

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

A simple image segmentation approach to overcome microstructure analysis and homogenization problems

João Pedro Portella Guedes Visconti



Mestrado em Mecânica Computacional

Supervisor: Prof. Dr. Mariana Rita Ramos Seabra

Second Supervisor: Prof. Dr. José César De Sá

June 22, 2021

A simple image segmentation approach to overcome microstructure analysis and homogenization problems

João Pedro Portella Guedes Visconti

Mestrado em Mecânica Computacional

June 22, 2021

Abstract

Multiscale modeling has proven to be a very useful tool in the production of new manufacturing methods, in the development of quality products, reducing production costs, and creating new materials. The need to improve some characteristics of this process drive the motivation for the current study that was developed at the Faculty of Engineering of the University of Porto (FEUP) together with INEGI - Institute of Science and Innovation in Mechanical and Industrial Engineering. The work developed in the thesis is complementary with a broader project (msCore) ¹ that aims at the development of a multiscale analysis model for advanced material manufacturing.

Therefore, the main goal of this work is to build a multiscale approach to derive macro properties of Duplex Stainless Steel from experimental images of the microstructure. In this way, this study uses image processing tools to recognize the different phases in the microstructure and adapt them to a file format that allows the integration to a finite element software. In turn, having this file, micro properties are homogenized to obtain information regarding the macroscale properties of this material.

In order to develop the referred software, a bibliographic study was made in order to observe similar problems and their resolutions. The advantages and disadvantages of different authors' studies were taken into account. The limitation regarding microstructure analysis of the material was addressed with a script made in Matlab that identifies the features in the microstructure image and converts them into numerical data to be used in a Finite Element software. One of the main issues that the msCore project presented was the transcription of the visual experimental data into a model for numerical simulation. A Python script is used to extract the needed information from the results simulations obtained using ABAQUS commercial software. Thereat, another Matlab script was created to obtain the homogenization of the properties and predict the elasticity matrix for the material in the microstructure.

In order to evaluate the developed software, a case study is presented. The steps of microstructure analysis, Finite Element simulation, and assembly of the elasticity matrix with real images of the microstructure of a Super Duplex Stainless Steel sample were applied. The results demonstrate the influence of the percentage of each phase present in the microstructure in the elastic material properties. The influence on inelastic material properties should be addressed in future work.

Keywords: Image Segmentation, Homogenization, FEM, Duplex Stainless Steel

¹Project NORTE-01-0145-FEDER032419 - "msCORE - Multiscale methodology with reduced ordering of models for advanced materials and processes and Project POCI-01-0145-FEDER-032466 - NanosFLiD"

Resumo

A modelagem multiescala tem se mostrado uma ferramenta muito útil na produção de novos métodos de fabricação, no desenvolvimento de produtos de qualidade, reduzindo custos de produção e criando novos materiais. A necessidade de melhorar algumas características deste processo motivou o presente estudo que foi desenvolvido na Faculdade de Engenharia da Universidade do Porto (FEUP) em conjunto com o INEGI - Instituto de Ciência e Inovação em Engenharia Mecânica e Industrial. O trabalho desenvolvido na tese é complementar a um projeto mais amplo (msCore)² que visa o desenvolvimento de um modelo de análise multiescala para fabricação avançada de materiais.

Portanto, o objetivo principal deste trabalho é construir uma abordagem multiescala para derivar macro propriedades do Aço Inoxidável Duplex a partir de imagens experimentais da microestrutura. Desta forma, este estudo utiliza ferramentas de processamento de imagens para reconhecer as diferentes fases da microestrutura e adaptá-las a um formato de arquivo que permite a integração a um software de elementos finitos. Por sua vez, de posse desse arquivo, as micropropriedades são homogeneizadas para se obter informações a respeito das propriedades da macroescala desse material.

Para avaliação do software desenvolvido, é apresentado um estudo de caso. Foram aplicadas as etapas de análise da microestrutura, simulação de Elementos Finitos e montagem da matriz de elasticidade com imagens reais da microestrutura de uma amostra de Aço Inoxidável Super Duplex. Os resultados demonstram a influência da porcentagem de cada fase presente na microestrutura nas propriedades elásticas do material. A influência nas propriedades do material inelástico deve ser abordada em trabalhos futuros.

Keywords: Segmentação de Imagem, Homogeneização, FEM, Aço Inoxidável Superduplex

²Projeto NORTE-01-0145-FEDER032419 – “msCORE - Multiscale methodology with model order reduction for advanced materials and processes and Project POCI-01-0145-FEDER-032466 – NanosFLiD”

Acknowledgements

Primeiramente gostaria de agradecer à minha orientadora Prof. Mariana Seabra por toda atenção e dedicação. Ao Prof. César Sá, pelos ensinamentos, paciência e a oportunidade de trabalhar no projeto NORTE-01-0145-FEDER032419 – “msCORE - Multiscale methodology with model order reduction for advanced materials and processes and Project POCI-01-0145-FEDER-032466 – NanosFLiD” cofinanciado pelo Programa Operacional Regional do Norte (NORTE2020) e pelo Programa Operacional de Competitividade e Internacionalização (Compete2020), através do Fundo Europeu de Desenvolvimento Regional (FEDER) e pela Fundação para a Ciência e Tecnologia (FCT) através da sua componente do orçamento do estado.

Agradeço ao INEGI e a FCT, pelo suporte logístico e financeiro para desenvolvimento desse trabalho.

Um agradecimento muito especial a minha esposa Vívian, que me encorajou e sempre incentivou minhas ambições. Agradeço imensamente aos meus pais Maria Eduarda e Renato, por todo apoio, suporte e amor incondicional. A minha irmã Guta, minha inspiração. Obrigado a todos que de alguma forma fizeram parte dessa jornada.

João Pedro Portella Guedes Visconti



*“If I had nine hours to chop down a tree,
I’d spend the first six sharpening my axe.”*

Abraham Lincoln

Contents

1	Introduction	1
1.1	Document Structure	3
2	Material Properties	5
2.1	Crystalline structure	5
2.2	Iron-based alloy	6
2.3	Stainless Steel	7
2.4	Duplex Stainless Steel (DSS)	7
2.5	Phase transformation	8
2.6	Secondary phases	9
2.6.1	σ Phase	9
2.6.2	χ Phase	9
2.7	Kinetics	11
2.7.1	Transformation sequence	11
2.7.2	TTT - Time-Temperature-Transformation diagram	12
2.7.3	CCT - Continuous-Cooling-Transformation	13
2.8	Precipitation during fabrication and heat treatment process	15
2.9	Sand Casting Process	16
2.10	Macroscale properties	16
3	Microstructure Analysis	19

3.1	Image Segmentation	20
3.1.1	Image Segmentation Technics	21
4	FEM - Finite Element Method	25
4.1	Formulation	25
4.1.1	Kinematics	26
4.1.2	Equilibrium Equations	28
4.1.3	Strong and Weak Form	30
4.1.4	Finite Element Discretization	31
4.2	Triangular Element	35
4.2.1	Isoparametric Element	35
4.2.2	Shape Functions	35
4.2.3	Jacobian	36
4.2.4	Gauss Quadrature	37
4.2.5	Homogenization Method	37
5	Matlab Code and Image Processing	41
5.1	Complete Image First Analysis	43
5.2	Sectioned Image Analysis	45
5.2.1	Pre-processing	45
5.2.2	Phase Segmentation Function	46
5.2.3	Body Recognition	48
5.2.4	CAD Files	49
6	Abaqus Simulation	51
6.1	Importing Sketch	51
6.2	Assemble and Material Assignment	51
6.3	Mesh	53
6.4	Boundary Conditions and Load	55

6.5	Post-processing Python Script	57
6.6	Elasticity Matrix D	57
6.6.1	Symbolic Jacobian Matrix	57
6.6.2	Jacobian Matrix Numerical Determinant	57
6.6.3	Assembling Elasticity Matrix D	58
7	Case Study	61
7.1	Problem Description	61
8	Conclusion	69
8.1	Results and Discussions	69
8.2	General Observations	69
A	Matlab Segmentation Code	71
A.1	Main Program	71
A.2	Phase Segmentation Function	72
A.3	Color Assignment Function	73
A.4	Body Detection Function	73
A.5	Create DXF file Function	74
B	Python Script	77
B.1	Extract Results Python Script	77
C	Matlab Post-processing Codes	81
C.1	Main Program	81
C.2	Symbolic Jacobian Function	82
C.3	Jacobian Determinant Evaluation Function	83
C.4	Build Elasticity Matrix Function	83
	References	87

List of Figures

2.1	Crystalline Structures.	6
2.2	Sample of a microstructure image of the DSS.	9
2.3	Schematic Time-Temperature precipitation diagram [26].	10
2.4	Phase diagram for 70% Fe in ternary Fe-Cr-Ni diagram [12].	10
2.5	Growth and Nucleation rates by temperature [17].	11
2.6	Fe-Cr-Ni phase diagram [59].	12
2.7	Time-Temperature-Transformation (TTT) and Continuous-Cooling-Transformation (CCT) curves for duplex stainless steel 2205 [25].	13
2.8	Isothermal fraction transformation versus logarithm of time [23].	14
2.9	TTT curves for σ -phase precipitation [68].	14
2.10	CCT Example [62].	15
2.11	Illustration of casting process.	16
2.12	Stress-Strain curve [4].	17
2.13	Impact of Sigma concentration [71].	18
3.1	Histogram of a microstructure image.	21
3.2	Clustering example [44].	22
3.3	Classical segmentation technique using morphological operators.	23
4.1	(a) Domain of the problem and (b) traction vectors acting on the infinitesimal element in a 2D problem [43].	26
4.2	Element elongation.	27

4.3	Finite element discretization in two dimensions.	32
4.4	A Single Triangular Element.	32
4.5	Mapping the Triangular Element.	36
5.1	Schematics image segmentation process.	42
5.2	Layer scheme with the three phases.	43
5.3	Pixels values in a gray scale.	44
5.4	Pixels values assigned to a particular phase.	44
5.5	Image segmentation for section A - γ -phase(Red), δ -phase(Green), σ -phase(Blue).	45
5.6	Pixels values assign to a particular phase.	45
5.7	Pre-processing result.	46
5.8	Schematic of the segmentation of the matrix containing the pixels.	47
5.9	Phase color assignment for the microstructure image.	47
5.10	Segmented image with color assign to the different phases.	48
5.11	γ -phase layer.	48
5.12	All bodies in γ -phase layer.	49
6.1	Spline sketch creation.	52
6.2	Difference in mesh before and after spline creation.	52
6.3	γ -phase, δ -phase and σ -phase merged in one model.	53
6.4	Material Assignment.	54
6.5	Mesh of the microstructure model.	54
6.6	Mesh optimization graph.	55
6.7	Detail on mesh through different phases.	56
6.8	Load and Boundary Conditions.	56
6.9	Schematics Abaqus and postprocessing process	59
7.1	C-ring specimen (dimensions in [mm]).	62
7.2	Microstructure of the 6 parts obtained from the C-ring - microscopy images.	62

7.3	Microstructure of the 6 parts obtained from the C-ring - SEM.	63
7.4	Results for the simulation in X direction.	64
7.5	Results for the simulation in Y direction.	65
7.6	Results for the simulation in shear.	65
7.7	Results for the simulation in shear.	66

List of Tables

2.1	Nominal chemical composition of cast DSS 25Cr-7Ni-Mo-N [68]	8
6.1	Phase properties of different phases of DSS	53
7.1	Percentage of phases on sections A to F	64
7.2	Percentage of precipitates on sections A to F	64
7.3	Elasticity Matrix for all sections on C-ring specimen (<i>GPa</i>)	66
7.4	Young's modulus (<i>E</i>) for each section of the C-ring (<i>MPa</i>)	67

Abbreviations

FEM	Finite Element Method
SDSS	Super Duplex Stainless Steel
RVE	Representative Volume Element
FCC	Face-Centered Cubic Structure
BCC	Body-Centered Cubic Structure
HCP	Hexagonal Close-Packed Structure
TTT	Time-Temperature-Transformation
CCT	Continuous-Cooling-Transformation
ASTM	American Society for Testing and Materials
SDSS	Super Duplex Stainless Steel
CSV	Comma-Separate Values
OOF	Object-Oriented Finite-Element
SEM	Scanning Electron Microscope
FEA	Finite Element Analysis

Chapter 1

Introduction

Engineers are always searching for tools that allow the development of better products with faster and cheaper manufacturing processes. Material and geometry are some of the points of great interest with regard to the quality of products. Physical tests and simulations developed by specific software are constantly used to analyze the mechanical properties of a given geometry. These analyses are made considering a homogeneous material at the macroscale and have proven to be successful in predicting failure in project without increasing too much the cost of production.

When it comes to microscale, the heterogeneous nature of the material has a deep impact on the mechanical properties of the product. Thus, multiscale material modeling technics come into place used to understand the relationship between the microstructure of the material and the final mechanical behavior. Financial motivation is one of the main reasons for the advance of multiscale modeling in the industrial sector. This sector focuses mainly on the reduction of product cost provided by multiscale modeling by avoiding trial-and-error interaction, innovating in the material and process design, reducing the scale of experimental testing, to name a few. Moreover, the accurate prediction of the multiscale modeling increases quality, performance, and reliability of the product [41]. The first challenge for multiscale material modeling is microstructure analysis. The accuracy of the multiscale modeling relies on the quality of the microstructure analysis of the material. For complex microstructures, the process of identifying the important information about a certain material can be often slow and labor intensive [24]. Identifying different features such as grain size and the present phases, helps to understand the mechanical properties of the material. Microstructure characteristics are commonly difficult to interpret and required intensive working hours from an expert material scientist. Often, not to rely only on a personal visualization, image analysis softwares are used to facilitate and improve the process. These commercial softwares provide some tools to segment an image into different groups and regions of interest but they often lack in the automation part of the process and repeatability of results.

Characterization of microstructural features may be used not only for multiscale modeling but

it also may play an important role in microstructure control during manufacturing. In this case, several analyzes are made regarding different points in a part and the different processes to which the product is subjected, such as, heat treatment, welding process, and so forth. All this analysis may cost too much personal time. In order to meet different needs and objectives, there is a requirement for new data processing technics that reduce time and human error. Besides that, other features can be included in the program such as to transform the visual data information into files that can be further analyzed by a finite element software, observe the main distribution of a particular phase as well as its percentage, among others. In order to accomplish these requirements, the developed software should have a simple and open-source code so that future investigations can adapt and improve it as necessary.

In particular, Duplex Stainless Steel (DSS) is of interest for the present work. This alloy presents a microstructure composed mainly of two phases, Austenite, also called the γ -phase and Ferrite, known as the δ -phase in nearly equal amount. Due to its higher mechanical and corrosion resistance, DSS is widely explored in the literature as it going to be presented in chapter 2. Both wrought and cast duplex stainless steel have their strengths and weaknesses depending on the application. The current work is mainly focused on sand casting duplex stainless steel, for which, cooling during manufacturing or heat treatment may be critical. If not done properly, the cooling process of the DSS can result in precipitate phases that are harmful to the material. Therefore, monitoring and control of the microstructure of the material throughout these processes are fundamental for the quality of the final product.

The motivation for the current work comes from a project that aims at a multiscale model to describe the behavior of the Duplex Stainless Steel alloy 25Cr-7Ni-Mo-N. The use of commercial software in relation to the recognition of the phases present in the microstructure has revealed some drawbacks during the analysis of the microstructure. This thesis proposes a simple program to overcome the challenges of microstructural analysis, identifying the characteristics within the microstructure without any further user intervention. In addition, it presents a solution that facilitates the transcription of experimental visual data in a file that can be used for numerical simulation such as Finite Element Analysis (FEA). It is important for this work that the program developed should be easy to understand so that it can be applied to different characteristics of the microstructure and ensure that in future work different image processing techniques can be incorporated to improve the automation of the analysis.

As a result, the developed approach comprises 4 stages. A threshold segmentation technic is used to divide the image into different phases and create three different layers with each one of them. These layers are then converted to a binary image to make it easier for the program to recognize the borders of the bodies formed by each phase and transform the visual boundaries of the image into numerical data. These numerical values correspond to the coordinates of the pixel bellowing to the border of the body. With these values, in a second stage, the program is capable of converting it into a CAD file allowing the analysis in the finite element commercial

software Abaqus, which constitutes the third stage. Finally, different simulations are made with the imported file regarding the boundary conditions and loads, these simulations are better explained in chapter 6. From the results obtained from the Finite Element Analysis, it is possible, using the theory explained in chapter 4, to obtain the elasticity matrix of the material at the macroscale.

1.1 Document Structure

This thesis is divided into eight chapters and in addition, it includes at the end, reference section and an appendix with the codes used in the developed program. The division of each chapter is made in the following manner:

Chapter 2

In chapter 2, the characterization of the material properties regarding the nanoscale, microscale and macroscale is addressed. The influence of the crystalline structure in the average size and shape of grains is pointed out. Additionally, the role of the composition of the microstructure in macroscale properties and their connection is presented. Finally, the phase transformation process and respective characteristics are described, in particular for the Duplex Stainless Steel.

Chapter 3

Chapter 3 outlines the importance of a well-executed analysis of the microstructure. Methods based on direct human observation require time and experience to be performed, and, for this reason, computational methods to overcome this limitation are presented. Image segmentation is a digital image processing technic that is the main focus of the current work. Different methods can be incorporated to obtain the best result in extracting the information contained in the microstructure, and thus some of them are described in this chapter.

Chapter 4

The finite element method is a key feature in the complete approach developed in this thesis, used at two different times. This chapter briefly describes the formulation of the strong and weak form of the equilibrium equation for structural problems, followed by the respective finite element discretization that is used in the elastic deformation problem solved by Abaqus software. In a second step, the theory involving the isoparametric elements is presented, as it will be used to build the script that will assemble the elasticity matrix.

Chapter 5

In chapter 5, a more detailed description of the image segmentation technique used is presented. At this point, a step-by-step procedure on how the images are treated and how the processes regarding the phase segmentation function work, are described. Then, the process of recognizing the borders and converting them into a CAD file is explained.

Chapter 6

Chapter 6 focuses on the simulation performed by the Abaqus software on the new microstructure file. This chapter describes how the files are imported in Abaqus and all the processes of assigning the material to each one of the phases, generating the mesh and evaluating it, and finally assigning the boundary conditions and loads applied to the model.

Chapter 7

In chapter 7, real images obtained from a Super Duplex Stainless Steel specimen are used to illustrate the developed methodology. From the two different technics used to obtain the images, optical microscopy and scanning electron microscope results for the percentage of the phases are shown. Furthermore, the rest of the procedure is applied to six images and the results of the Abaqus simulation and elasticity matrix obtained for each one of the images are explored and discussed.

Chapter 8

In this chapter the conclusions regarding the software developed in this thesis are presented and discussed, along with some suggestion for improvement for future work. General observations are made mainly about the algorithm used and the additional functionality that the in-house software has compared to commercial software.

Chapter 2

Material Properties

A good characterization of material properties is essential to solve virtually any engineering problem. These properties may be affected at different instantiations such as the nano, micro, or macroscale. Regarding the nanoscale, size, shape, and direction of atoms may be pointed out, while at the macroscale materials are often treated as a continuum exhibiting relevant structural properties, such as tensile strength, elasticity modulus or thermal conductivity, just to name a few. Intermediate scales may also be defined and thus are termed mesoscales (e.g. polycrystalline structures). Therefore, in this chapter, the most relevant aspects of multi-scale material modelling of steels will be addressed, in particular of duplex stainless steels (DSS).

2.1 Crystalline structure

In general, the finer scale addressed when studying the behavior of metallic materials is the nanoscale. Here, the type of bond between atoms define their interactions. Thus, there are three different types of interactions, as follows [1]:

- Ionic bond, an electron is transferred to other atom
- Covalent bond, two atoms can share a pair of electrons
- Metallic bond, the electrons form a kind of sea or cloud, that is, electrons are shared by all of the atoms

As the name suggests, metallic bounds are those present in metals, explaining their general behavior. Regularity in the spacial arrangement of atoms results in typical crystalline structures. For metals there are three main common structures: Face-Centered Cubic (FCC), Body-Centered Cubic (BCC) and Hexagonal Close-Packed (HCP) [23], as demonstrated in figure 2.1.

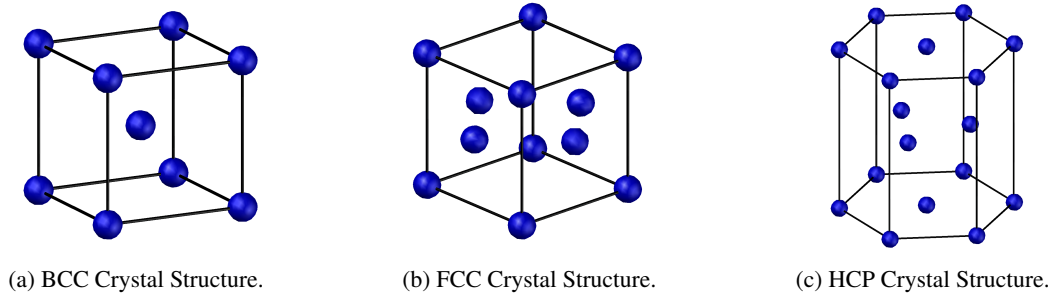


Figure 2.1: Crystalline Structures.

