V$$

where $B^* = \sum_{k=1}^{M} L_1^k B^k \left(L_1^k\right)^T$. So we can conlude that $b(\underset{\sim}{v_h},\underset{\sim}{v_h}) = V^T B^* V$, being B^* the global stiffness matrix.

The matrix $B^* = \sum_{k=1}^{M} B_1^k$ is symmetric if B_1^k is symmetrical $\forall k \in \{1,...,M\}$. As the local matrix B^k is symmetric because matrix C is also symmetric. Then B_1^k is symmetric provided that $\left(B_1^k\right)^T = B_1^k$. Since

$$\left(B_1^k\right)^T = \left(L_1^k B^k \left(L_1^k\right)^T\right)^T = L_1^k \left(B^k\right)^T \left(L_1^k\right)^T = L_1^k B^k \left(L_1^k\right)^T = B_1^k$$

then B^* is symmetric.

After obtaining the matrix B^*, the next step is to determine the vector F_1^*.

A.2 Right-hand side F_1^*

In this section we will calculate the vector F_1^* in (3.7).

From (3.5) and (3.7), we obtain

$$V^T F_1^* = l_{1,h}(\underset{\sim}{v_h}) = \int_{\partial\Omega_1} \underset{\sim}{g}\cdot\underset{\sim}{v_h}\,ds + \omega^2\rho\int_\Omega \underset{\sim}{u_h}^*\cdot\underset{\sim}{v_h}\,dx + \int_\Omega \underset{\sim}{f}\cdot\underset{\sim}{v_h}\,dx. \tag{A.4}$$

Note that in (A.4) the first integral is defined over the boundary $\partial\Omega_1$ and the others are defined over Ω. Let´s start to calculate the first integral. As mentioned earlier, $\partial\Omega_1$ is a plane surface. Let $K_1,..,K_{M_1}$ be all the triangles that belong to one of the faces of one of the tetrahedrons of the partition presented before which are contained in $\partial\Omega_1$. The resulting subdivision (or mesh) is denoted by $\partial\Omega_{h1}$.

So we can write

$$\int_{\partial\Omega_1} \underset{\sim}{g}\cdot\underset{\sim}{v_h}\,ds = \sum_{K_1\in\Omega_{h1}} \int_{K_1} \underset{\sim}{g}\cdot\underset{\sim}{v_h}\,ds. \tag{A.5}$$

As a consequence of (3.2) we have that the M_1 triangles intersect only at a vertex, or along an edge or don't intersect. For each triangle K_1 of the discretization of $\partial\Omega_1$, let's consider the points $r_i = (x_i, y_i, z_i)\,,\ i = 1,...,3$ which correspond to the coordinates of its vertices. Without loss of generality we can assume that $\partial\Omega_1$ is parallel to the plane XOY and therefore all vertices of the respective triangles have coordinates $z_i = \alpha, \alpha\in\mathbb{R},\ i = 1,...,3$. Consider the local coordinates (ξ,η)

where the vertices of the reference triangle are represented on the coordinate plane over the axes, as in Figure A.1. In this form, we can write the coordinates of each point $r=(x,y,z)$ of K_1 as a convex

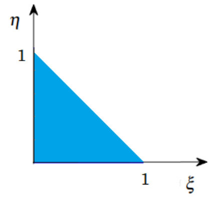

Fig. A.1 Triangle represented in local coordinates.

combination of the coordinates of the reference triangle

$$r=(x,y,z)=r_1\varphi_1(\xi,\eta)+r_2\varphi_2(\xi,\eta)+r_3\varphi_3(\xi,\eta),\; z=\alpha, \tag{A.6}$$

where

$$\varphi_1(\xi,\eta)=1-\xi-\eta,\;\; \varphi_2(\xi,\eta)=\xi,\;\; \varphi_3(\xi,\eta)=\eta.$$

The elements of $\{\varphi_i, i=1,2,3\}$ are called the nodal bases of the set of linear polynomials relatively to the local coordinates. For this transformation, the Jacobi's matrix is given by

$$J_1=\frac{\partial(x,y)}{\partial(\xi,\eta)}=\begin{pmatrix} x_2-x_1 & x_3-x_1\\ y_2-y_1 & y_3-y_1\end{pmatrix}$$

and the Jacobian is

$$|J_1|=\det\begin{pmatrix} x_2-x_1 & x_3-x_1\\ y_2-y_1 & y_3-y_1\end{pmatrix}=\det\begin{pmatrix} x_1 & y_1 & 1\\ x_2 & y_2 & 1\\ x_3 & y_3 & 1\end{pmatrix}.$$

Note that $|J_1|=2A_{123}$, where A_{123} is the area of the triangle K_1 defined by $r_1,\ldots,r_3$.