Nevertheless, crystalline structures always contain imperfections, namely as vacancies, self-interstitials, impurities, and dislocations [23]. Thus, materials are rarely single crystals, but composed of many small crystals termed grains. Naturally, grain boundaries are present wherever there is contact between different grains or some atomic mismatch. Thus, shape and average size of grains as well as their mutual interactions affect the mechanical properties, which are reflected at the macroscopic scale [23, 40]. Consequently, microstructure examination is essential for a good characterization of material performance at all scales. [27, 34, 55].

2.2 Iron-based alloy

Due to its mechanical properties and good cost-quality relation, steel has been used in a variety of applications over the years, in several industries such as manufacturing, automotive and construction. [14, 21, 57, 74]. Steel is composed mainly by iron and carbon, nonetheless, different alloying elements are added to this base composition to meet the constant demand for better fracture, corrosion or fatigue resistance, weldability or other manufacturing characteristics [74]. Chromium, nickel, molybdenum, niobium are examples of such alloying elements, resulting in several complex steel compositions [33, 46].

As the basic steel alloy is composed of iron (Fe) and carbon (C), it may display three allotropic variants, namely, two with a crystalline structure body-centered cubic (BCC), alpha and delta, also called α -ferrite and δ -ferrite, and one with faced-centered cubic (FCC) structure, named austenite or gamma (γ) phase [46]. Moreover, strength and hardness of the material increase with the increasing carbon concentration, while corrosion resistance is gained through the addition of chromium and other elements, which promote stabilization of the ferrite and austenite phases. Corrosion resistant steels form the Stainless Steel group, which is described in the following subsection.

2.3 Stainless Steel

Stainless Steel is a group of iron-based alloys with higher corrosion and long-lasting corrosion resistance [14]. It is characterized by predominance of chromium, with a minimum weight percentage equal to 11 wt% Cr, and may be improved by addition of nickel and molybdenum. Due to its wide range of mechanical properties and corrosion resistance, stainless steels have numerous applicabilities [23], from structural components, oil and gas industry, aerospace components, to chemical industry, food, pharmacy, and so forth.

This type of alloys is often classified accordingly to its microstructure composition. Therefore, stainless steels are classified as Martensitic, Austenitic, Ferritic, or a combination of the last two called Duplex Stainless Steel.

Austenite stainless steel are composed of austenite (or γ phase) and have a Face-Centered Cubic structure. Typically, Ni is added to base composition of Fe-C to stabilize the austenitic structure [46]. This type of steel is used in extreme temperature environments.

On the other hand, ferrite stainless steel is composed of δ or ferrite phase and body-centered cubic crystal structure (BCC). This type of steel is characterized by 12 to 30 wt% Cr while C levels are kept at a minimum, to avoid the formation of austenite. One of the advantages of ferrite stainless steel is the corrosion resistance due to its high Cr-contents.

Finally, duplex stainless steels possess a microstructure with balanced Ferrite (BCC) and Austenite (FCC) content [65], which provides a higher resistance to stress corrosion cracking and higher strength when compared to austenitic stainless steels [30]. This type of structure is obtained adding Cr and Mo to promote ferrite stabilization together with the addition of Ni to promote the formation of austenite. Other alloying elements such as Si, Al, Mn and N may also be present. This high alloy content is responsible by the exceptionally good combination of corrosion resistance and high tensile strength, nevertheless, during processing or heat treatment, it may result in precipitation of detrimental intermetallic secondary phases [71].

2.4 Duplex Stainless Steel (DSS)

As described in the previous section Duplex Stainless Steels (DSS) possess a balanced mix of ferrite (δ -ferrite) and austenite (γ -austenite) phases in their microstructure, stabilized mainly by chromium, nickel, and molybdenum. This work focus on a particular casting Duplex Stainless Steel alloy 25Cr-7Ni-Mo-N, which presents two times more mechanical resistance when compared to a normal austenite steel [6]. It is constituted by 25% Cr 7.5% Ni 3.5% Mo 0,25% N each one playing an important role in the mechanical and thermal properties of the material (see table 2.1).

C	Si	Mn	Ni	Mo	Cr	Cu	W	N
0.015%	0.5%	0.5%	7.5%	3.5%	25.0%	0.75%	0.75%	0.25%

Table 2.1: Nominal chemical composition of cast DSS 25Cr-7Ni-Mo-N [68]

It possesses a microstructure with a ferrite base and austenite islands of various morphologies aligned in a certain orientation, as can be observed in figure 2.2

One of the most attractive characteristics of the cast duplex stainless steel (DSS) is the combination of specific strength, resistance to stress corrosion crack, lower cost, and resistance to pitting and crevice corrosion. The proper assessment of these properties requires an analysis at the micro and macro scales, as provided on the following sections.

2.5 Phase transformation

The final microstructure of a steel alloy is dependent on the thermal history throughout the fabrication process. During processes like casting, heat treatment, welding and so forth, material undergoes different thermal cycles which affect the number and type of microstructural features, their proportion and composition [28]. A particularly relevant example of phase transformation for DSS is the formation of austenite phase. Verily, DSS start solidification with 100% ferrite, however, upon cooling, austenite nucleates predominantly at ferrite grain boundaries, following favorable crystallographic orientations inside of the grains. Moreover, the final duplex microstructure is strongly dependent on the solidification sequence, e. g., while wrought DSS exhibits a ferrite matrix with austenite islands aligned to a certain direction, cast DSS contains austenite islands with varied morphologies [46, 49].

These transformations not only affect the δ/γ balance in the microstructure but also may lead to the formation of additional phases due to the high alloy content. These phases exhibit a relatively complex transformation kinetics and generally deteriorate the mechanical properties of DSS [68], some properties of these phases are shown in table 6.1. Therefore, to successfully address DSS performance under certain loading and environmental conditions it is essential to characterize secondary phases and their respective transformation kinetics. In this way, in the following subsections, the main secondary phases present in 25Cr-7Ni-Mo-N steel alloy are characterized. It follows a description of the essential transformation kinetics mechanisms and ways they may be accessed, in particular Time-Temperature-Transformation Diagrams and Continuous-Cooling-Transformation Diagrams.

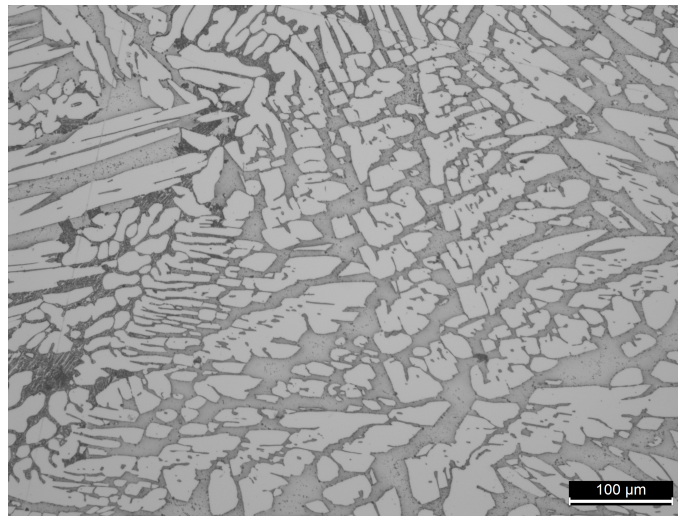


Figure 2.2: Sample of a microstructure image of the DSS.

2.6 Secondary phases

Due to the high amount of alloying elements presented in the duplex stainless steel, this material is prone to the formation of complex precipitation. A schematic Time-Temperature-Precipitation (TTP) diagram, shown in figure 2.3, demonstrates the correlation between alloy elements and aging time and temperature. Among these intermetallic precipitates, σ and χ phases are of particular interest, because these have been associated with drastic deterioration of the mechanical properties of the material. These phases usually result from improper cooling of the material in the temperature range of 600°C to 1000°C. In this way, understanding the characteristics and kinetics of σ and χ phases is fundamental to properly design manufacturing processes and heat treatments.

2.6.1 σ Phase

Sigma phase is an intermetallic precipitation phase with a tetragonal crystallographic structure that precipitates due to an eutectoid transformation mechanism [50, 13, 22]. The precipitation of σ phase and its distribution along the microstructure of the material depends on the temperature [59], as shown in the phase diagram in figure 2.4

2.6.2 χ Phase

Precipitation of the χ -phase may occur in austenitic, ferritic, and duplex stainless steels. χ -phase only occurs in ternary and quaternary systems [33] contrary of the σ -phase that can be present in binary Fe-Cr system. The precipitation of the χ -phase in thermodynamically not stable [59] and may transform itself in σ -phase. Several studies [8, 39, 33, 68, 59] show that the precipitation

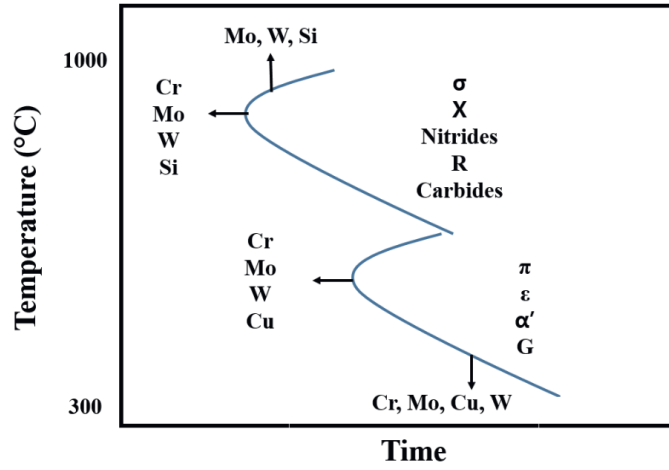


Figure 2.3: Schematic Time-Temperature precipitation diagram [26].

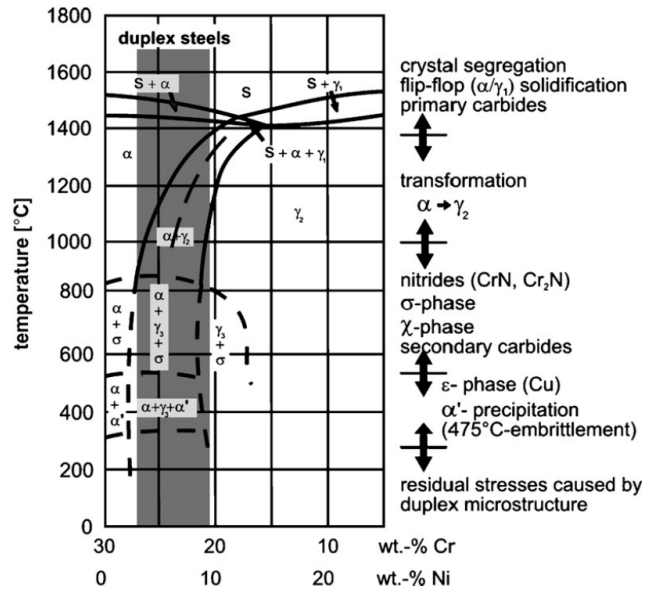


Figure 2.4: Phase diagram for 70% Fe in ternary Fe-Cr-Ni diagram [12].

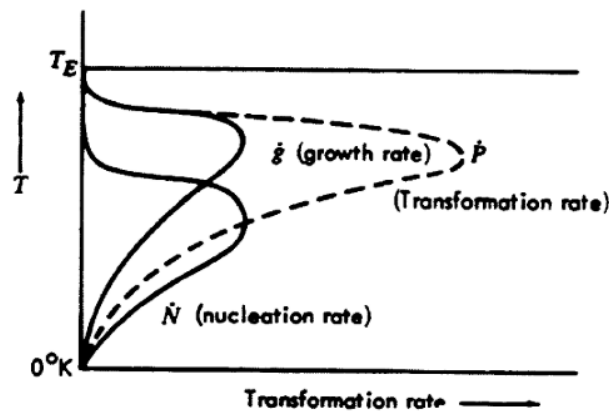


Figure 2.5: Growth and Nucleation rates by temperature [17].

and distribution of the χ -phase depends on the temperature. Additionally it occurs for lower aging times when compared to subsequently earlier precipitation of σ -phase, where the concentration of the first is reduced in favor of the second. In this way, DSS microstructure becomes a mixture of γ -phase, δ -phase, σ -phase and χ -phase.

Besides the formation of the intermetallic phases σ -phase and χ -phase, other precipitation like γ' -phase, tertiary austenite (γ_3), and chromium nitrides [37, 42, 61, 68] can be observed after aging at high temperatures. However, these phases have not been observed in the particular alloy studied in this work, and therefore their detailed description will be omitted.

2.7 Kinetics

2.7.1 Transformation sequence

Phase transformation in DSS depends on several mechanisms, namely, atomic diffusion, nucleation and growth. Diffusion is the most fundamental transformation mechanism, consisting in movement of certain atoms across the crystal lattice. It may be affected by concentration gradients, being inversely proportional to the atomic weight; alternatively, it may be affected by temperature and defects in the material such as dislocation and vacancies [46].

Next, in the onset of precipitation, nucleation consists on the formation of small groups of atoms into stable nuclei. [11]. After stabilization these nuclei start to grow, starting the growth phase. This growth depends on the cooling temperature as well as on the nucleation process (2.5).

Multiple microstructural features may be created during a certain thermomechanical process mainly due to different phase transformations. Properties related to the material can be explained

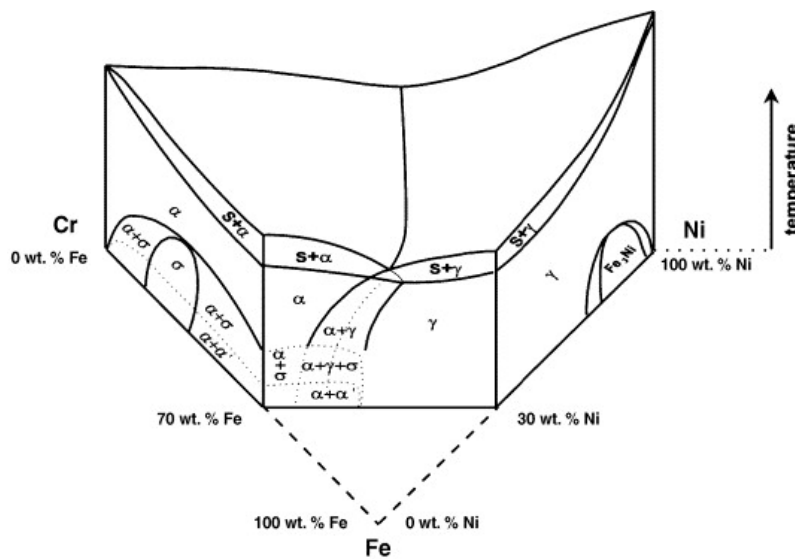


Figure 2.6: Fe-Cr-Ni phase diagram [59].

firstly by the composition of the present phases, together with their specific characteristics such as crystalline structure, and physical properties. Secondly, the distribution of these phases in the microstructure, their shapes, sizes, orientation, and interactions must also be inspected. Therefore, phase diagrams are a fundamental tool to understand the microstructural constitution. They show the relationship between the different phases in an equilibrium condition [52]. An example of this type of diagram is illustrated in the figure 2.6, displaying the different phases that may form in the ternary Fe-Cr-Ni system. A part of this diagram was used in section 2.6.1 to demonstrate the influence of temperature in phase transformation.

Nevertheless, as phase transformations are dynamic phenomena, depending on the thermal history of the processes, the information displayed by phase diagrams is complemented with information gathered in other types of diagrams. In this respect, time-temperature-transformation (TTT) diagrams and continuous-cooling-transformation (CCT) diagrams provide critical information to understand the phase transformation characteristics [47].

The difference between these two diagrams can be observed in the figure 2.7 where the TTT and CCT curves for Duplex Stainless Steel 2205 are plotted together showing the transformation for 1, 3, 5, and 10% σ -phase.

2.7.2 TTT - Time-Temperature-Transformation diagram

Time-Temperature-Transformation is a type of transformation diagram that relates temperature versus the time required for an isothermal phase transformation to occur. This type of diagram is critical to control the properties of a variety of materials. The outcome of an isothermal transformation through time results in a C-shape curve that predicts, at a fixed temperature, which fraction

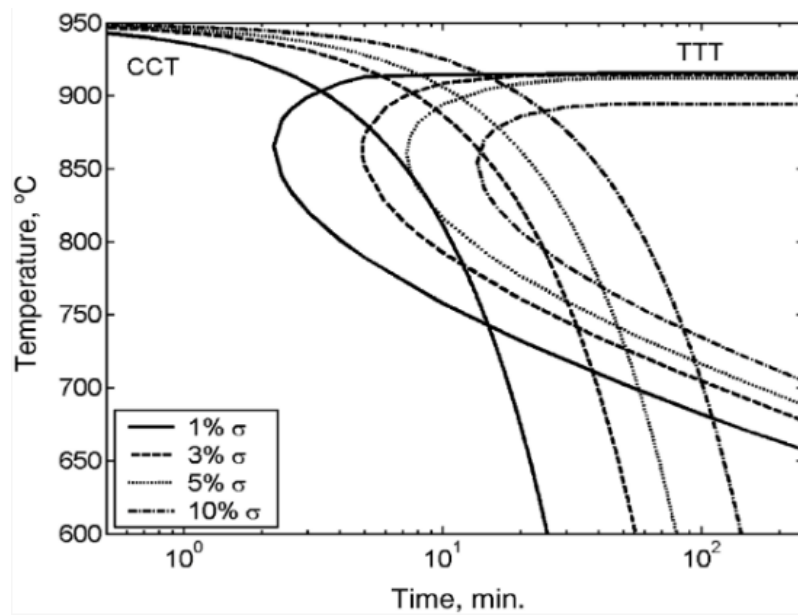


Figure 2.7: Time-Temperature-Transformation (TTT) and Continuous-Cooling-Transformation (CCT) curves for duplex stainless steel 2205 [25].

of the phase is going to transform over time, and is called the isothermal transition (IT) diagram. Transformation is faster in the so-called nose of the curve [47]. Taking into account different temperatures, the percentage of transformation versus the logarithm of the respective occurrence time results in an S-shaped curve, as illustrated in figure 2.8

For convenience, curves with different amounts of transformed material are plotted. They are generated from several isothermal experiments over a period of time. The amount of transformed material for each temperature and each time provides the data to construct the Time-Temperature Treatment diagram [68] (See figure 2.9).

2.7.3 CCT - Continuous-Cooling-Transformation

Continuous-Cooling-Transformation diagram relates both time and transformation changes under a continuously decreasing temperature, unlike the Time-Temperature-Transformation, which refers to isothermal aging. Therefore, CCT curves are widely used in industries for heat treatment and welding [47].

CCT diagrams are formed by plotting a series of cooling curves relating temperature in the y-axis and time in the x-axis as is shown in the figure 2.10. CCT curves also show the starting and finishing temperatures for the transformation [77]. In order to obtain the data needed to build the CCT diagram, a large number of experiments are required just like for the TTT diagram. For both graphs the image analysis process is fundamental to obtain the data needed to construct them.

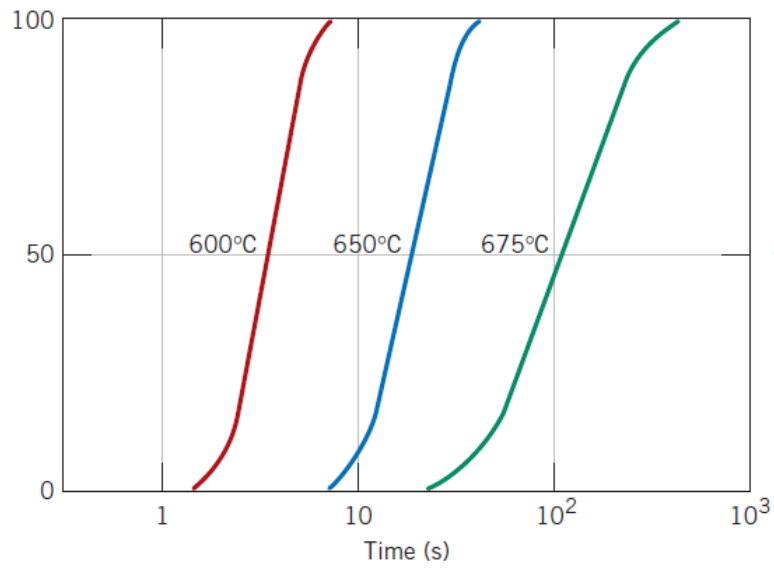


Figure 2.8: Isothermal fraction transformation versus logarithm of time [23].

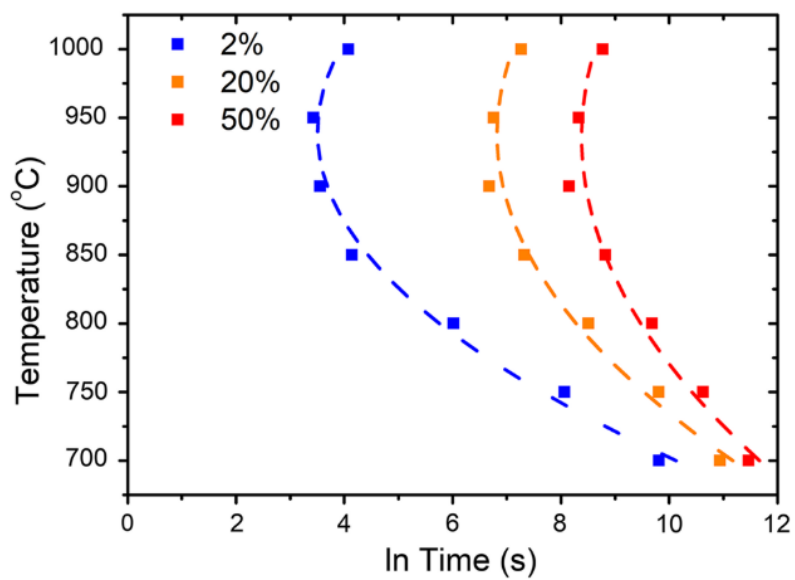


Figure 2.9: TTT curves for σ -phase precipitation [68].

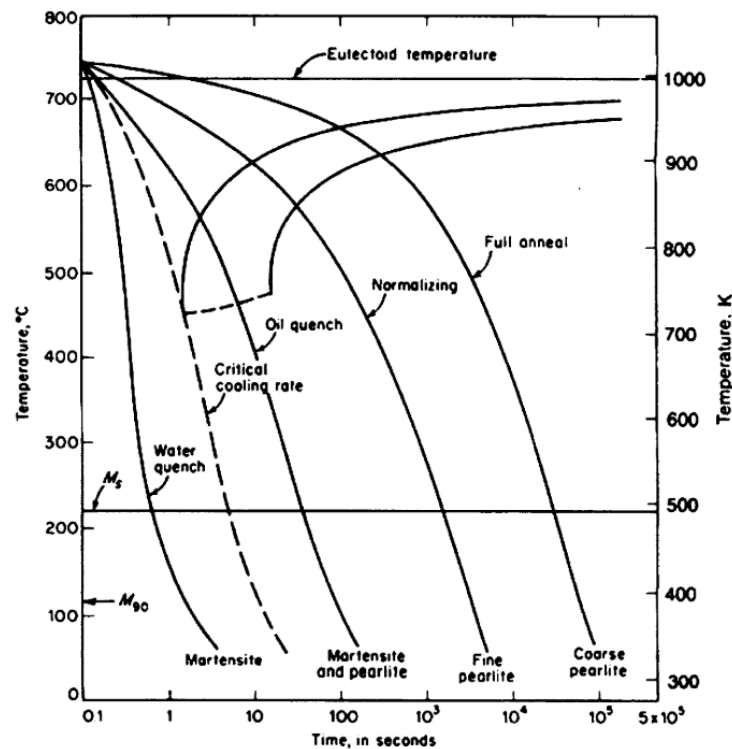


Figure 2.10: CCT Example [62].

2.8 Precipitation during fabrication and heat treatment process

As previously explained, due to the high content of alloy elements present in their composition, duplex stainless steel are prone to the formation of undesirable secondary phases caused by improper cooling during a heat treatment or fabrication processes.

Various experimental procedures with different setups are reported in the literature, as for instance, the isothermal aging proposed by [68], where the amount and distribution of secondary phases were observed to construct the TTT diagram shown in figure 2.9. From this diagram it can be observed that the time needed for the formation of the σ -phase decreases with the increase of temperature.

Observing the diagram for the Duplex Stainless Steel 2205 in figure 2.7 it becomes clear that a rapid cooling rate in higher temperatures is essential to avoid formation of precipitates. Even at the faster cooling rate presented in the mentioned graph, overcome in a 5% σ -phase formation when it reaches somewhere close to 10 minutes of cooling. The study of the behavior of the material at different temperatures is the main tool to prevent the formation of precipitates and, therefore, is of primordial importance the observation of the CCT and TTT diagrams for a proper design of manufacturing and heat treatment procedures. The current work focus on the casting manufacturing processes, more specifically sand casting process.

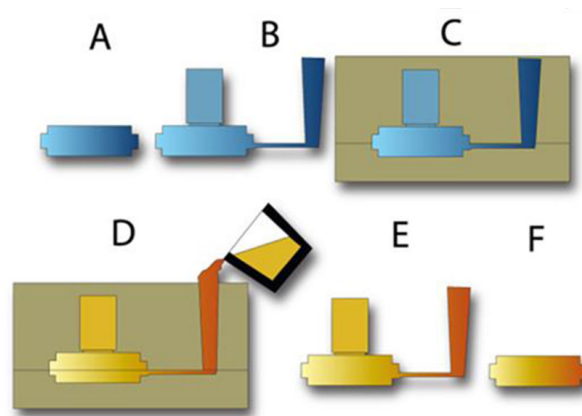


Figure 2.11: Illustration of casting process.

2.9 Sand Casting Process

Casting is a manufacturing process in which a liquid metal is poured into a mold and left to solidify (see figure 2.11). Secondary phase precipitation is especially influenced by the thickness of the mold and the respective molding material [68]. In general, in sand casting components, a slower cooling rate is obtained when compared to cast in permanent mold [73]. And thus the formation of intermetallic phases may be unavoidable. In this case, an analysis of the microstructure of the cast part is needed after its manufacturing, highlighting the proper distribution, amount and type of the formed phases.

With the information of the microstructure a correction process, such as heat treatment, can be developed to ensure the quality of the piece. This can be done, for example, by increasing the temperature to a point where only ferrite (δ -phase) and austenite (γ -phase) coexist. This biphasic region is achieved usually in temperatures higher than 1100°C [56]. After that, a drastic cooling is needed to sustain the microstructure and avoid the formation of undesirable secondary phases [38]. After the heat treatment, another microstructure analysis is necessary to evaluate the success of the treatment.

2.10 Macroscale properties

As discussed so far, the amount, shape, size and distribution of the secondary phases, have different impacts in the finished piece. In this way the intrinsic relationship between the microscale properties and the macroscale properties of the material becomes clear.

For example, the volume percentage of the σ -phase strongly affect toughness of the material [54, 15] due to its bad deformability [59]. Toughness is a property related to the ability of the

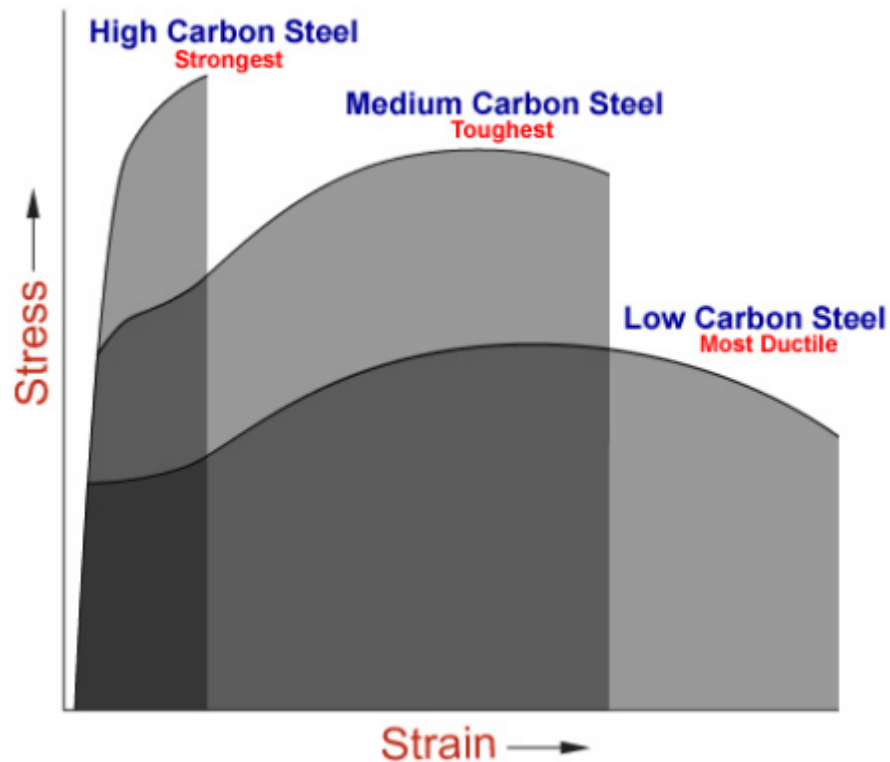


Figure 2.12: Stress-Strain curve [4].

material to plastically deform without fracturing. It is easy to observe the toughness of material in its stress-strain curve. This property is proportional to the area below the stress-strain curve (see figure 2.12). Moreover, the increase of σ -phase in general, decreases plastic properties in general for duplex stainless steel [71].

Hardness is another mechanical property that is influenced by the amount of precipitation in the composition of the microstructure. Hardness is the ability of a material to resist deformation [66]. This is an important characteristic for components such as rolling bearing, and gears [7].

The influence of the percentage of precipitation, up to level 5% of σ -phase, in the hardness is minimal [54]. But other studies [47, 54] have demonstrated that, once this threshold is overcome, hardness increase following the σ -phase formation.

In addition to the mechanical properties directly affected by the formation of secondary phases, residual stresses may be generated during the cooling process. Regions with different cooling rates within the same piece result in regions with different specific volumes leading to those residual stresses. Especially for sand casting parts, residual stresses may be observed during the cooling process of the material inside the mold, and may even remain at room temperature if the material yield limit is reached, affecting the results for future heat treatment [56]. Moreover, residual stress can cause distortion in the final product [20], and eventually lead to brittle fracture in the part [72].

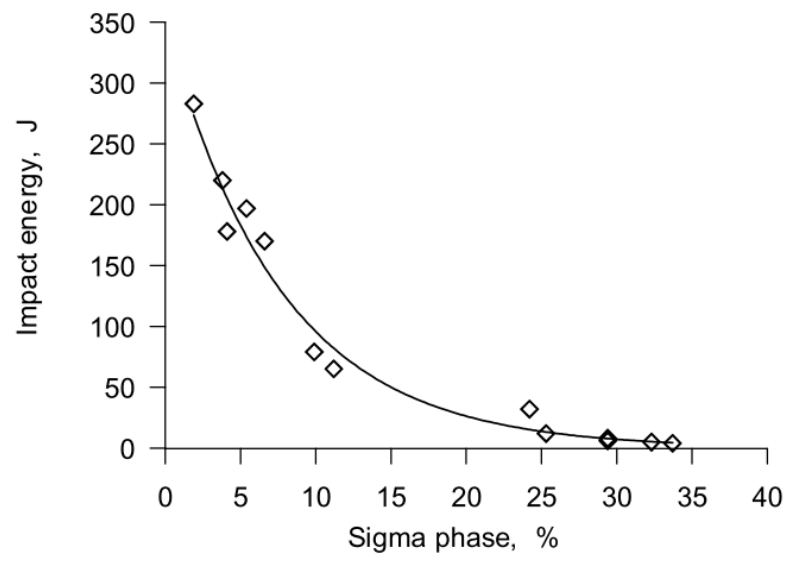


Figure 2.13: Impact of Sigma concentration [71].

Having highlighted the main features of DSS, in the next chapter the main tools for microstructure analysis will be presented.

Chapter 3

Microstructure Analysis

A good analysis of the microstructure of the material after the manufacturing process and eventual heat treatments is indispensable to predict the behavior and quality of the finished piece. Control of the microstructure requires knowledge of the phases that compose it, as well as knowledge of their impact in the macroscale, due to their composition, shape, particular properties, and quantities. This knowledge is essential for several manufacturing methods, allowing an optimization of material properties regarding a specific application [32].

Complex microstructures can lead to a slow, expensive, and time-consuming analysis, which reflects directly on the cost of the product. A standard analysis of the material structure was developed by the American Society for Testing and Material (ASTM), as the streamline method for measuring the grain size, called ASTM E112, and measurement of the volume fraction based on the point counting called ASTM E562. The latter, for example, uses the percentage of points in a image that fall in a given phase to give the percentage of that phase [24]. Nevertheless, as has been said, the amount of time and experience required to apply these manual methods is very expensive. Therefore, to overcome this limitations, some computer programs are currently used to improve the speed and repeatability of existing manual analysis methods [24], such as OOF and ImageJ described next.

OOF is a public domain software used to investigate properties of microstructure as second phases, cracks, pores, and others, through tools that allow the user to graphically select features in the microstructure image and assign properties to it [5]. The main drawback of this program is the time-consuming process of the tools presented by OOF to designate the properties of the different phases in the microstructure, driving a need to better automation of the entire process of image segmentation. In addition OOF allows the construction of FEM meshes, but because it is a program focused on image analysis, this tool does not contain the additional resources that specific FEM programs have.

ImageJ is also a public domain program that allows editing, analyzing and processing images. Unlike the OOF software, which has the specific purpose of evaluating microstructures, ImageJ can calculate pixel values and user-defined selection areas, as well as can be used in standard image processing tools, such as edge detection and filtering [2]. All of these features can be of great help in the image analysis process, but they do not contribute to the automation of the process when it comes to processing multiple images in a fast and accurate way.

Both commercial softwares have limitations regarding the requirements of this project, which justify the development of an in-house software. Thus, the next section presents an overview of some of the techniques that can be used to overcome the necessities of the current work.

3.1 Image Segmentation

Image segmentation is a method that may be used to investigate features contained in microstructure images. It is a digital image processing technique for partitioning a digital image into multiple segments in order to recognize them as objects. It is widely used in biology research [16, 31, 36, 58] to determine boundaries and locate objects. There are many studies with different methods and techniques for image segmentation [9, 29, 35, 75, 76], though most of them can be classified into three different groups namely, characteristics feature threshold, edge detection, and region extraction [35].

These three groups are vast, reflecting a large amount of research on image segmentation carried throughout the years, from them several methods have emerged. Within this, seven different categories may be highlighted [64]:

- Characteristics Feature Threshold {
 - Histogram thresholding
 - Clustering
 - Fuzzy approaches
 - Neural network and Generic Algorithm

- Region Extraction {
 - Region growing, region splitting and merging
 - Physical model-based

- Edge Detection {
 - Edge-based

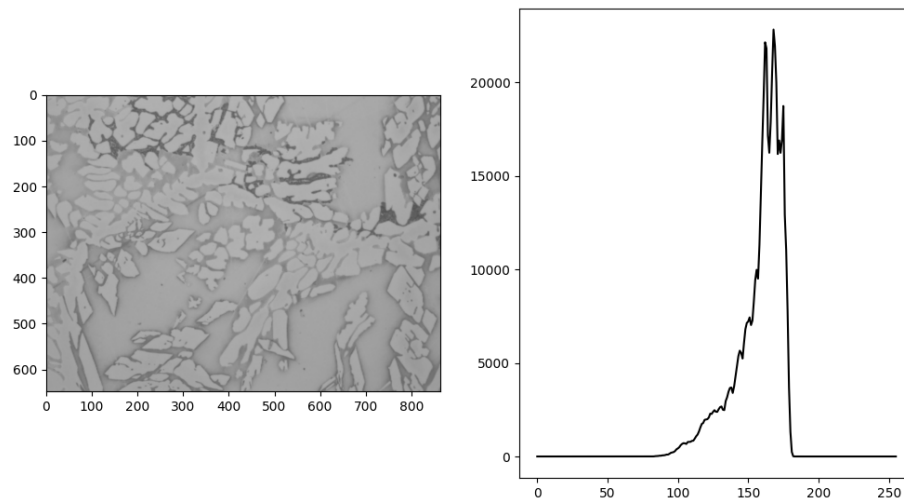


Figure 3.1: Histogram of a microstructure image.

The choice between methods that better address a particular problem with smaller computational cost is not an easy task. Nevertheless, several approaches can be combined in an image segmentation problem to obtain high-quality results.

3.1.1 Image Segmentation Technics

One of the most used and simultaneously simplest methods of image segmentation is thresholding [9]. It consists of selecting threshold values, which allow the conversion of the image into a binary image [69]. These threshold values may be manually chosen or a threshold algorithm may be used to compute them automatically. For noiseless images with a uniform background, the first methods can be applied, but for most cases, it becomes cumbersome [70]. Therefore, a most sophisticated approach has to be implemented using, for instance, a histogram of the target image.

A histogram (figure 3.1) plots the distribution of the pixel intensity, and may be used to observe the distribution of values between background and bodies. From these values, it is possible to segment the image into different groups of pixels, one for the background and another for the body group. Some iterative processes can be applied to define clear values and overcome problems such as noisy images. For example, a threshold initial value can be chosen manually for a first segmentation of the image into a background and bodies pixel groups, and, after that, an average of the previous values is computed and a new threshold value is calculated. The process is repeated with the new value until convergence is reached [70].

The segmentation performed by the threshold value is of binary type, assigning values of 0 or 1 to each pixel group. This binary segmentation algorithm can be used in various applications together with some cleanup technics, but it has its limitations when applied to multi-class datasets

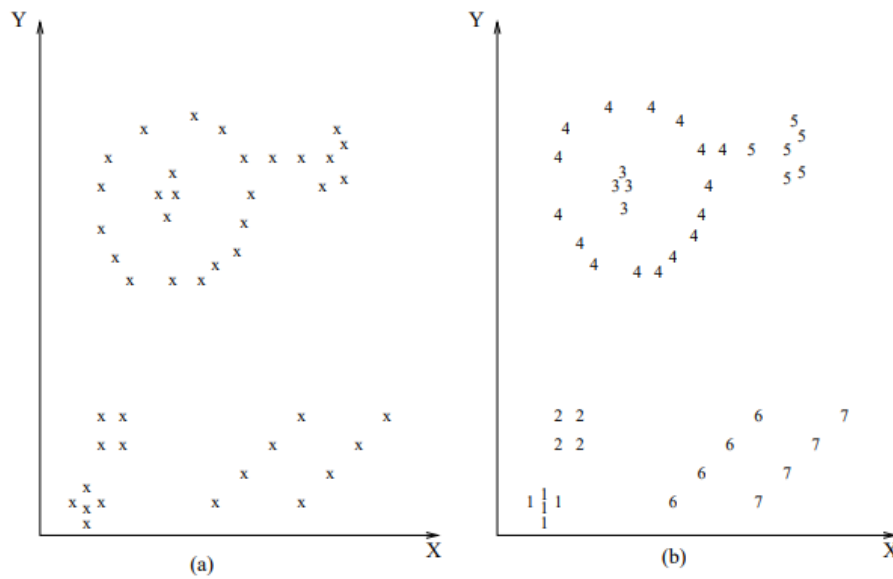


Figure 3.2: Clustering example [44].

[67]. For this reason, thresholding is commonly used as a first step for the image segmentation process, corresponding to separate the background pixels from the rest of the image [10].

Clustering is another powerful technic for image segmentation that can be described as "the process of organizing objects into groups whose members are similar in some way" [48]. Some of the most common clustering methods are *k mean*, *improved k mean*, *fuzzy c mean* and *improved fuzzy c mean* [64].

The simplest clustering method is the *k mean*. It is a partitioning method that finds the best *k* partition of the various objects in the image [53]. It is based on the similarity between objects, calculating the Euclidean distance between the *k* initial centroid, previously chosen, and each sample, and assigning the sample one by one to the *k* cluster [45, 51]. In other words, it tries to assign one point to the closest cluster based on the sum of the square distance from the cluster's centroid. Some input data and the desired clustering for it are shown in the figure 3.2.