For any function $\underset{\sim}{v_h}\in\underset{\sim}{V_h}$, and for $(x,y,z)\in\partial\Omega_{h1}$, we have that

$$v_{ih}(x,y,z)=v_{ih}(r(\xi,\eta))=\sum_{j=1}^{3}V_{ji}\varphi_j(\xi,\eta),\; i=1,\ldots,3,\; z=\alpha, \tag{A.7}$$

where V_{ji} is the value of the function v_{ih} in the vertex of the tetrahedron K with position r_j, $i=1,\ldots,3,\; j=1,\ldots,3$. Therefore

$$\underset{\sim}{v_h}=\begin{pmatrix} v_{1h}(x,y,z)\\ v_{2h}(x,y,z)\\ v_{3h}(x,y,z)\end{pmatrix}=\begin{pmatrix} V_{11}\\ V_{12}\\ V_{13}\end{pmatrix}\varphi_1(\xi,\eta)+\begin{pmatrix} V_{21}\\ V_{22}\\ V_{23}\end{pmatrix}\varphi_2(\xi,\eta)+\begin{pmatrix} V_{31}\\ V_{32}\\ V_{33}\end{pmatrix}\varphi_3(\xi,\eta).$$

Now let's consider that, for any triangle K_1, the points r_1,r_2,r_3 correspond to the vertices V_1,V_2,V_3 respectively. Consider that the function $\underset{\sim}{g}$ in (A.5) defined in triangle K_1 can be written, in terms of the local coordinates, as

$$g_j\,(x,y,z) = g_j\left(\frac{1}{3}\sum_{i=1}^{3} V_i\right)\left(\sum_{i=1}^{3}\varphi_i\right),\ j = 1,2,3. \tag{A.8}$$

Therefore, by (A.7) and (A.8) we have

$$\int_{K_1} \underset{\sim}{g}\cdot \underset{\sim}{v}_h\, ds = \sum_{j=1}^{3}\int_{K_1} g_j v_{jh}\, ds$$

$$= |J_1|\sum_{j=1}^{3}\int_{\Delta_T}\left[g_j\left(\frac{1}{3}\sum_{i=1}^{3} V_i\right)\left(\sum_{i=1}^{3}\varphi_i\right)\right](V_{1j}\varphi_1 + V_{2j}\varphi_2 + V_{3j}\varphi_3)\, d\xi\, d\eta,$$

where Δ_T defines the triangle in the local coordinates (ξ,η), i.e.,

$$\Delta_T = \{(\xi,\eta) : 0 \le \xi \le 1, 0 \le \eta \le 1-\xi\}.$$

We can summarize the last equality in the following matrix form

$$\int_{K_1} \underset{\sim}{g}\cdot \underset{\sim}{v}_h\, ds = |J_1|\sum_{j=1}^{3}\int_{\Delta_T}[V_{1j}V_{2j}V_{3j}]\, D\begin{bmatrix} G_j \\ G_j \\ G_j \end{bmatrix} \tag{A.9}$$

where $G_j = g_j\left(\frac{1}{3}\sum_{i=1}^{3} V_i\right)$, $j = 1,\ldots,3$ and D is a 3×3 matrix where the component (i,j) is given by

$$\int_{\Delta_T}\varphi_i\varphi_j d\xi d\eta,\quad i,j\in\{1,2,3\}. \tag{A.10}$$

To obtain the entries of the matrix D we have to calculate the integrals in (A.10). Let us start with the diagonal elements, i.e., the integrals where the integrating function is $\varphi_i^2, i = 1,2,3$, $\int_{\Delta_T}\varphi_i^2\, d\xi\, d\eta$ $= \int_0^1\int_0^{1-\xi}\varphi_i^2\, d\eta\, d\xi$. We obtain

$$\int_{\Delta_T}\varphi_1^2\, d\xi\, d\eta = \int_{\Delta_T}\varphi_2^2\, d\xi\, d\eta = \int_{\Delta_T}\varphi_3^2\, d\xi\, d\eta = \frac{1}{12}.$$

Next, lest us calculate the integrals whose integrating function is the product $\varphi_i\varphi_j$ for $i \neq j$. In this case $\int_{\Delta_T}\varphi_i\varphi_j\, d\xi\, d\eta = \int_0^1\int_0^{1-\xi}\varphi_i\varphi_j\, d\eta\, d\xi$ and

$$\int_{\Delta_T}\varphi_1\varphi_2\, d\xi\, d\eta = \int_{\Delta_T}\varphi_1\varphi_3\, d\xi\, d\eta = \int_{\Delta_T}\varphi_2\varphi_3\, d\xi\, d\eta = \frac{1}{24}.$$