Another important region-based method to image segmentation is the Watershed Algorithm [19]. In this case, the image is thought of as being a topographic landscape composed with valleys and peaks [60] depending on the intensity of the corresponding pixel, where bright points are assigned to peaks and dark points to valleys [63]. With the brightness of each pixel representing its height, the Watershed Algorithm determines the line that goes through the top. One of the bigger problems with this algorithm is oversegmentation [18, 24], nevertheless, it can be overcome by applying morphological filter [18]. The most basic morphological operators are Dilatation and Erosion, operators that act in the input image by basically adding or removing pixels to the boundaries of objects in the image. Applying dilatation to an image, it makes objects more visible and fills small holes in it, while using erosion operator removes small islands and bodies clearing

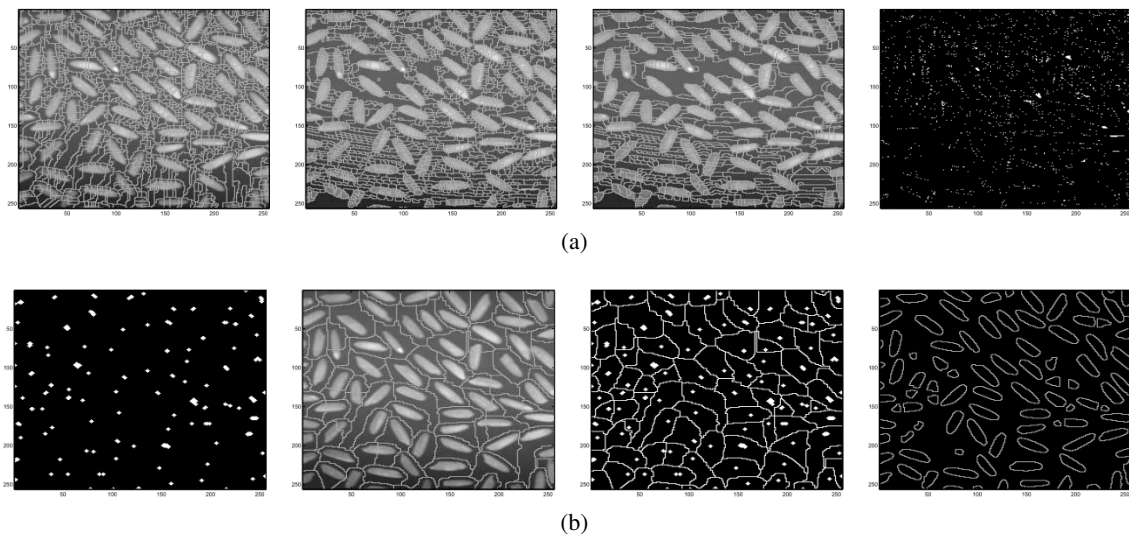


Figure 3.3: Classical segmentation technique using morphological operators.

substantive objects. A classical segmentation using morphological operators is presented by [18], where through the images, the process of segmentation of the bodies become clear. This process is illustrated in figure 3.3.

This example demonstrate how to handle oversegmentation problem by taking the watershed of the original image, the watershed of its gradient, and showing the segmentation result and minima of the original image 3.3a. After that, this filtered image is used. The minima of the filtered image are displayed in the first picture of figure 3.3b, followed by the new influence zones and finally the segmented image after all the process.

All segmentation technics have their own advantages and drawbacks, making them useful to a specific image segmentation problem. In general, different methods can be used in a stack form, together with pre- and post-processing technics to overcome the challenges of image segmentation. The current work proposes a script that is easy to understand and that can be modified according to specific needs, which will be presented on chapter 5.

Chapter 4

FEM - Finite Element Method

Finite Element Method (FEM) is a powerful numerical tool that may be used to solve differential equations that describe diverse physical phenomena [43]. For this particular case, the FEM is used in the computed simulation to solve the stress analysis of the proposed multiscale model. Basically, the domain of the problem is divided into a mesh that contains a number of finite elements attached to each other by nodes. The main interest of the problems are the unknown variables contained in the nodes of the mesh.

Several factors influence the number of nodes and elements inside the mesh. The correlation between precision and the computational cost is one of the main factors. Nevertheless, the geometry and complexity of the problem in analysis, act proportionally in relation to size and quantity of elements.

In this chapter the basic features of the FEM are outlined. Furthermore, as the FEM is also employed for multiscale analysis purposes. Therefore, the homogenization method used to derive elastic properties at the macroscale from microstructural features is explained.

4.1 Formulation

The Finite Element Method in this section is formulated from the strong form, the governing equation of the problem, and the weak form, which is an integral form of this equation [43]. Engineering problems are often stated using differential equations. In particular, structural problems are described by the following equation:

$$\nabla \cdot \sigma - b = 0 \tag{4.1}$$

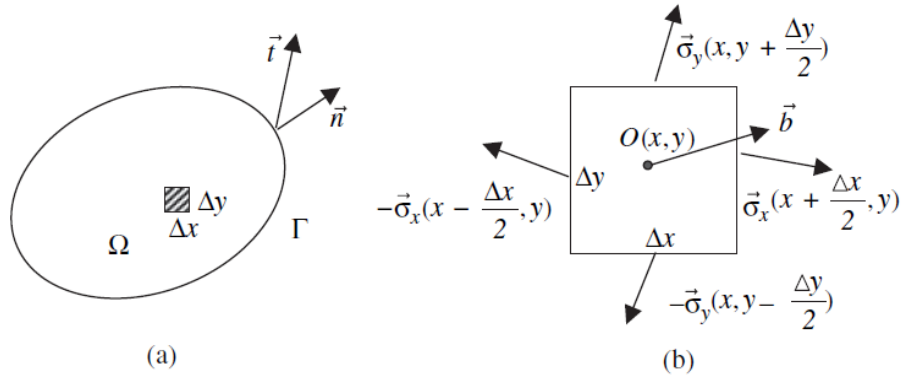


Figure 4.1: (a) Domain of the problem and (b) traction vectors acting on the infinitesimal element in a 2D problem [43].

where b is the body force vector and σ is the tensor stress acting on an infinitesimal element.

Also called strong form, this equation presented together with the boundary conditions, reflects the balance of internal and external forces of the body illustrated in the figure 4.1. A brief explanation of this is presented in section 4.1.2.

4.1.1 Kinematics

The problem addressed in the current work focus mainly in the linear relationship between the stress and strain, which is called constitutive equation. Thus the constitutive equations for elasticity at the macroscale is given by:

$$\sigma = D\varepsilon \quad (4.2)$$

where σ is the vector, that represent the force components per unit area acting on the plane in the 2D case. D is called elasticity matrix, that is a 3×3 matrix for a 2D problem, whether one assumes plane stress or plane strain condition. For isotropic materials, the elasticity matrix D is written as

$$D = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1 - \nu)/2 \end{bmatrix} \quad (4.3)$$

for plane stress, and for plane strain it has the following form:

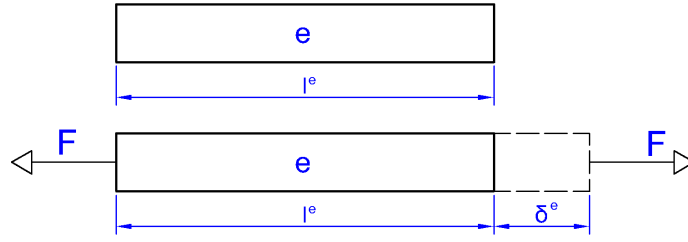


Figure 4.2: Element elongation.

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & (1-2\nu)/2 \end{bmatrix} \quad (4.4)$$

in turn, ε is the strain vector that is defined as the ratio of the elongation δ^e by the original length, where the subscript e in the equation is related to the element:

$$\varepsilon^e = \frac{\delta^e}{l^e} \quad (4.5)$$

Each segment has an original length of Δx and Δy and the elongation of it is represented by the difference between displacements in apposite edges of the infinitesimal element $u_x(x + \Delta x, y) - u_x(x, y)$ and $u_y(x, y + \Delta y) - u_y(x, y)$ for x a y direction respectively. So the strain becomes:

$$\begin{aligned} \varepsilon_{xx} &= \lim_{\Delta x \rightarrow 0} \frac{u_x(x + \Delta x, y) - u_x(x, y)}{\Delta x} = \frac{\partial u_x}{\partial x} \\ \varepsilon_{yy} &= \lim_{\Delta y \rightarrow 0} \frac{u_y(x, y + \Delta y) - u_y(x, y)}{\Delta y} = \frac{\partial u_y}{\partial y} \end{aligned} \quad (4.6)$$

The shear strain γ_{xy} is related to the angle change in between the two unit vectors in the x and y directions, as follows:

$$\gamma_{xy} = \lim_{\Delta x \rightarrow 0} \frac{u_y(x + \Delta x, y) - u_y(x, y)}{\Delta x} + \lim_{\Delta y \rightarrow 0} \frac{u_x(x, y + \Delta y) - u_x(x, y)}{\Delta y} = \frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y} \quad (4.7)$$

And thus, strain is assembled in the vector form as:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{bmatrix} \quad (4.8)$$

Knowing that for a two dimensional problem the displacement vector is composed for the two components in the direction x and y,

$$\boldsymbol{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad (4.9)$$

the strain vector described in 4.9 can be written in terms of the derivatives of displacement

$$\boldsymbol{\varepsilon} = \nabla_s \boldsymbol{u} = \nabla_s \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad (4.10)$$

where ∇_s is a symmetric gradient matrix operator, given by:

$$\nabla_s = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \quad (4.11)$$

4.1.2 Equilibrium Equations

Considering the body on equilibrium shown on figure 4.1, the equilibrium equations for the element can be written as [43]

$$\begin{aligned} -\vec{\sigma}_x \left(x - \frac{\Delta x}{2}, y \right) \Delta y + \vec{\sigma}_x \left(x + \frac{\Delta x}{2}, y \right) \Delta y \\ -\vec{\sigma}_y \left(x, y - \frac{\Delta y}{2} \right) \Delta x + \vec{\sigma}_y \left(x, y + \frac{\Delta y}{2} \right) \Delta x \\ + \vec{b}(x, y) \Delta x \Delta y = \vec{0} \end{aligned} \quad (4.12)$$

Dividing the equation 4.12 by $(\Delta x \Delta y)$ and taking the limits as $\Delta x \rightarrow 0$ and $\Delta y \rightarrow 0$, the following equations are obtained:

$$\lim_{\Delta x \rightarrow 0} \frac{\vec{\sigma}_x(x + \frac{\Delta x}{2}, y) - \vec{\sigma}_x(x - \frac{\Delta x}{2}, y)}{\Delta x} = \frac{\partial \vec{\sigma}_x}{\partial x} \quad (4.13)$$

$$\lim_{\Delta y \rightarrow 0} \frac{\vec{\sigma}_y(x, y + \frac{\Delta y}{2}) - \vec{\sigma}_y(x, y - \frac{\Delta y}{2})}{\Delta y} = \frac{\partial \vec{\sigma}_y}{\partial y} \quad (4.14)$$

So, the equilibrium equation becomes

$$\frac{\partial \vec{\sigma}_x}{\partial x} + \frac{\partial \vec{\sigma}_y}{\partial y} + \vec{b} = \vec{0} \quad (4.15)$$

It should be noticed that vectors $\vec{\sigma}_x$, $\vec{\sigma}_y$, and \vec{b} are composed as follows:

$$\vec{\sigma}_x = \begin{bmatrix} \sigma_{xx} \\ \sigma_{xy} \end{bmatrix} \quad (4.16)$$

$$\vec{\sigma}_y = \begin{bmatrix} \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} \quad (4.17)$$

$$\vec{b} = \begin{bmatrix} b_x \\ b_y \end{bmatrix} \quad (4.18)$$

Thus, in terms of components, the equilibrium equations are given by:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + b_x = 0 \quad (4.19)$$

$$\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + b_y = 0 \quad (4.20)$$

Then, assembling the linear system of equations in matrix form by introducing the following symmetric gradient matrix operator from 4.11

$$\nabla_s^T = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}, \quad \boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} \quad (4.21)$$

it becomes

$$\nabla_s^T \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \quad (4.22)$$

4.1.3 Strong and Weak Form

In order to define the boundary conditions the domain boundary is divided into two different parts: one part of the boundary where prescribed tractions are applied, denoted by Γ_t , and another part where displacements are prescribed, denoted by Γ_u . These are complementary boundaries, thus:

$$\Gamma_u \cup \Gamma_t = \Gamma \quad \Gamma_u \cap \Gamma_t = \emptyset \quad (4.23)$$

The boundary conditions of the traction boundary may be written as

$$\sigma_x \cdot n = t_x \quad \text{and} \quad \sigma_y \cdot n = t_y \quad \text{on} \quad \Gamma_t \quad (4.24)$$

And for the displacement as

$$u = \bar{u} \quad \text{on} \quad \Gamma_u \quad (4.25)$$

Summarizing everything the strong form for the linear elasticity problem can be defined as

- (a) $\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0}$ on Ω
- (b) $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon}$
- (c) $\sigma_x \cdot n = t_x$ and $\sigma_y \cdot n = t_y$ on Γ_t

(d) $u = \bar{u}$ on Γ_u

The weak form is an integral form of equation (a), obtained by multiplying it by an arbitrary weight function, w , and integrating over the domain [43], as follows:

$$-\int_{\Omega} (\nabla \cdot \boldsymbol{\sigma}) w d\Omega = \int \mathbf{b} w d\Omega \quad (4.26)$$

The constitutive law 4.2 and the relation between the strain and the displacement 4.10, demonstrate that $(\nabla \cdot \boldsymbol{\sigma})$ have second order derivatives of the unknown u , thus, applying the Green Theorem

$$\int_{\Omega} (\nabla \cdot \boldsymbol{\sigma}) w d\Omega = \int_{\Omega} \boldsymbol{\sigma} \nabla w d\Omega - \int_{\Gamma} (\boldsymbol{\sigma} \mathbf{n}) w d\Gamma \quad (4.27)$$

Where the quantity $(\boldsymbol{\sigma} \mathbf{n})$ is known as the traction or stress vector. In this way, equation 4.26 can be written in the weak form as:

$$\int_{\Omega} \nabla w \boldsymbol{\sigma} d\Omega = \int_{\Gamma} \bar{\mathbf{t}} w d\Gamma + \int \mathbf{b} w d\Omega \quad (4.28)$$

4.1.4 Finite Element Discretization

In order to solve equation 4.28 it is necessary to define trial functions, which depend on the type of element in which the domain is discretized. Considering the domain Ω with boundary Γ discretized with two-dimensional three-node triangular elements, as illustrated in figure 4.3, for triangular element with nodes 1, 2 and 3 as shown in the figure 4.4, the nodal displacements are:

$$\mathbf{u} = \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix} \quad (4.29)$$

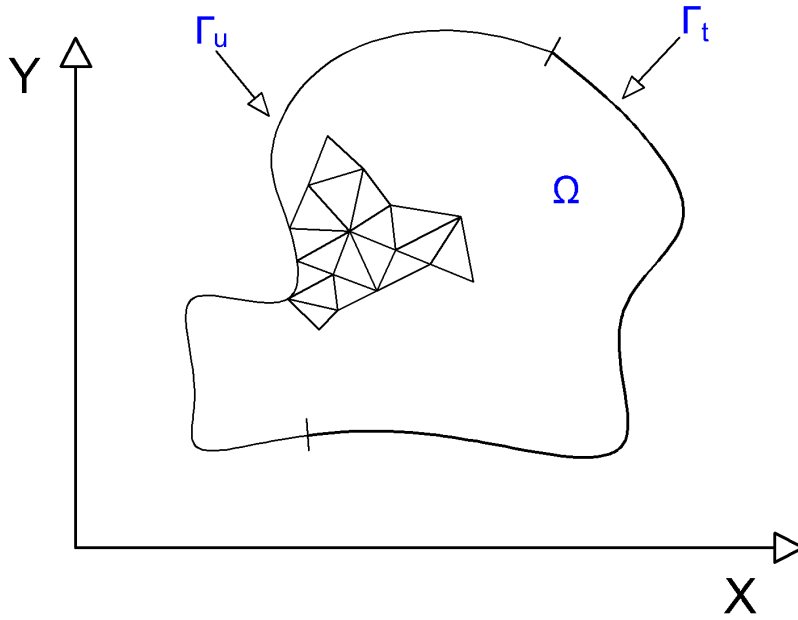


Figure 4.3: Finite element discretization in two dimensions.

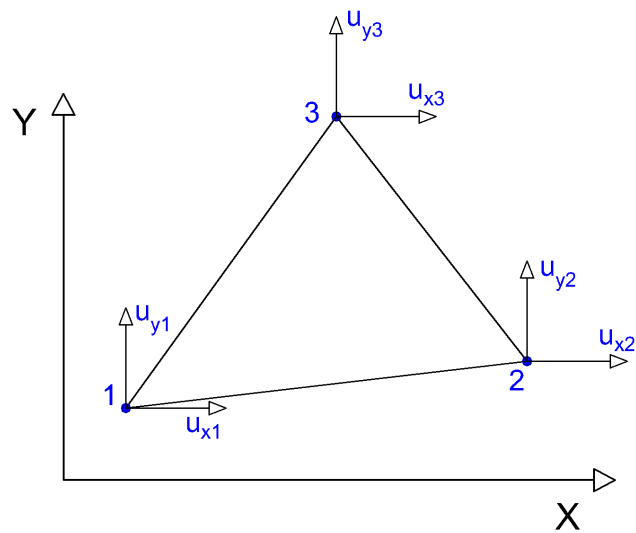


Figure 4.4: A Single Triangular Element.

The interpolation process of the displacement field can be done by the interpolation functions, usually called shape functions. So the finite element approximation of the trial solution and the weight function on each element can be, respectively expressed by:

$$\begin{bmatrix} u_x^e \\ u_y^e \end{bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{bmatrix} u_{x1}^e \\ u_{y1}^e \\ u_{x2}^e \\ u_{y2}^e \\ u_{x3}^e \\ u_{y3}^e \end{bmatrix} = [N][u^e] \quad (4.30)$$

$$\begin{bmatrix} w_x & w_y \end{bmatrix}^T = \begin{bmatrix} w_{x1} \\ w_{y1} \\ w_{x2} \\ w_{y2} \\ w_{x3} \\ w_{y3} \end{bmatrix}^T \begin{bmatrix} N_1 & 0 \\ 0 & N_1 \\ N_2 & 0 \\ 0 & N_2 \\ N_2 & 0 \\ 0 & N_2 \end{bmatrix}^T = [w]^T [N]^T \quad (4.31)$$

Now the displacement can be related to strain

$$[\varepsilon] = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_y}{\partial y} \\ \frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y} \end{bmatrix} \quad (4.32)$$

Substituting the shape functions in the vector for each one of the nodes of the element.

$$\varepsilon = \begin{bmatrix} \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial x} u_{xi} \\ \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial y} u_{yi} \\ \sum_{i=1}^{n_e} \frac{\partial N_i}{\partial y} u_{xi} + \frac{\partial N_i}{\partial x} u_{yi} \end{bmatrix} \quad (4.33)$$

The strain vector can be written as

$$\begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix} \rightarrow [\boldsymbol{\varepsilon}] = [\mathbf{B}][\mathbf{u}] \quad (4.34)$$

For the derivatives of weight functions

$$(\nabla \mathbf{w})^T = (\mathbf{B}\mathbf{w})^T = \mathbf{w}^T \mathbf{B}^T \quad (4.35)$$

Therefore, the integral over the domain of an element for the weak form presented before can be written using the equations 4.35, 4.34, 4.31 ,and 4.30, as:

$$\sum_{e=1}^{n_{el}} \mathbf{w}^e \mathbf{u}^e \left[\int_{\Omega^e} \mathbf{B}^{eT} \mathbf{D}^e \mathbf{B}^e d\Omega - \int_{\Gamma_t^e} \mathbf{N}^{eT} \bar{\mathbf{t}} d\Gamma - \int_{\Omega^e} \mathbf{N}^{eT} \mathbf{b} d\Omega \right] = 0 \quad (4.36)$$

Discarding the trivial solution

$$\mathbf{w}\mathbf{u} = 0 \quad (4.37)$$

The stiffness matrix is identified as being

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega \quad (4.38)$$

And the external force matrix is

$$\mathbf{F}_{\text{ext}} = \int_{\Gamma_t} \mathbf{N}^T \bar{\mathbf{t}} d\Gamma + \int_{\Omega} \mathbf{N}^T \mathbf{b} d\Omega \quad (4.39)$$

So the equation can be written as

$$\mathbf{F}_{\text{ext}} = \mathbf{K}\mathbf{u} \quad (4.40)$$

4.2 Triangular Element

Results coming from the finite element analysis of a particular element corresponds to the displacement calculated in the nodes of this element. After that, the stress and strain can be calculated for the integration points that the element contains. Abaqus calculates and interpolates the results to present the user with the value of interest that would be the stress and strain on the element nodes. The image 4.4 illustrates the triangular three-node element with nodes numbered counterclockwise and with two degrees of freedom. This is a linear displacement element [43] that is used in the current work.

4.2.1 Isoparametric Element

Typically in FEM isoparametric elements are used, in which the displacement field and geometric coordinates are approximated using the same shape function. Furthermore these shape functions are defined in a local set of coordinates where integrations are performed using Gauss quadrature. The mapping process to a different set of coordinates is illustrated in figure 4.5.

4.2.2 Shape Functions

Assuming the cartesian coordinate system for the triangle element, the mapping coordinate system represented by ξ and η is chosen such that the coordinates of the nodes 1, 2, and 3 are $(0,0)$, $(1,0)$ and $(0,1)$ respectively, this system maps the irregular triangular element to a regular shape. As it stated before the same shape functions are used to interpolate the node values of displacement (or coordinates) to the Gauss Point of the element following the equations 4.41 and 4.42

$$u_x = N_1u_{x1} + N_2u_{x2} + N_3u_{x3} \quad (4.41)$$

$$u_y = N_1u_{y1} + N_2u_{y2} + N_3u_{y3} \quad (4.42)$$

where the shape functions are define as

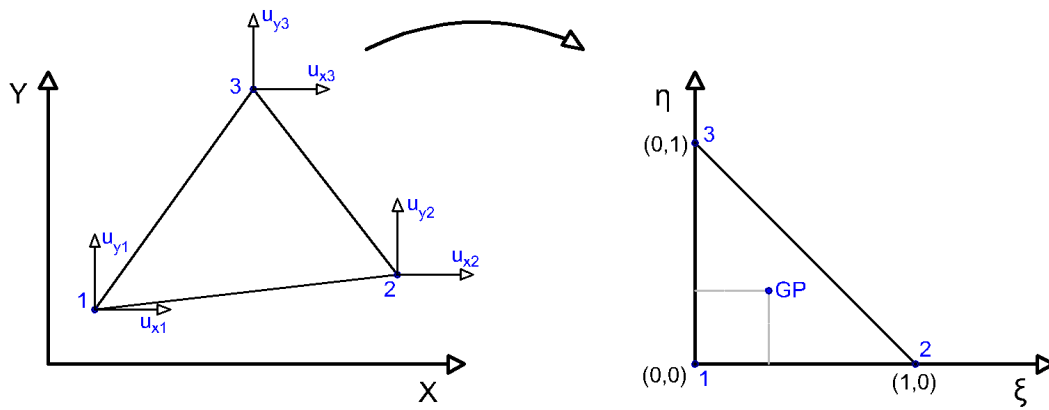


Figure 4.5: Mapping the Triangular Element.

$$N1 = 1 - \xi - \eta \quad (4.43)$$

$$N2 = \xi \quad (4.44)$$

$$N3 = \eta \quad (4.45)$$

4.2.3 Jacobian

As it was pointed the isoparametric elements allow the use of regular elements described in the natural coordinate space composed by ξ and η . In order to obtain the results in the real space the Jacobian is used. The Jacobian is a matrix that relates the real space coordinates and the natural space coordinates. The Jacobian is defined as follows:

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \xi} \quad (4.46)$$

$$\frac{\partial N_i}{\partial \eta} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \eta} \quad (4.47)$$

in matrix form

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}}_{\text{Jacobian Matrix}} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} \quad (4.48)$$

4.2.4 Gauss Quadrature

Equations 4.38 and 4.39 requires numerical integration. Gauss Quadrature is a numerical integration technique very efficient for polynomial functions [43], so it is especially adequate for FEM employing polynomial approximation functions. In the this integration procedure such integral is approximated by a set of weights and integration points, which are chosen so the higher possible polynomial is integrated exactly. In the case of the three nodes triangular element, only one integration point, or Gauss Point, is required to produce accurate integration for the shape functions used. The integration formula is given by:

$$I = \int_{\Omega^e} f(\xi, \eta) d\Omega^e = \sum_{i=1}^{n_{gp}} f(\xi, \eta) W_i |J^e(\xi, \eta)| \quad (4.49)$$

Therefore, the numerical form of the element stiffness matrix becomes:

$$K^e = \int_{\Omega^e} \mathbf{B}^{eT} \mathbf{D}^e \mathbf{B}^e d\Omega = \int_1^{-1} \int_1^{-1} \mathbf{B}^{eT} \mathbf{D}^e \mathbf{B}^e |J| d\xi d\eta \quad (4.50)$$

$$= \sum_{i=1}^2 \sum_{j=1}^2 W_i W_j |J^e(\xi_i \eta_j)| \mathbf{B}^{eT}(\xi_i \eta_j) \mathbf{D}^e \mathbf{B}^e(\xi_i \eta_j) \quad (4.51)$$

For the present work, this equation is solved by the Abaqus software. The real interest in the development of the Finite Element Method is to find the elasticity matrix for the macro material under study, based on the results of stresses obtained by Abaqus for a representative volume element of the microstructure (RVE).

4.2.5 Homogenization Method

Micromechanics accounts explicitly for the heterogeneities within the microstructure of a given material. Nevertheless, at a higher length scale, the material in question may be treated as a homogeneous continuum. Therefore, the finite element formulation presented constitutes the base to homogenize the micro properties of the material and obtain the respective macro properties. Considering a representative volume, which captures the essential features of the microstructure in study, the average stress and strain tensors are given by:

$$\bar{\sigma}_{ij} = \frac{1}{V} \int_V \sigma_{ij} dV \quad (4.52)$$

$$\bar{\varepsilon}_{ij} = \frac{1}{V} \int_V \varepsilon_{ij} dV \quad (4.53)$$

where V designates the volume of the RVE. These average tensors may be related in order to calculate a macro elasticity matrix:

$$\bar{\sigma} = \mathbf{D}\bar{\varepsilon} \quad (4.54)$$

alternatively in matrix form:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \end{bmatrix} \quad (4.55)$$

The finite element formulation defined in the previous sections may be used to calculate each element of the D matrix. For this, the RVE is discretized into a finite element mesh and appropriate boundary conditions are applied. For instance, to calculate the elements in the first column of the matrix D , an unitary deformation is applied to the RVE in the x-direction, as follows:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (4.56)$$

After matrix multiplication the following equations are obtained:

$$\sigma_{11} = D_{11} \quad (4.57)$$

$$\sigma_{22} = D_{21} \quad (4.58)$$

$$\sigma_{12} = D_{31} \quad (4.59)$$

The above equations may be solved using the FEM, as follows. Firstly, each stress component is calculated at the element level. The stress at each Gauss point is interpolated from the nodal

values using the shape functions of the element. For the triangle presented in section 4.2.2 the values for the weight and the quadrature points are $\frac{1}{2}$ and $\frac{1}{3}$ respectively. Then, the contributions of all elements are averaged and thus, for an applied displacement in x-direction, the global stress equations become:

$$D_{11}^e = \frac{1}{V} \sum_{i=1}^{n_{elem}} (\sigma_{11,1}^e N_1 + \sigma_{11,2}^e N_2 + \sigma_{11,3}^e N_3) * W * \det[J] \quad (4.60)$$

$$D_{22}^e = \frac{1}{V} \sum_{i=1}^{n_{elem}} (\sigma_{22,1}^e N_1 + \sigma_{22,2}^e N_2 + \sigma_{22,3}^e N_3) * W * \det[J]$$

$$D_{12}^e = \frac{1}{V} \sum_{i=1}^{n_{elem}} (\sigma_{12,1}^e N_1 + \sigma_{12,2}^e N_2 + \sigma_{12,3}^e N_3) * W * \det[J]$$

Where D_{11} represent the first element of the elasticity matrix D , n_{elem} is the number of elements in the finite element mesh, $\sigma_{11,1}^e$ is the stress in the x-direction in node 1, $\sigma_{11,2}^e$ in node 2, and $\sigma_{11,3}^e$ in node 3. W is the weight and $\det[J]$ is the determinant of the jacobian. This method can be sequentially applied to calculate the remaining elements of the elasticity matrix. As a result, the presented formulation will be used in further steps.

Chapter 5

Matlab Code and Image Processing

This thesis is inserted in a broader project, which aims building a multiscale model to describe the behavior of DSS. One of the cornerstones within this larger project is the transcription of visual experimental data into data that can be used for numerical simulation, in particular in FEM simulations. In order to achieve this goal, a software which recognizes relevant features in images of material micro structures was developed. Moreover, a whole procedure to convert the relevant information into a finite element mesh was also implemented. In turn, the finite element meshes representative of the microstructure of the material are instrumental to establish a relation between relevant micro features and the macro properties of the material. Thus, in this chapter the developed methodology is described.

The proposed method entails several steps. It starts by a complete image first analysis based on a multilayer image segmentation method. This step is followed by a segmented imaged analysis in order to optimize the process. Finally, a methodology to convert the segmented images in relevant finite element meshes is proposed. The whole process is applicable to any kind of images, however this project targets optical microscopy images and scanning electron microscope (SEM) images.

A schematic diagram of the process described in this section is illustrated in figure 5.1. It shows that the program starts by importing the entire image, after that a quality check is made and the result demonstrates the need for some pre-processing by filters on the image. After that, the full image passes through the processes of phase segmentation and the respective percentages are stored. A section of the image is cropped and the same process of phase segmentation is applied comparing the results of the percentage of each phase of the sectioned image with the previous results for the full image. If the percentages do not match, the sectioning of the image is done again. With the same percentage for each phase, the body recognition can be made and DXF files are generated. All this steps are going to be detailed in the next sections.

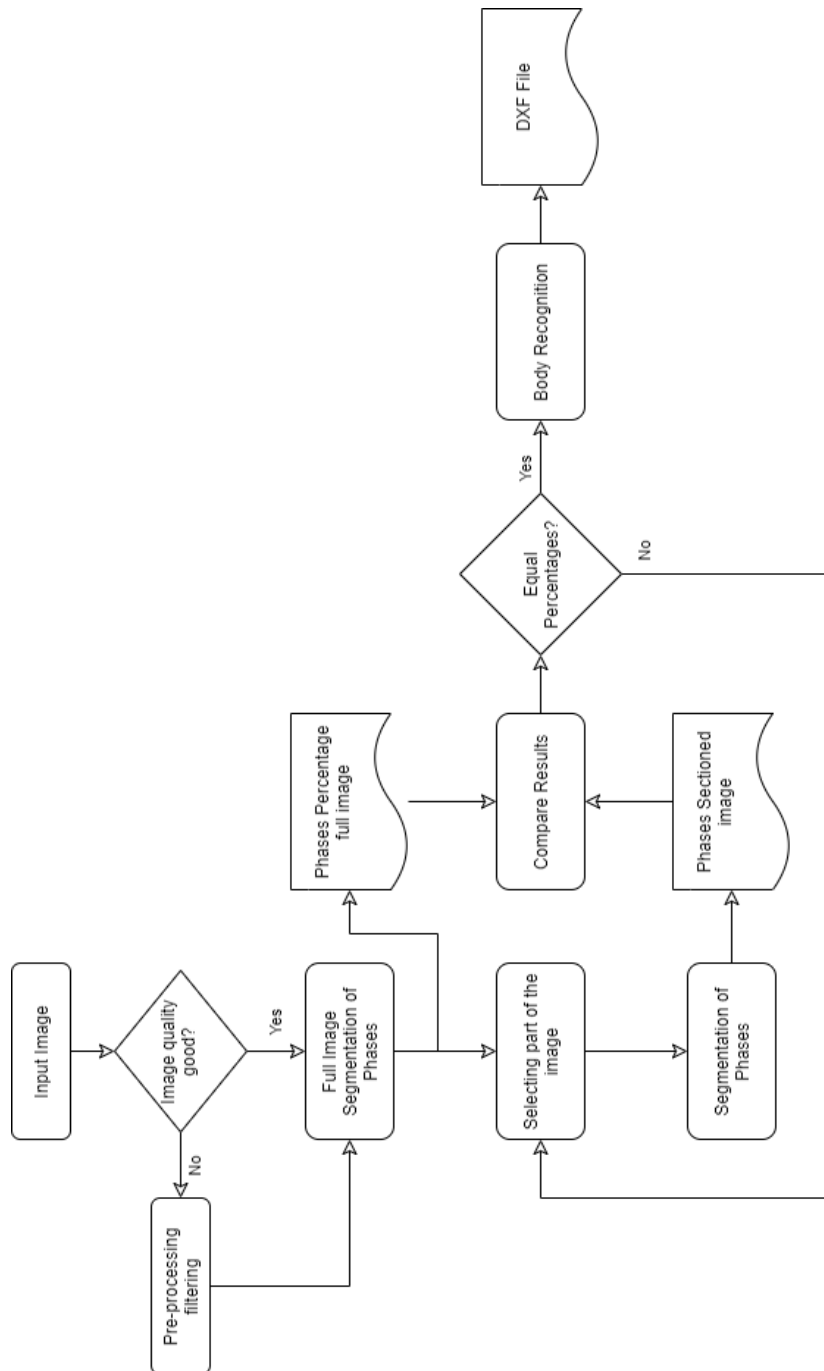


Figure 5.1: Schematics image segmentation process.

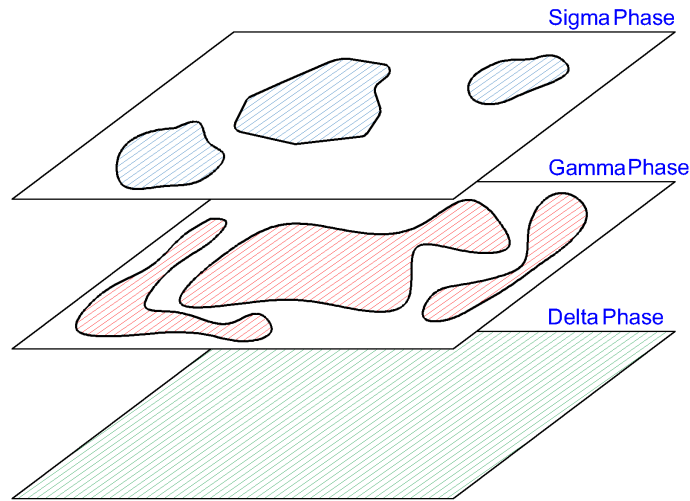


Figure 5.2: Layer scheme with the three phases.

5.1 Complete Image First Analysis

A microstructure contains several important information regardless of the material. One of the main characteristics that can be observed in the microstructure is the percentage of each phase. In order to calculate the phase percentage, an image segmentation script was created. This script is based mainly on the threshold segmentation technic. This technic, as it was described in section 3.1.1, is a simple way to separate the image into two groups, bodies and background. To take advantage of threshold technic simplicity and apply it to generate three different groups, a multilayer stacking was created. In this way, two phases correspond to two different layers and the background corresponds to the last phase, as illustrated in figure 5.2.

Two different threshold values are chosen manually to impose limits between one phase and the other

$$(\underbrace{\sigma}_{\text{darker}} - \text{phase}) < (\underbrace{\delta}_{\text{middle}} - \text{phase}) < (\underbrace{\gamma}_{\text{lighter}} - \text{phase}) \quad (5.1)$$

These threshold values are based on the values of the pixels in the image for a grayscale. An example of how the pixels are evaluated is shown in figure 5.3. The maximum value in the grayscale is 255 assigned to the white color, and the minimum is 0 assigned to black color. This values can be easily be inverted to cover images with reverse shades of gray.

Then, segmentation is done by evaluating each pixel value with respect to the previously imposed limits. Image 5.4 shows how pixels are assigned into three different groups.

160	150	150	140
30	20	60	140
80	20	80	200
100	100	160	205

Figure 5.3: Pixels values in a gray scale.

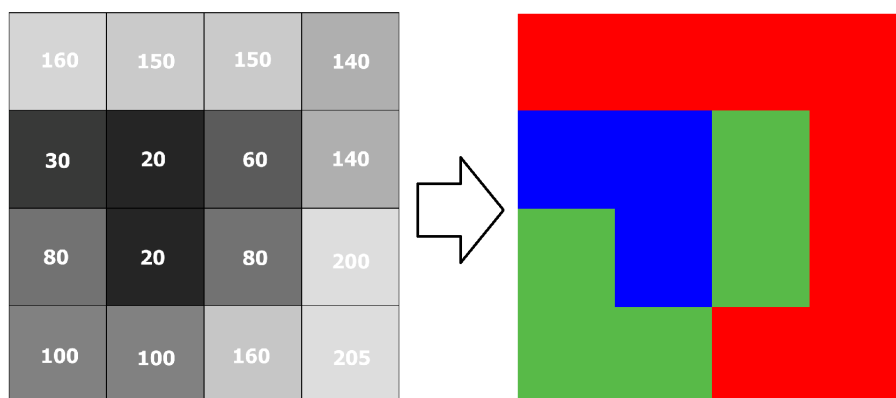


Figure 5.4: Pixels values assigned to a particular phase.

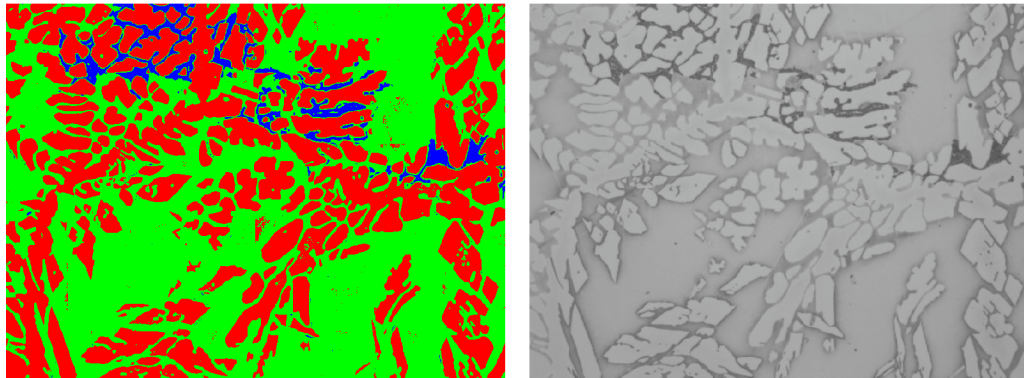


Figure 5.5: Image segmentation for section A - γ -phase(Red), δ -phase(Green), σ -phase(Blue).

This initial process is applied to several microstructure images as described in the particular case study of this thesis. The result of this segmentation for the first image is shown in figure 5.5.

5.2 Sectioned Image Analysis

In this section, the pre-processing tools used, Matlab script, and functions created to analyze the microstructure image are going to be briefly described. It is worth mentioning that the entire process described from now on for a section of the image was also performed for the complete image in order to obtain the values presented in the table 7.1.

5.2.1 Pre-processing

First, the size of the analyzed image is reduced by selecting a region with equal percentages of the phases as the original image. (See image 5.6)

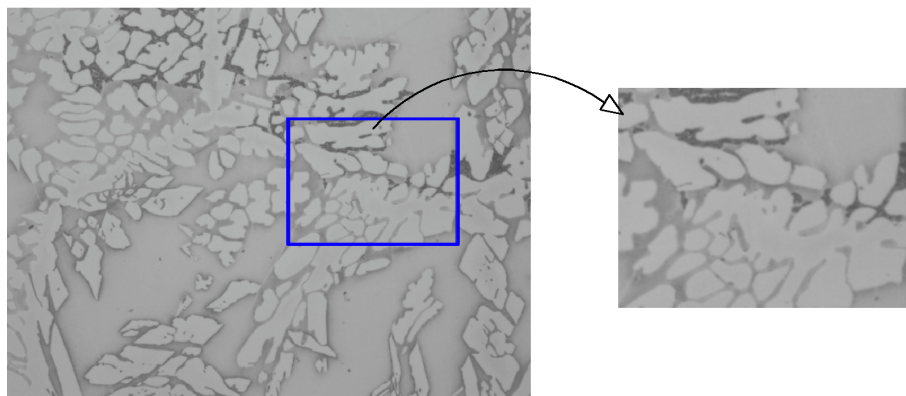


Figure 5.6: Pixels values assign to a particular phase.

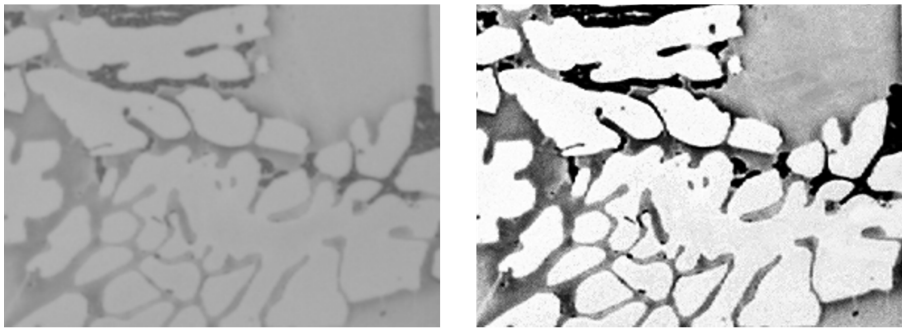


Figure 5.7: Pre-processing result.

A pre-processing procedure was applied to the image in the Krita software [3]. This is a free and open-source raster graphics editor with a wide range of tools and filters. The image was uploaded to the software and an enhancement sharpening filter was applied, this filter increases the contrast between the different shades of gray. If needed, manual adjustment can be performed through the histogram of the picture. The result of the pre-processing procedure is shown in figure 5.7

5.2.2 Phase Segmentation Function

The enhanced image is stored in a variable inside the main script of the program. To get rid of the unwanted channels present in the RGB image file, this image is converted to a grayscale format. This creates a matrix whose size equals the size of the image and containing all the values of the pixels. Still, in the main function, the threshold values are manually assigned.

The phase segmentation function is called from the main function passing as parameters of the grayscale matrix and the threshold limits. The elements of the matrix are compared with the threshold values and separated into three different groups. These groups can be interpreted as being a matrix containing only the elements, or pixels, that correspond to one particular phase, and the rest have a value equal to zero. Finally, a particular color is assigned to the non-null pixels that corresponds to the phase where they belong, namely, blue for σ -phase, red for γ -phase, and green for δ -phase, as illustrated in figure 5.8. When submitting the microstructure image to this process the result displayed in figure 5.9 is achieved.

The results of this process in the microstructure image are shown in figure 5.9.