Taking into account the previous expressions for the entries of the matrix D, we can write (A.9) as follows:

$$\int_{K_1} \underset{\sim}{g} \cdot \underset{\sim}{v_h}\, ds = |J_1| \sum_{j=1}^{3} [V_{1j}V_{2j}V_{3j}V_{4j}]\, D_1 \begin{bmatrix} G_j \\ G_j \\ G_j \\ 0 \end{bmatrix},$$

i.e., is the same as

$$\left(V^K\right)^T D_2 G^K = \left(V^K\right)^T g^k,$$

where $G^K = [G_1 G_1 G_1\, 0\, G_2 G_2 G_2\, 0\, G_3 G_3 G_3\, 0]^T$,

$$D_2 = |J_1| \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_1 & 0 \\ 0 & 0 & D_1 \end{pmatrix}, D_1 = \begin{bmatrix} D & 0_{3\times 1} \\ 0^T_{3\times 1} & 0 \end{bmatrix}, D = \frac{1}{24} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix},$$

and $0_{3\times 1}$ is a 3×1 vector of zeros.

Next, we present the calculation of the second integral in (A.4):

$$\int_{\Omega} \underset{\sim}{u_h}^* \cdot \underset{\sim}{v_h}\, dx = \sum_{K \in \Omega_h} \int_K \underset{\sim}{u_h}^* \cdot \underset{\sim}{v_h}\, dx.$$

Consider that, for any tetrahedron K, the points r_1, r_2, r_3, r_4 correspond to the vertices V_1, V_2, V_3, V_4, respectively. The function $\underset{\sim}{u_h}^*$ defined in the tetrahedron K in terms of the local coordinates, can be written as

$$u^*_{jh}(x,y,z) = u^*_{jh}\left(\frac{1}{4}\sum_{i=1}^{4} V_i\right)\left(\sum_{i=1}^{4} \psi_i\right),\ j = 1,2,3.$$

So, by this approach we have

$$\int_K \underset{\sim}{u_h}^* \cdot \underset{\sim}{v_h}\, dx = \sum_{j=1}^{3} \int_K u^*_{jh}(x,y,z) v_{jh}(x,y,z)\, dx\, dy\, dz = |J| \sum_{j=1}^{3} \int_{\Delta} u^*_{jh}(r(\xi,\eta,\tau)) v_{jh}(r(\xi,\eta,\tau))\, d\xi\, d\eta\, d\tau, \tag{A.11}$$

where Δ defines the tetrahedron in local coordinates (ξ, η, τ), i.e.,

$$\Delta = \{(\xi,\eta,\tau) : 0 \le \xi \le 1, 0 \le \eta \le 1-\xi, 0 \le \tau \le 1-\xi-\eta\}.$$

Replacing (3.9) in (A.11) we obtain

$$\int_K \underset{\sim}{u_h}^* \cdot \underset{\sim}{v_h}\, dx = |J| \sum_{j=1}^{3} \int_{\Delta} \left[u^*_{jh} \left(\frac{1}{4}\sum_{i=1}^{4} V_i\right)\left(\sum_{i=1}^{4} \psi_i\right)\right] \times$$

$$\times \left(V_{1j}\psi_1 + V_{2j}\psi_2 + V_{3j}\psi_3 + V_{4j}\psi_4\right) d\xi\, d\eta\, d\tau.$$

This equality can be summarized in the following matrix form

$$\int_K \underset{\sim}{u_h}^* \cdot \underset{\sim}{v_h}\, dx = |J| \sum_{j=1}^{3} [V_{1j}V_{2j}V_{3j}V_{4j}]\, E \begin{bmatrix} U_j^* \\ U_j^* \\ U_j^* \\ U_j^* \end{bmatrix}, \tag{A.12}$$

where $U_j^* = u_{jh}^*\left(\frac{1}{4}\sum_{i=1}^{4} V_i\right)$, $j = 1,...,3$ and E is a 4×4 matrix where the component (i,j) is given by

$$\int_\Delta \psi_i \psi_j \, d\xi \, d\eta \, d\tau, \quad i,j \in \{1,2,3,4\}. \tag{A.13}$$

To obtain the entries of the matrix E we have to calculate the integrals in (A.13). We start with the diagonal elements, i.e., the integrals where the integrating function is $\psi_i^2, i = 1,2,3,4$, $\int_\Delta \psi_i^2 \, d\xi \, d\eta \, d\tau = \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} \psi_i^2 d\tau \, d\eta \, d\xi$. We obtain

$$\int_\Delta \psi_1^2 \, d\xi \, d\eta \, d\tau = \int_\Delta \psi_2^2 \, d\xi \, d\eta \, d\tau = \int_\Delta \psi_3^2 \, d\xi \, d\eta \, d\tau = \int_\Delta \psi_4^2 \, d\xi \, d\eta \, d\tau = \frac{1}{60}.$$

Next, we will calculate the integrals where the integrand function is $\psi_i \psi_j$ for $i \neq j$, $\int_\Delta \psi_i \psi_j \, d\xi \, d\eta \, d\tau = \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} \psi_i \psi_j d\tau \, d\eta \, d\xi$. We obtain

$$\int_\Delta \psi_1 \psi_2 \, d\xi \, d\eta \, d\tau = \int_\Delta \psi_1 \psi_3 \, d\xi \, d\eta \, d\tau = \int_\Delta \psi_1 \psi_4 d\xi \, d\eta \, d\tau = \int_\Delta \psi_2 \psi_3 \, d\xi \, d\eta \, d\tau = \frac{1}{120} \text{ and}$$

$$\int_\Delta \psi_2 \psi_4 \, d\xi \, d\eta \, d\tau = \int_\Delta \psi_3 \psi_4 \, d\xi \, d\eta \, d\tau = \frac{1}{120}.$$

Taking into account the previous expressions for the entries of the matrix E, we can write (A.12) as follows:

$$\int_K \underset{\sim}{u_h}^* \cdot \underset{\sim}{v_h}\, dx = \frac{|J|}{120} \left(V^K\right)^T \begin{pmatrix} E_1 & 0 & 0 \\ 0 & E_1 & 0 \\ 0 & 0 & E_1 \end{pmatrix} (U^*)^K = \left(V^K\right)^T B' \, (U^*)^K$$

where $(U^*)^K = [U_1^*U_1^*U_1^*U_1^*U_2^*U_2^*U_2^*U_2^*U_3^*U_3^*U_3^*U_3^*]^T$ and

$$E_1 = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}, B' = \frac{|J|}{120} \begin{pmatrix} E_1 & 0 & 0 \\ 0 & E_1 & 0 \\ 0 & 0 & E_1 \end{pmatrix}.$$

Next the purpose is to determine the explicit form of F_1^*. For this it should be noted that the expression in (A.3) is different. Instead of having the matrix L_1^k we will define another one that is L_2^k. It is a block matrix where each diagonal block has a matrix of dimension $N \times 4$ but the fourth column will be with zeros. So for the triangles we obtain $V^K = \left(L_2^k\right)^T V$.

To calculate the last integral in (A.4) it´s enough to consider what has been done previously for the integrals over Ω. Therefore, we have that

$$\int_K \underset{\sim}{f} \cdot \underset{\sim}{v_h}\, dx = \left(V^K\right)^T B' \left(F^*\right)^K$$

where $\left(F^*\right)^K = [F_1^* F_1^* F_1^* F_1^* F_2^* F_2^* F_2^* F_2^* F_3^* F_3^* F_3^* F_3^*]^T$.

From (A.4) we have

$$\begin{aligned} l_{1,h}(\underset{\sim}{v_h}) &= \sum_{K_1 \in \Omega_{h1}} \int_{K_1} \underset{\sim}{g} \cdot \underset{\sim}{v_h}\, ds + \omega^2 \rho \sum_{K \in \Omega_h} \int_K \underset{\sim}{u_h}^* \cdot \underset{\sim}{v_h}\, dx + \sum_{K \in \Omega_h} \int_K \underset{\sim}{f} \cdot \underset{\sim}{v_h}\, dx \\ &= \sum_{k=1}^{M_1} \left(V^K\right)^T g^k + \sum_{k=1}^{M} \omega^2 \rho \left(V^K\right)^T B' \left[\left(U^*\right)^K + \left(F^*\right)^K\right]. \end{aligned}$$

Then replacing V^K and $\left(U^*\right)^K$ results

$$l_{1,h}(\underset{\sim}{v_h}) = \sum_{k=1}^{M_1} V^T L_2^k g^k + \sum_{k=1}^{M} \omega^2 \rho V^T L_1^k B' \left(L_1^k\right)^T U^* + V^T L_1^k B' \left(F^*\right)^K = V^T \left(F^* + F_1\right) = V^T F_1^*$$

where $F^* = \sum_{k=1}^{M_1} L_2^k g^k + \sum_{k=1}^{M} L_1^k B' \left(F^*\right)^K$ and $F_1 = \omega^2 \rho \sum_{k=1}^{M} L_1^k B' \left(L_1^k\right)^T U^*$.