Moreover, the outputs of the segmentation function are three matrices, one for each phase, appended to a list. These matrices are interpreted as layers that, stacked on top of each other, create the entire image with the segmented phases, as shown in figure 5.10.

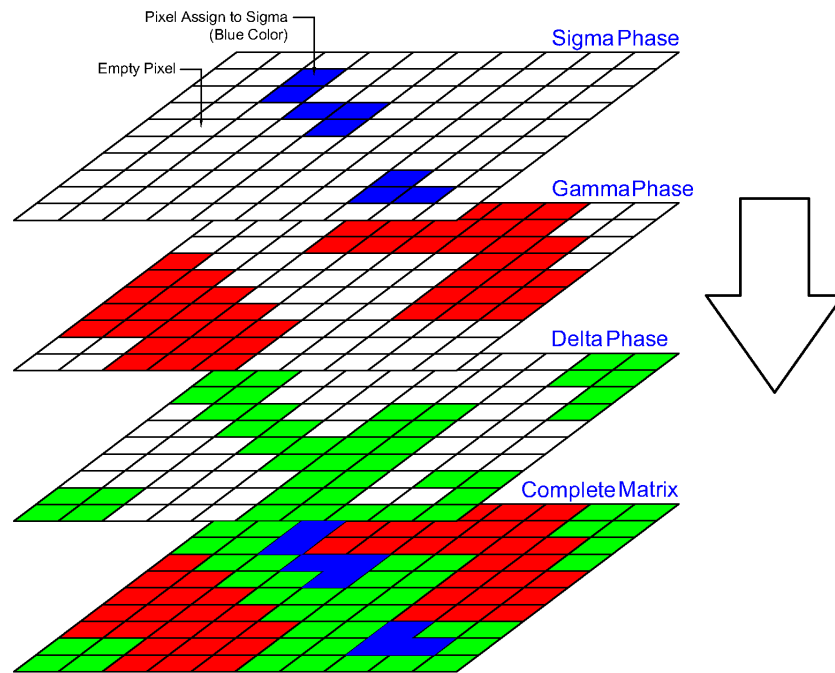


Figure 5.8: Schematic of the segmentation of the matrix containing the pixels.

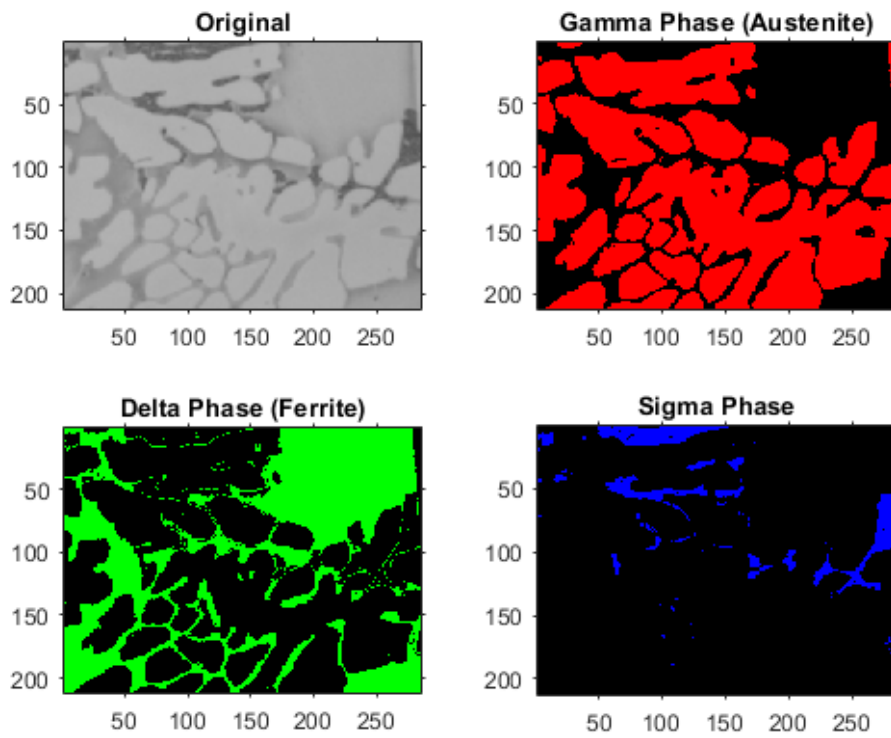


Figure 5.9: Phase color assignment for the microstructure image.

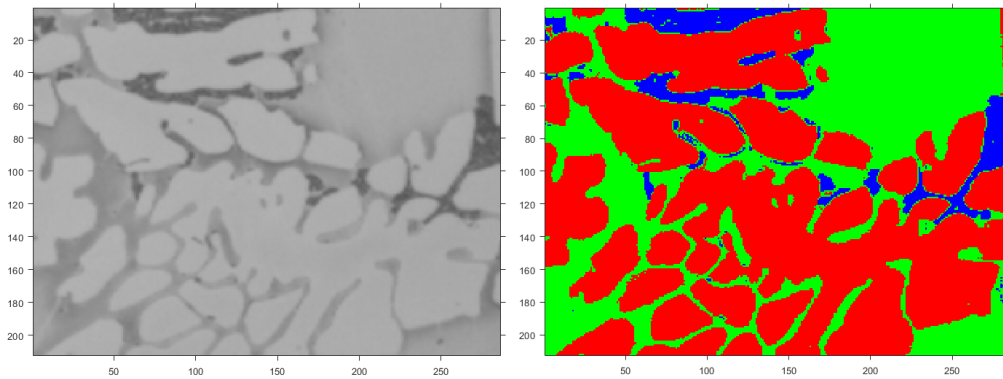


Figure 5.10: Segmented image with color assign to the different phases.

5.2.3 Body Recognition

Body recognition plays a fundamental rule within the whole microstructure analysis process. In fact, the main difference between the current work and the analyzed bibliography is the recognition of the bodies belonging to each of the phases and subsequent creation of a sketch file capable of being imported to a FEM software. First, the recognition of the bodies is done by a function that takes as parameters the list containing all matrices created in the previous section, and the phase of interest. For instance, let's focus on the γ -phase shown in figure 5.11, the layer correspondent to this particular phase is selected from the list of matrices.

Firstly, the layer is converted into a binary image with bodies in white and background in black. This approach, in which phase images are regarded as layers, allows a simplification in the shapes of the bodies by filling the holes inside of them. Then, the boundaries of these bodies are recognized by a Matlab build-in function that returns the coordinates of the pixels that belong to the body/background border as a 2x2 matrix for each body. Finally, the output for the body recognition function is a matrix with boundary points x and y coordinates, the number of objects

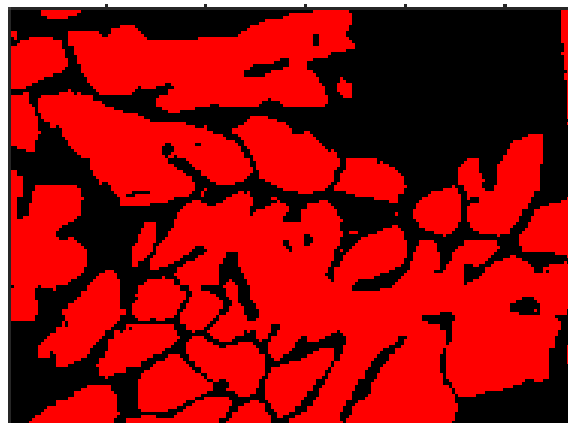


Figure 5.11: γ -phase layer.

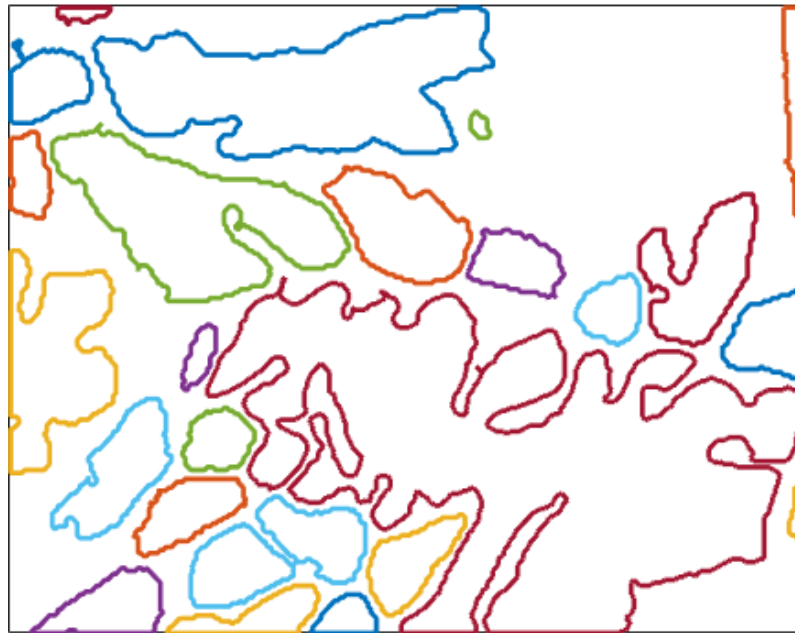


Figure 5.12: All bodies in γ -phase layer.

on the layer, and their area.

5.2.4 CAD Files

Now the CAD (Computer-Aided Design) file can be created from the coordinates of the points that compose the border of the body. This file is created by connecting all border pixels with a line using a function that takes as parameters the X and Y coordinates and has as output one file, saved in DXF format, for each object. This format allows a simple connection with Abaqus since this file format can be directly imported into the software. The construction of the bodies together is illustrated in image [5.12](#)

It is worth mentioning that because of the layer scheme used so far, the procedure for creating sketches is used only with the σ -phase and γ -phase. Due to the arrangement of the phases in the microstructure, it can be considered that the δ -phase is a background matrix where the two other phases are spread, as was illustrated in figure [5.2](#). Upon completing the procedures described in the chapter, the FEM simulation follows, as explained in the next chapter.

Chapter 6

Abaqus Simulation

After recognizing all the relevant features in the experimental microstructure images, and acquiring the DXF files, these can be used to construct the model that is going to be simulated in Abaqus software, in order to obtain the stresses values discussed in section 4.2.4.

6.1 Importing Sketch

The DXF files can be directly imported to Abaqus as sketches to be assembled. A critical issue was observed in this part of the study. Once the construction of the DXF files is made by connecting two consecutive points in the border with a line, the sketch on Abaqus is established by several lines leading to a high number of nodes/elements and very high computational cost. This becomes a problem when the mesh for these bodies is created. To overcome this issue, a spline is created connecting all the points by a single line, this process is illustrated in figure 6.1.

The number of elements and nodes in the sketch decreases and, additionally, the body shape is smoothed, and the control over the mesh is improved significantly as shown in figure 6.2, increasing simulation accuracy. Also with this modification, a smaller number of nodes can be created reducing the computational cost drastically.

6.2 Assemble and Material Assignment

Once the sketches are imported they are assembled creating one part to γ -phase and one part to σ -phase. For the δ -phase, a rectangle part with the microstructure image dimensions is drawn in the Abaqus part creation section. All these three parts are merged together to build the model displayed in figure 6.3

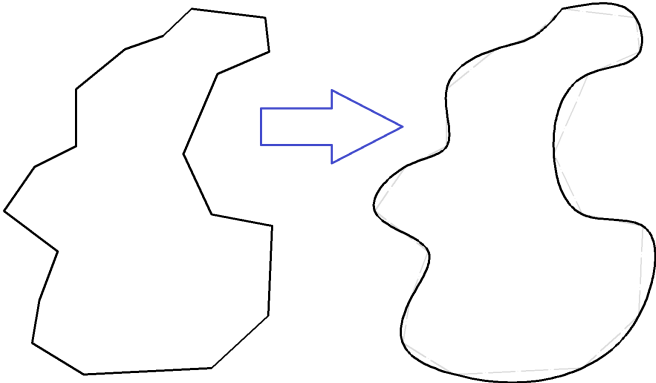


Figure 6.1: Spline sketch creation.

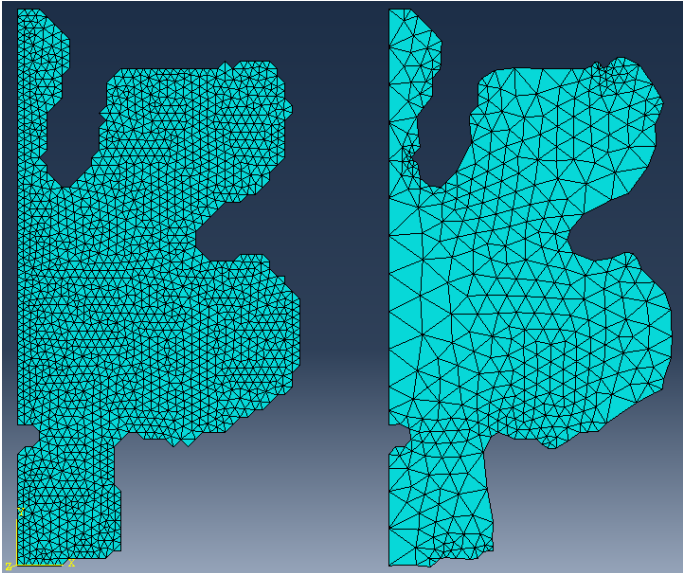


Figure 6.2: Difference in mesh before and after spline creation.



Figure 6.3: γ -phase, δ -phase and σ -phase merged in one model.

As it was mentioned before, each phase has different mechanical properties. The Young's Modulus and Poisson's Ratio are summarized in table 6.1

Table 6.1: Phase properties of different phases of DSS

Phase	Young's Modulus [GPa]	Poisson's Ratio
γ -phase	189	0.3
δ -phase	220	0.3
σ -phase	201	0.3

Three different materials were created on Abaqus and, each phase was assigned to them, as figure 6.4 shows.

6.3 Mesh

Once the material properties are assigned, the mesh of the microstructure can be created as shown in figure 6.5.

This mesh is constructed from triangular elements with a variable global size, depending on the microstructure analyzed. To determine the optimal size for the element, a node of interest was selected, aiming at a critical point in the simulation. The results for the main stress were obtained from this node. Then the global size of the element was reduced from the initial mesh until the values of stress in the selected node reaches convergency (figure 6.6).

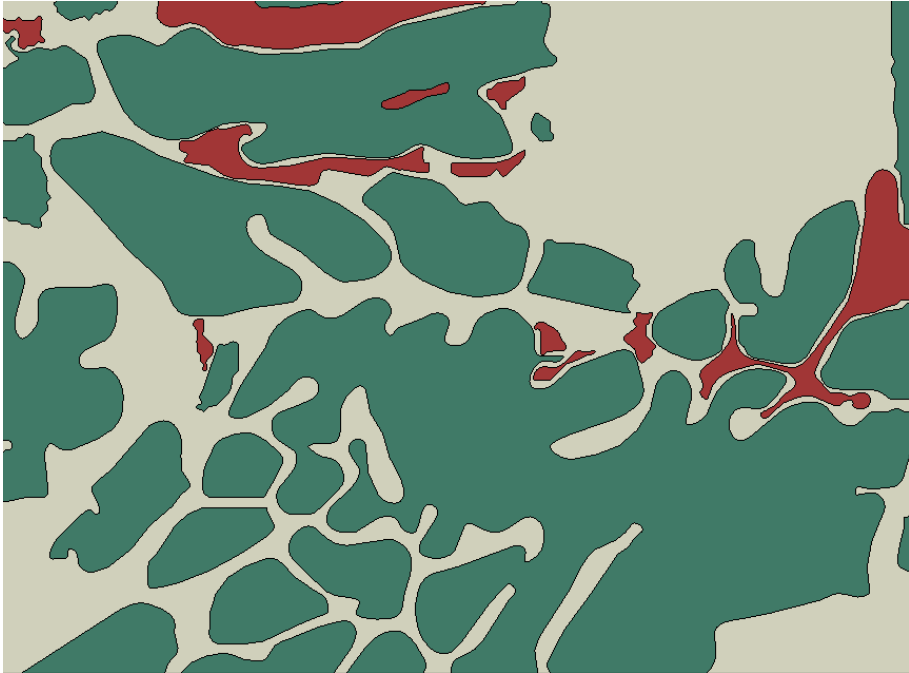


Figure 6.4: Material Assignment.

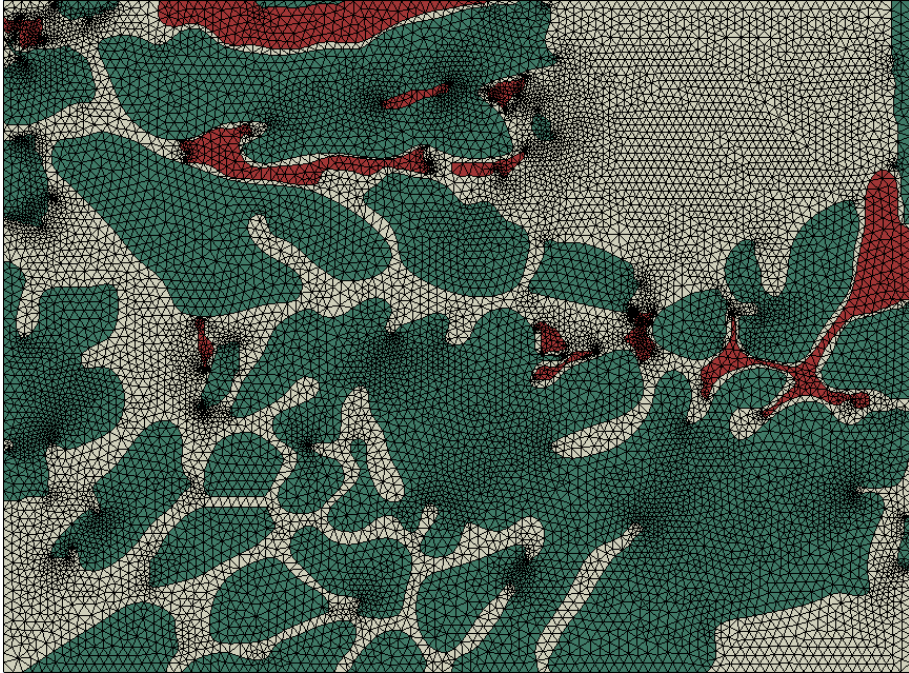


Figure 6.5: Mesh of the microstructure model.

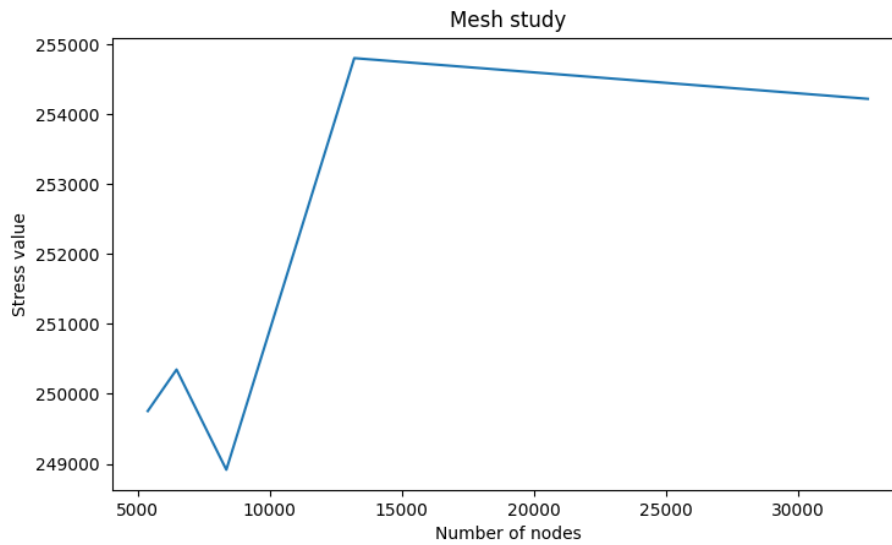


Figure 6.6: Mesh optimization graph.

Another important achievement of this study is that with the division between the phases well-delimited, elements of the mesh do not cross between phases, as illustrated in figure 6.7, without the need for any further action. In commercial softwares, to overcome this kind of problem, the shapes of the bodies are distorted so that the elements do not cross the boundaries between them.

6.4 Boundary Conditions and Load

Based on the theory presented in chapter 4, three different simulations are required to construct the elasticity matrix for the 2D model. One for the x direction tension, one for the y direction tension, and one for the shear loading. The last one has to be divided into two different simulations to capture all the needed values. In this way, four different setups were created, with different loads and boundary conditions, as illustrated in figure 6.8

Following the theory presented in section 4.2.5, the loads for the four simulations described vary according to the size of the image under analysis. Making the total displacement equal to the elongation of the part under study, the deformation becomes unitary for the main direction of the simulation as shown on 4.56 system of equations, this allows that with the calculations presented in the same section, it is possible to obtain the elasticity matrix D . More details about the assembling of the elasticity matrix are going to be addressed in the next sections.

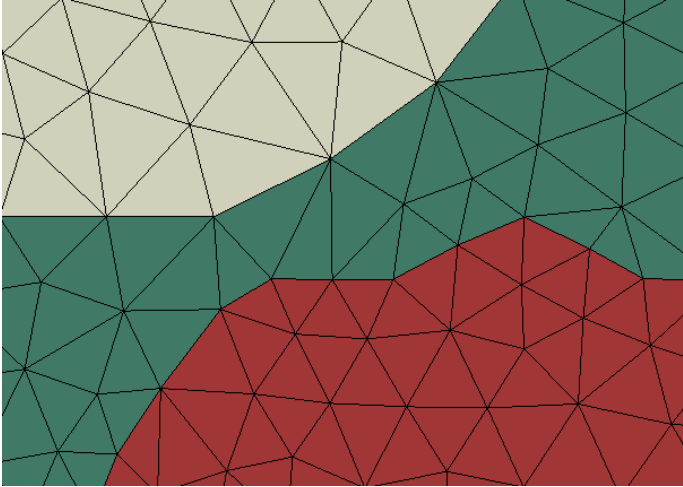


Figure 6.7: Detail on mesh through different phases.

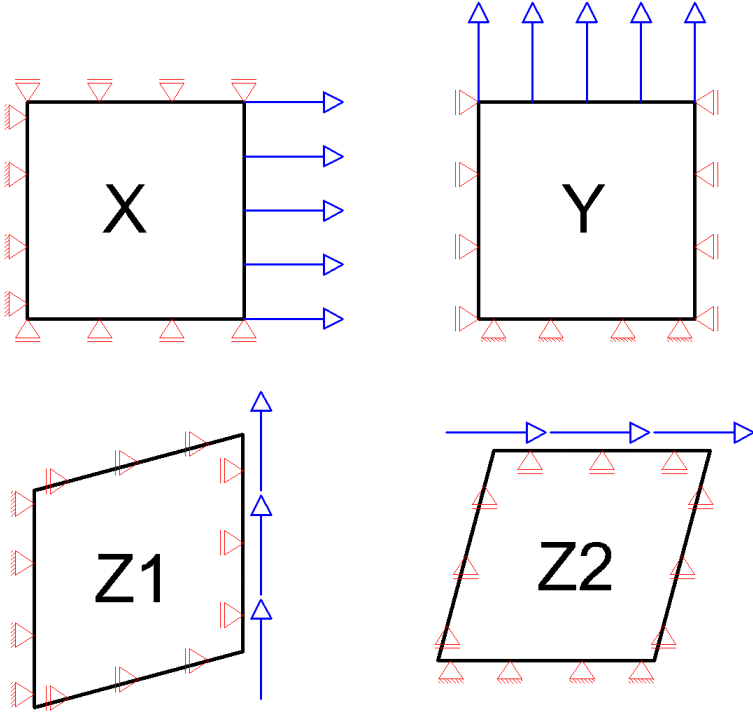


Figure 6.8: Load and Boundary Conditions.

6.5 Post-processing Python Script

The four simulations described in the previous sections result in four different ODB format files created by Abaqus, containing all the information regardless of the simulation. A python script is developed aiming to extract the needed information from the ODB file. This script extracts the initial and final coordinates to all nodes inside the mesh. As also extract the stress in every direction.

This information is store in a CSV (Comma-Separate Values). Where the first column is the number of the nodes, the second, third, and fourth columns store the S11 (x-direction stress), S22 (y-direction stress), and S12 (shear stress) respectively. And the last four columns hold the x and y initial and final coordinates of the nodes.

6.6 Elasticity Matrix D

Again, the Matlab software is used to create a post-processing program that read the results store in the CSV file and the theory from chapter 4 is applied to construct the elasticity matrix D. This task was divided into different functions improving the computational time and allowing a generalization of the program for future use with different types of elements and shape functions.

6.6.1 Symbolic Jacobian Matrix

To make it easier for future use of different shape functions, the Jacobian matrix is calculated in a symbolic way in the first step and later the values for the variables are substituted. In the main script, the results of the four simulations stored in the CSV file are converted into an array of numbers. After that, the shape functions are declared. The shape functions and the type of the element are the parameters of the functions that construct the symbolic Jacobian matrix, this function is based on the theory presented in the section 4.2.3. So far the only element used is the triangular 3-node element. The Jacobian build function performs the differentiation of the shape functions with respect to ξ and η and then, depending on the element, build the symbolic jacobian matrix, its symbolic determinant, and the shape functions with the respective numerical values of ξ and η already substituted.

6.6.2 Jacobian Matrix Numerical Determinant

The program uses another function to substitute the numerical values of the initial and final coordinates obtained from the CSV file for each node, into the variables of the symbolic equation

created from the determinant of the Jacobian matrix. This function returns an array containing the numerical values of the determinant for each element present in the mesh.

6.6.3 Assembling Elasticity Matrix D

The construction of the elasticity matrix D is also done in a separate function and also depends on the type of element used. The values of stress, obtained from the CSV file, and the values of the determinant of the Jacobian matrix, calculated in the previous function, are used in the formulation introduced in equation 4.60 to calculate each element in the D matrix. Finally, the 3x3 elasticity matrix is stored in a text file.

A schematic diagram of the process described in chapter 6 is illustrated in figure 6.9. The process consists of, at first, importing the DXF file created, described in chapter 5, as a sketch to Abaqus software. This file is used to carry the simulation based on the boundary conditions and loads described in section 6.4. After that, Abaqus provides the user a ODB file containing all the results of the simulation, this file is processed by a python script that extract only the necessary results. These results are then used in a Matlab script that creates the Jacobian matrix, evaluate the node results, and assemble the elasticity matrix based on the equations 4.60. Next chapter illustrates the whole methodology by applying it to a case study.

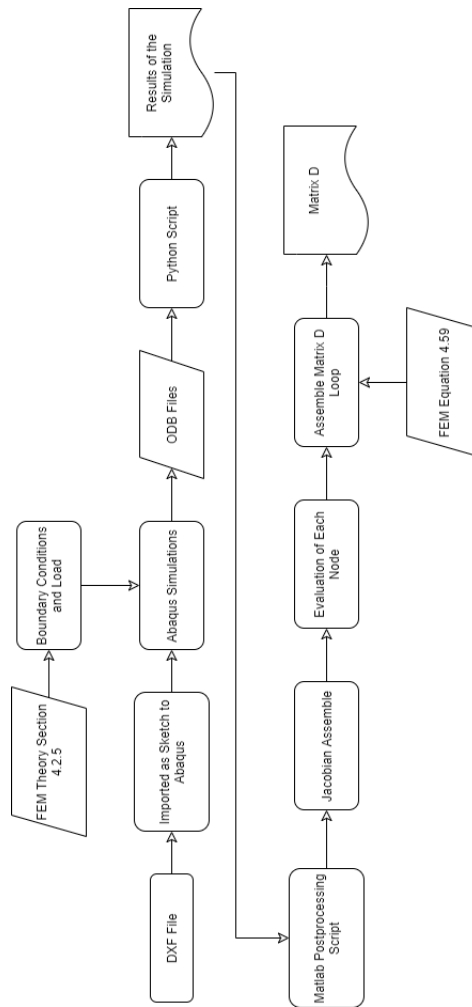


Figure 6.9: Schematics Abaqus and postprocessing process

Chapter 7

Case Study

Real images are taken from a C-shape part made of Super Duplex Stainless Steel (SDSS). These images are obtained from two different techniques optical microscopy images and scanning electron microscope (SEM) images, which are used to illustrate the developed methodology.

7.1 Problem Description

The specimen used in this case study is a sand casting Super Duplex Stainless Steel poured in a C-shape model. The C-ring specimen, shown in figure 7.1, due to its geometry exhibits different cooling rates along its shape causing a non-uniform phase distribution.

Different microstructures along the sample were captured by dividing the C-ring into 6 parts named from A to F. Differences in the areas of these sections cause a reduction in the cooling rates from the smaller (A) to the bigger one (F). A sample from each section was analyzed on optical microscopy. The microstructure of each one is displayed in figure 7.2, where can be observed the three phases present in the material. In the microscopy images the lightest shade of gray corresponds to the γ -phase (Austenite), the middle shade to the δ -phase (Ferrite), and the darkest to sigma precipitation. These images can be observed in figure 7.2. In the images obtained with scanning electron microscope this shades are inverted as it is shown in figure 7.3. The almost white color corresponds to the χ -phase precipitations, that was not observed in the microscopy images, and the lightest shade of gray corresponds to σ -phase precipitation. The middle tones that are suppose to be between the γ and δ phases are not so well contrasted in the SEM images. So for these images only the precipitations of the two different χ -phase and σ -phase was taking into consideration. Because of this, it was only possible for the SEM images to be analyzed by the segmentation process of the program and the percentage of the precipitations was taking as a result.

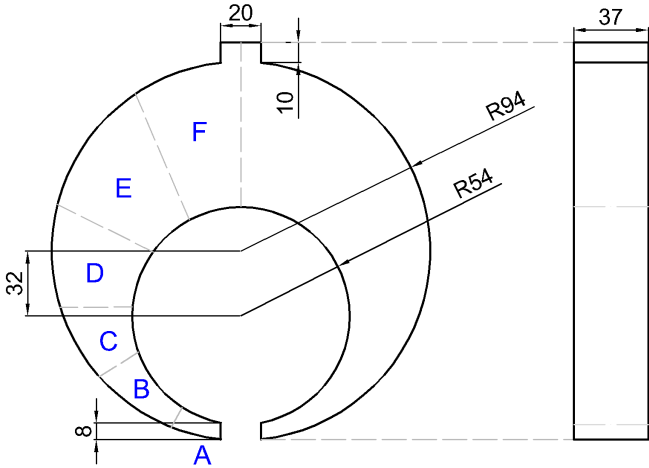


Figure 7.1: C-ring specimen (dimensions in [mm]).

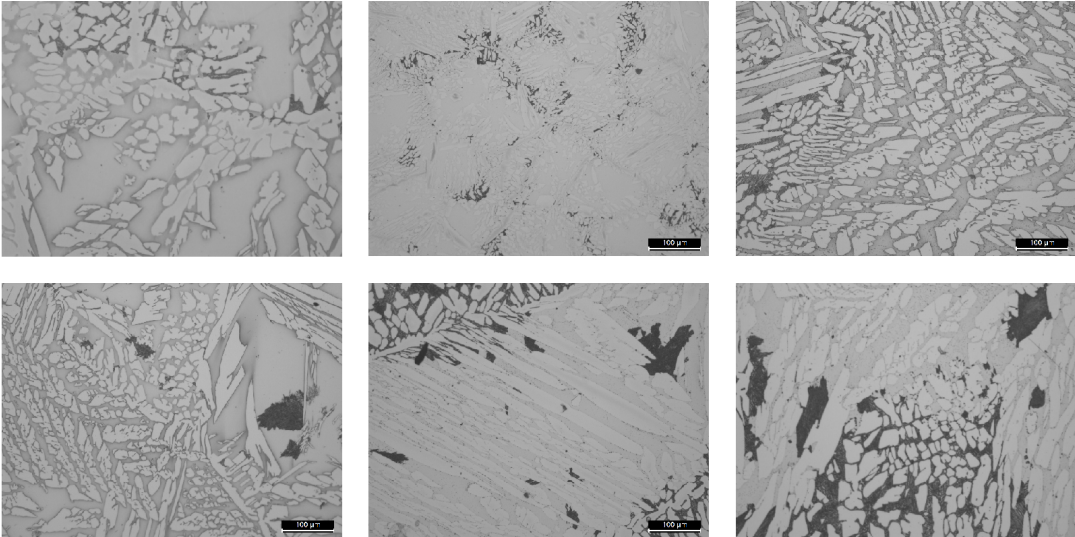


Figure 7.2: Microstructure of the 6 parts obtained from the C-ring - microscopy images.

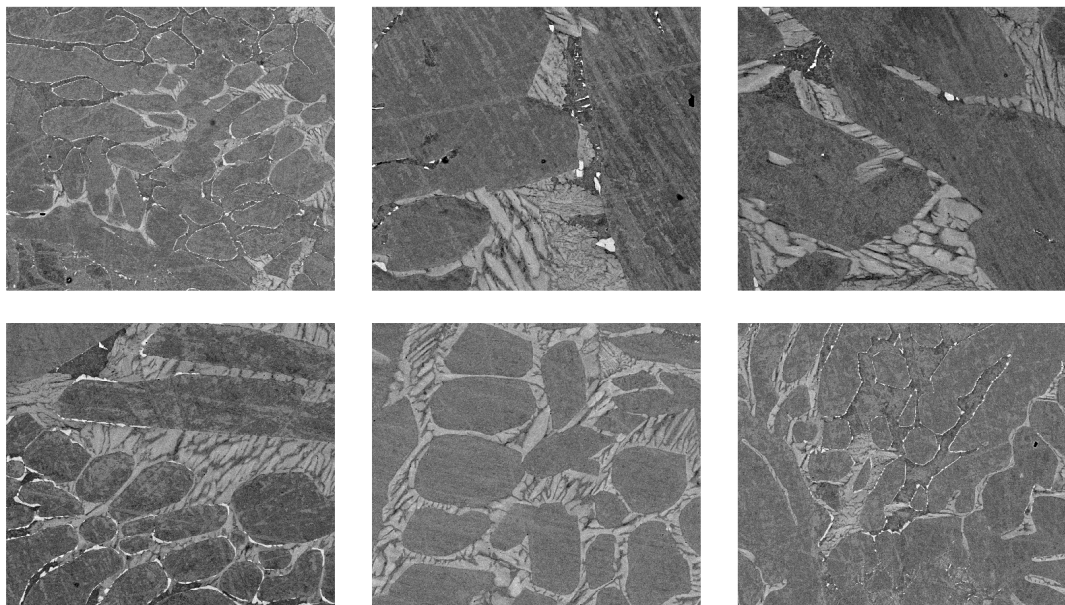


Figure 7.3: Microstructure of the 6 parts obtained from the C-ring - SEM.

For the microscopy images, 6 different microstructures, one for each section of the C-ring, were analyzed and the respective percentage of each phase are displayed in table 7.1.

Table 7.1: Percentage of phases on sections A to F

	Section A	Section B	Section C	Section D	Section E	Section F
γ	49.46%	48.32%	55.34%	44.20%	62.92%	55.34%
δ	2.06%	3.13%	7.76%	2.98%	12.97%	19.45%
σ	48.48%	48.55%	36.90%	52.82%	52.82%	25.21%

For the SEM images various microstructures were obtained for each section, an average percentage of each image made and the results are presented in the table 7.2

Table 7.2: Percentage of precipitates on sections A to F

	Section A	Section B	Section C	Section D	Section E	Section F
σ	10.89 %	11.97 %	20.19 %	13.94 %	16.32 %	21.76 %
χ	0.69 %	0.36 %	0.16 %	0.95 %	0.47 %	0.007 %

For the six images made from microscopy, the next steps of the studies were applied. After the segmentation of the phases, the recognition of the bodies, and the creation of the CAD files, the simulation made by the finite element software Abaqus had very consistent results with the expected for this material. Some of the results are displayed in figures 7.4 to 7.7

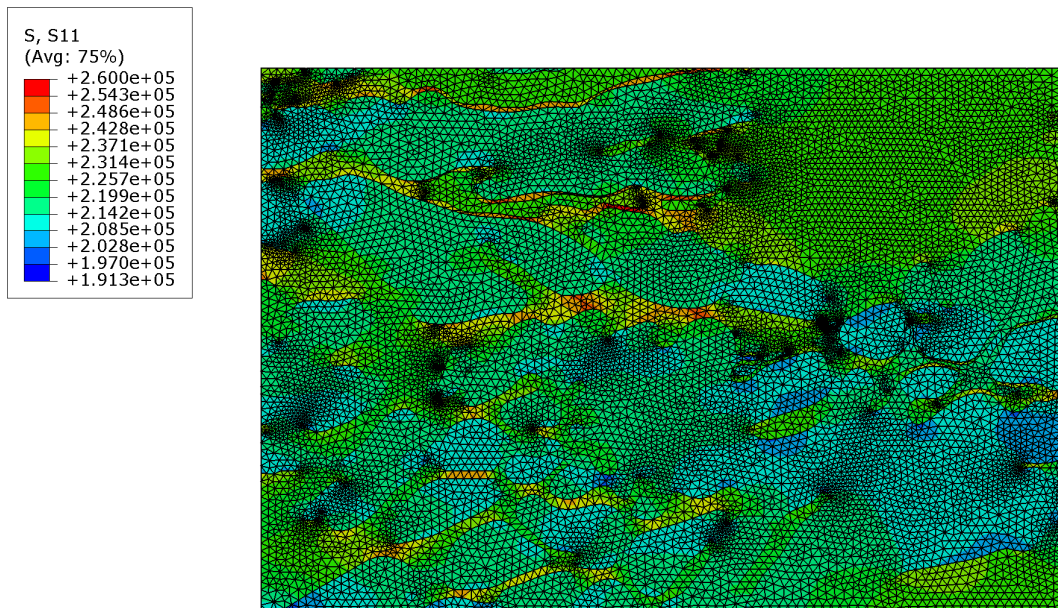


Figure 7.4: Results for the simulation in X direction.

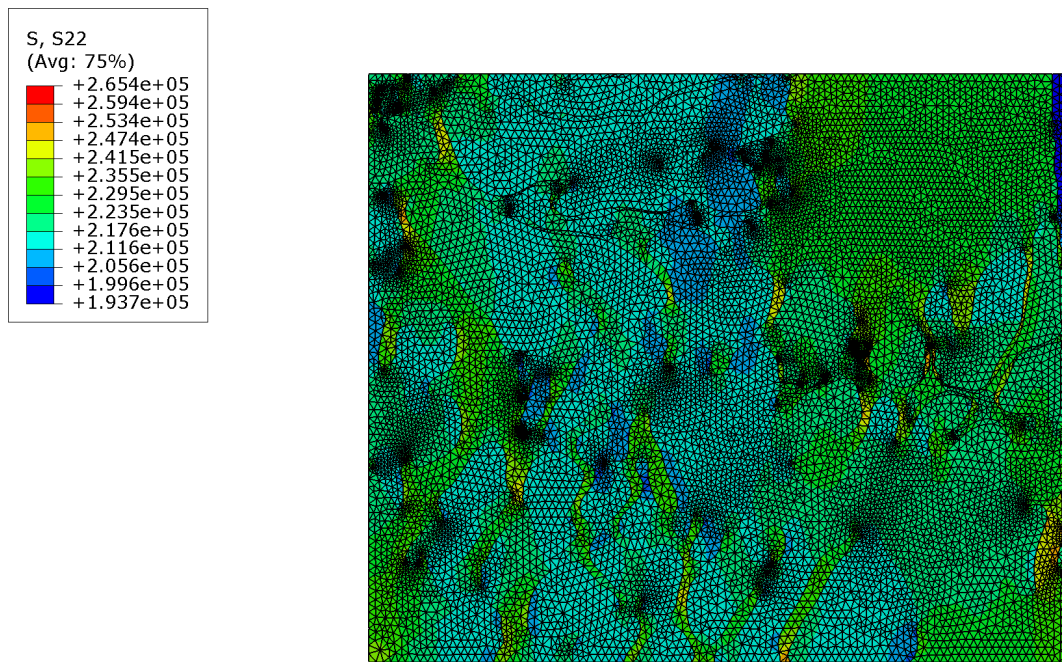


Figure 7.5: Results for the simulation in Y direction.

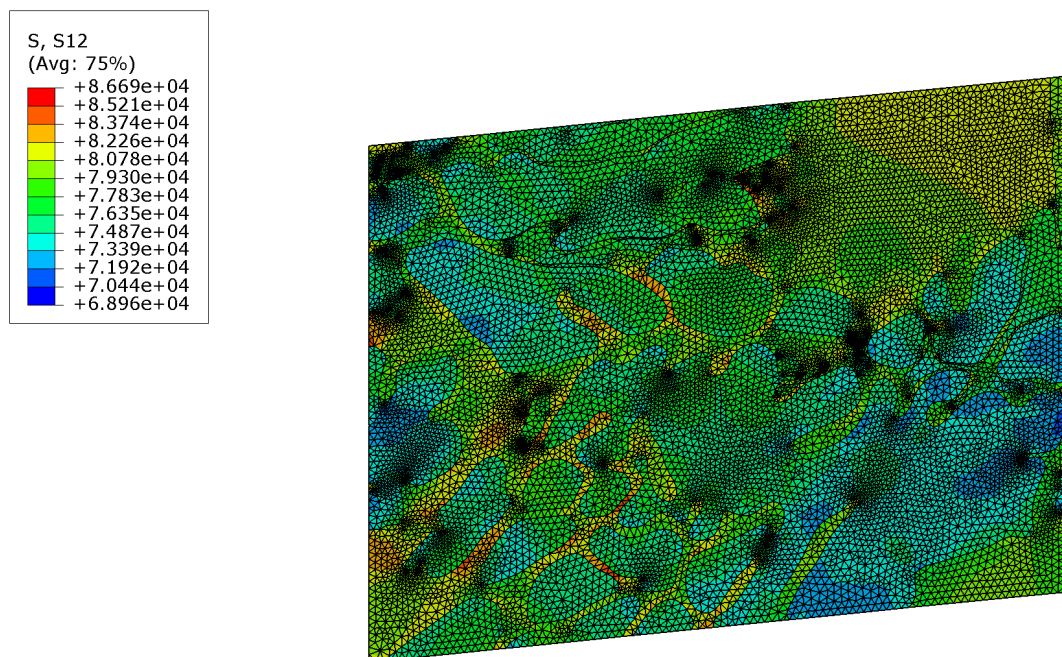


Figure 7.6: Results for the simulation in shear.

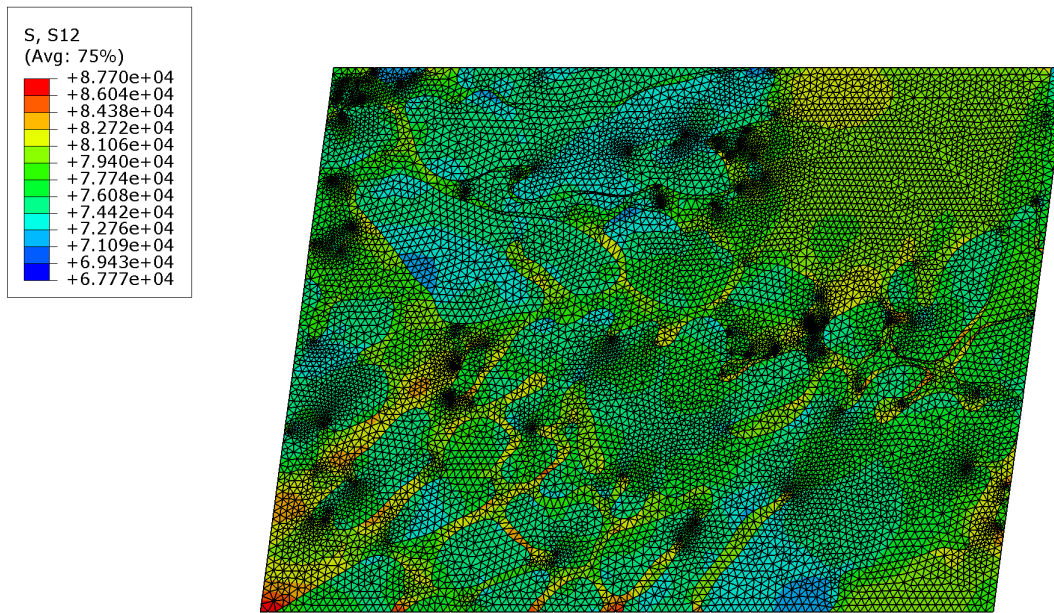


Figure 7.7: Results for the simulation in shear.

The elasticity matrices obtained for each section of the C-ring specimen are shown in the table 7.3.

Table 7.3: Elasticity Matrix for all sections on C-ring specimen (*GPa*)

Section A	Section B	Section C
$\begin{vmatrix} 226.39 & 67.96 & 0.86 \\ 67.84 & 226.61 & 0.57 \\ 1.58 & 0.72 & 79.31 \end{vmatrix}$	$\begin{vmatrix} 225.52 & 67.59 & 0.24 \\ 67.58 & 225.39 & 0.12 \\ 0.36 & -0.16 & 79.00 \end{vmatrix}$	$\begin{vmatrix} 222.31 & 66.68 & 0.24 \\ 66.67 & 222.57 & 0.02 \\ 0.38 & -0.04 & 77.81 \end{vmatrix}$
Section D	Section E	Section F
$\begin{vmatrix} 224.25 & 67.35 & -0.19 \\ 67.35 & 224.90 & -0.21 \\ -0.21 & -0.27 & 78.57 \end{vmatrix}$	$\begin{vmatrix} 218.30 & 65.56 & -0.21 \\ 65.54 & 218.30 & -0.18 \\ -0.28 & -0.30 & 76.46 \end{vmatrix}$	$\begin{vmatrix} 220.15 & 66.11 & 0.14 \\ 66.11 & 220.59 & 0.01 \\ 0.23 & 0.05 & 77.13 \end{vmatrix}$

From the results regarding the percentage of each phase compared with the elasticity matrix obtained for the six sections of the C-ring, can be observed that with the increasing of the σ -phase precipitation the δ -phase suffers a reduction in their percentage while, in turn, the γ -phase does not undergo significant changes. This change results in a reduction of the values of the diagonal line of the elasticity matrix. Knowing the Poisson's ratio for the material it is possible to estimate the values for the Young's modulus using the equation presented in 4.3, the results can be seen in

Table 7.4: Young's modulus (E) for each section of the C-ring (MPa)

	Section A	Section B	Section C	Section D	Section E	Section F
E	206014.9	205223.2	202302.1	204067.5	198653	200336.5

the table [7.4](#)

These values are consistent with the expected results, where they lie between the limits of a material that is composed only of the ferrite phase (δ) and a material composed only of the austenite phase (γ) in its microstructure according to the values presented in the table [6.1](#). Higher values obtained for Young's modulus represent a more fragile material, but this is not the single factor. It was widely discussed in the literature review section that the increase in precipitates in the material's microstructure reduces its resistance to deformation. As only elastic properties were analyzed in the present study, it was not possible to conclude this statement yet.

Chapter 8

Conclusion

8.1 Results and Discussions

The image segmentation code developed in Matlab was able to separate the three different phases that compose the microstructure of a DSS, as well as, calculate the percentage of each of the phases. The main difference between the developed program and existing softwares is the ability to create a 2D CAD file ready to be used in a commercial finite element software. In addition, the simplicity presented by the code, allows it to be modified to meet different uses and needs. From small modifications, the current script can extract different features from the images. Some limitations in the program could be observed. The quality of the image directly affects the results and the need to pre-process it with filters and other techniques.

For future work, automation for the selection of threshold values can be explored further. This process would reduce the time needed for analysis and the need for more detailed pre-processing for the image. The limitations regardless of the quality of the image can be overcome by the use of different image processing tools, as the morphological operator's dilatation and erosion, to clear the images from undesirable noises. Different image segmentation techniques, like watershed algorithms, can be also used for a more automatic process in the image analysis. It is worth mentioning that with the continuous development of image recognition by Artificial Intelligence technologies, this process as a whole can be automated, increasing the speed with which images are processed and the accuracy of results.

8.2 General Observations

The results obtained from this work allow the following conclusions.

1. A simple threshold algorithm used based on the pixel values of the grayscale image of a microstructure was able to accurately segment the existent phases γ , δ , and σ . The use of this algorithm can be explored to detect other types of features in the microstructure with small adjustments to the threshold values. This script can be also modified with more sophisticated technics to overcome problems with automation of the process and the need for pre-processing to improve the image.
2. The creation of a CAD file for future analysis in specific Finite Element programs greatly facilitates the structural analysis of the material. It allows the creation of a better mesh and better simulations. Future works can go deeper into the development of better tools for creating sketches, bringing them together before importing them into the FEM program and using splines for their construction, avoiding the need for future conversions.
3. The use of Abaqus as a finite element analysis program allowed easy integration with a script developed in Python to extract the results of the simulations.
4. This work was successful in creating a fundamental unity of multiscale analysis. The whole methodology integrates identification of microstructural features and respective homogenization to derive macroscopic properties.
5. The aforementioned multiscale unit is essential in the study of DSS as the precipitation of secondary phases may severely deteriorate their mechanical properties.

Appendix A

Matlab Segmentation Code

A.1 Main Program

```
1 %%
2 clc; clear; close all;
3
4 %% Read the microstructure image
5 I = imread('microstructureA_01.png');
6
7 % Converting to gray scale
8 I = rgb2gray(I);
9
10 [nn, mm] = size(I);
11
12 % Threshold values
13 limSigma = 80;
14 limDelta = 230;
15
16 [RGB_Phases, RGB_Tol, RGB_Delta, RGB_Gamma, RGB_Sigma] = phaseSeparation(Image,
    limSigma, limDelta);
17
18 %% Bodies recognition
19 phase = 'gamma';
20
21 [B, n, A] = meshBody(RGB_Phases, phase);
22
23 k = [];
24 Atot = 0;
25
26 % Percentage of Phase
27 per = (Atot/(nn*mm)) * 100;
```

```

28 disp('\n Percentage of phase: ');
29 disp(per);
30
31 % Noise reduction
32 for i = 1:length(A)
33     if A(1,i) > 10
34         k(end+1) = i;
35         Atot = Atot + A(1,i);
36     end
37 end
38
39 numBodies = length(k);
40
41 for n=1:numBodies
42     b = B{k(n)};
43     plot(b(:,2), b(:,1), 'LineWidth', 2), hold on, grid off
44     set(gca, 'YDir', 'reverse')
45     set(gca, 'XTick', [], 'YTick', [])
46     xlim([0 mm])
47     ylim([0 nn])
48 end
49
50 for n = 1:numBodies
51     fileName = sprintf('%s_body_%d', phase, n);
52     fullFileName = fullfile('dxFiles\'', fileName);
53     writeDXF(fullFileName, B{k(n)}(:,1), B{k(n)}(:,2));
54 end

```

A.2 Phase Segmentation Function

```

1 function [X_RGB, X_RGB_Tol] = phaseSeparation(X_, limSigma, limDelta)
2
3     % Compare threshold values
4     X_Sigma = X_ >= limSigma;
5     X_Delta = X_ < limSigma | X_ >= limDelta;
6     X_Gamma = X_ < limDelta;
7
8     % Call function to assign color
9     RGB_Gamma = colorAssign(X_Gamma, 'r');
10    RGB_Delta = colorAssign(X_Delta, 'g');
11    RGB_Sigma = colorAssign(X_Sigma, 'b');
12
13    % List with results
14    X_RGB = {RGB_Sigma RGB_Gamma RGB_Delta};
15    X_RGB_Tol = RGB_Sigma + RGB_Gamma + RGB_Delta;

```

```
16
17 figure ()
18 subplot (1,2,1)
19 imshow(X_RGB_Tol), hold on
20 grid on
21 subplot (1,2,2)
22 imshow(X_)
23
24 end
```

A.3 Color Assignment Function

```
1 function [RGB_Image] = colorAssign(Xdata, color)
2
3   red = 0;
4   green = 0;
5   blue = 0;
6
7   if color == 'r'
8       red = 255;
9   elseif color == 'g'
10      green = 255;
11  elseif color == 'b'
12      blue = 255;
13  end
14
15  [fil, col] = size(Xdata);
16  RGB_Image = zeros(fil,col, 3);
17  [posX , posY] = find(Xdata==0);
18  numIter = size(posX,1)*size(posX,2);
19
20  for ii = 1 : numIter
21      RGB_Image(posX(ii),posY(ii), 1) = red;
22      RGB_Image(posX(ii),posY(ii), 2) = green;
23      RGB_Image(posX(ii),posY(ii), 3) = blue;
24  end
25
26 end
```

A.4 Body Detection Function

```
1 function [B, n, Areas] = meshBody( RGB_Phases, phase)
2
3     if phase == 'sigma'
4         index = 1;
5     elseif phase == 'gamma'
6         index = 2;
7     elseif phase == 'delta'
8         index = 3;
9     else
10        disp('There is no phase with this name')
11    end
12
13    I = RGB_Phases{index};
14
15    figure()
16    subplot(2, 2, 1)
17    imshow(I)
18
19    BW = im2bw(I);
20    subplot(2, 2, 2)
21    imshow(BW)
22
23    subplot(2, 2, 3)
24    imshow(BW)
25
26    BW = imfill(BW, 'holes');
27
28    subplot(2, 2, 4)
29    imshow(BW);
30
31    measurements = regionprops(BW, 'Area');
32    Areas = [measurements.Area];
33
34    [B,L,n,A] = bwboundaries(BW);
35
36    fprintf('Number of objects: %f', n)
37 end
```

A.5 Create DXF file Function

```
1 function [] = writeDXF(name, x, y)
2
3     fileName = sprintf('%s.dxf', name);
4
```

```
5 file = fopen(fileName, 'w');
6
7 fprintf(file, '0\n SECTION\n 2\n ENTITIES\n 0\n');
8 for i = 1:(length(x) - 1)
9
10     % Create a new line element in the file
11     fprintf(file, 'LINE\n 8\n 0\n');
12
13     % First coordinate
14     fprintf(file, '10\n %.10f\n 20\n %.10f\n', x(i), y(i));
15
16     % Second coordinate
17     fprintf(file, '11\n %.10f\n 21\n %.10f\n', x(i+1), y(i+1));
18
19     fprintf(file, '0\n');
20 end
21
22 fprintf(file, 'ENDSEC\n 0\n EOF\n');
23 fclose(file);
24 end
```

Appendix B

Python Script

B.1 Extract Results Python Script

```
1 from odbAccess import *
2 import odbAccess
3 import os, glob
4
5 ## Get the values in the nodes
6 def nodeValues(odbFile, nodeNumber, instanceName, outFile):
7
8     # Geometry assembly
9     assembly = odbFile.rootAssembly
10
11     # Grab node
12     node = assembly.instances[instanceName].getNodeFromLabel(nodeNumber)
13
14     # Stress Field
15     stressField = odbFile.steps.values()[-1].frames[-1].fieldOutputs['S']
16
17     # Stress in the evaluated node
18     stressNode = stressField.getSubset(region=node, position=ELEMENT_NODAL)
19
20     str11 = stressNode.values[0].data[0]
21     str22 = stressNode.values[0].data[1]
22     str12 = stressNode.values[0].data[3]
23
24     outFile.write('\n %5s, %15s, %15s, %15s,' %
25                 (nodeNumber, str11, str22, str12,))
26
27     xyFinal = odbFile.steps.values()[-1].frames[-1].fieldOutputs['COORD']
28     xyInit = odbFile.steps.values()[-1].frames[0].fieldOutputs['COORD']
```

```

29
30     xyNodeFinal = xyFinal.getSubset(region=node)
31     xyNodeInit  = xyInit.getSubset(region=node)
32
33     xNodeFinal = xyNodeFinal.values[0].data[0]
34     yNodeFinal = xyNodeFinal.values[0].data[1]
35     xNodeInit  = xyNodeInit.values[0].data[0]
36     yNodeInit  = xyNodeInit.values[0].data[1]
37
38
39     outFile.write('%20s, %20s, ' % (xNodeInit, yNodeInit))
40     outFile.write('%20s, %20s' % (xNodeFinal, yNodeFinal))
41
42
43     return ([str11, str22, str12])
44
45
46 #####
47
48 instanceName = 'ALLPHASES-1'
49
50 filePath = os.getcwd()
51 OdbFiles = glob.glob('*.*odb')
52
53 for file in OdbFiles:
54
55     # Name each .odb file
56     fileName = file
57
58     # Assign each odb file
59     odbFilePath = filePath + '/' + fileName
60     odbFile = odbAccess.openOdb(path=odbFilePath, readOnly=True)
61
62     # Create a outFile for each .odb
63     nameOutFile = 'Results' + fileName[0:-4] + '.csv'
64     outFile = open(nameOutFile, 'w')
65
66     outFile.write('\n %5s, %15s, %15s, %15s, %20s, %20s, %20s, %20s' %
67                 ('Node', 'S11', 'S22', 'S12', 'X_init', 'Y_init', 'X_final', '
68                 Y_final'))
69
70     assembly = odbFile.rootAssembly
71     nodes = assembly.instances[instanceName].nodes
72     numNodes = len(nodes)
73
74     for n in range(numNodes):
75
76         ## Take each node number

```

```
77     nodeNumber = nodes[n].label
78
79     stress = nodeValues(odbFile, nodeNumber, instanceName, outFile)
80
81     odbFile.close()
82 exit()
```

Appendix C

Matlab Post-processing Codes

C.1 Main Program

```
1 %%
2 clc; clear; close all;
3
4 %%
5 X = readtable('ResultsX.csv', 'HeaderLines',2);
6 Y = readtable('ResultsY.csv', 'HeaderLines',2);
7 Z1 = readtable('ResultsZ1.csv', 'HeaderLines',2);
8 Z2 = readtable('ResultsZ2.csv', 'HeaderLines',2);
9
10 X = table2array(X);
11 Y = table2array(Y);
12 Z1 = table2array(Z1);
13 Z2 = table2array(Z2);
14
15 %% Shape Functions
16 syms xi eta
17
18 N = [1-xi-eta, xi, eta];
19
20 %%% Jacobian Build %%%
21
22 % 3 Nodes Triangular Element = triangle3nodes
23 [detJ, dN, N, w] = jacobian_build('triangle3nodes', N);
24
25 %% Jacobian Matrix and D assemble
26
27 S11 = 2;
28 S22 = 3;
```

```

29 S12 = 4;
30 xIni = 5;
31 yIni = 6;
32
33 [detJ, node1, node2, node3, numElem] = node_jac_matrix(X, detJ, xIni, yIni);
34
35 D = assemble_D (N, node1, node2, node3, X, Y, Z1, Z2, detJ, numElem, ...
36               S11, S22, S12, w);
37
38 %%%% Write txt file
39 writematrix(D, 'ElasticityMatrix.txt', 'Delimiter', 'tab');

```

C.2 Symbolic Jacobian Function

```

1 function [detJ, dN, N, w] = jacobian_build(elemType, shapeFunc)
2
3     syms xi eta
4
5     N = shapeFunc;
6     dN = zeros(2, size(N, 2));
7
8     for i = 1:size(dN, 2)
9         dN(1,i) = diff(N(i), xi);
10        dN(2,i) = diff(N(i), eta);
11    end
12
13    %% 3 Nodes Triangular Element
14    if elemType == 'triangle3nodes'
15
16        syms x1 x2 x3 y1 y2 y3
17
18        w = 1/2;
19
20        d = [x1, y1;
21            x2, y2;
22            x3, y3];
23
24        J = dN * d;
25        detJ = abs(det(J));
26
27        N = double(subs(N, {xi, eta}, {1/3, 1/3}));
28    end
29
30 end

```

C.3 Jacobian Determinant Evaluation Function

```

1 function [detJ, node1, node2, node3, numElem] = node_jac_matrix(X, detJnum, xIni,
   yIni)
2
3     syms x1 x2 x3 y1 y2 y3
4
5     Nodes = readtable('node.csv');
6
7     numElem = height(Nodes);
8
9     % Node number
10    node1 = table2array(Nodes(:,2));
11    node2 = table2array(Nodes(:,3));
12    node3 = table2array(Nodes(:,4));
13
14
15    for i=1:numElem
16        detJ(i) = double(subs(detJnum, {x1, y1;
17                                x2, y2;
18                                x3, y3}, {X(node1(i),xIni), X(node1(i),yIni);
19                                                X(node2(i),xIni), X(node2(i),yIni);
20                                                X(node3(i),xIni), X(node3(i),yIni)}));
21    end
22 end

```

C.4 Build Elasticity Matrix Function

```

1 function [D] = assemble_D(N, node1, node2, node3, X, Y, Z1, Z2, detJ, numElem, ...
2     S11, S22, S12, w)
3
4 % Position of the nodes
5 coord = [X(:,5) X(:,6)];
6
7 % Size of the image and the total Area
8 maxX = max(coord(:,1));
9 maxY = max(coord(:,2));
10 A = maxX*maxY;
11
12 D = zeros(3,3);
13
14 %%% For 3 Node Triangle %%%
15 for i=1:numElem

```

```

16  % First Column
17  D(1,1)=D(1,1)+((X(node1(i),S11)*N(1) + ...
18                X(node2(i),S11)*N(2) + ...
19                X(node3(i),S11)*N(3)) * ...
20                w * detJ(i));
21  D(2,1)=D(2,1)+((X(node1(i),S22)*N(1) + ...
22                X(node2(i),S22)*N(2) + ...
23                X(node3(i),S22)*N(3)) * ...
24                w * detJ(i));
25  D(3,1)=D(3,1)+((X(node1(i),S12)*N(1) + ...
26                X(node2(i),S12)*N(2) + ...
27                X(node3(i),S12)*N(3)) * ...
28                w * detJ(i));
29  % Second Column
30  D(1,2)=D(1,2)+((Y(node1(i),S11)*N(1) + ...
31                Y(node2(i),S11)*N(2) + ...
32                Y(node3(i),S11)*N(3)) * ...
33                w * detJ(i));
34  D(2,2)=D(2,2)+((Y(node1(i),S22)*N(1) + ...
35                Y(node2(i),S22)*N(2) + ...
36                Y(node3(i),S22)*N(3)) * ...
37                w * detJ(i));
38  D(3,2)=D(3,2)+((Y(node1(i),S12)*N(1) + ...
39                Y(node2(i),S12)*N(2) + ...
40                Y(node3(i),S12)*N(3)) * ...
41                w * detJ(i));
42  % Third Column
43  D(1,3)=D(1,3)+(((Z1(node1(i),S11)*N(1) + ...
44                Z1(node2(i),S11)*N(2) + ...
45                Z1(node3(i),S11)*N(3)) * ...
46                w * detJ(i)) + ...
47                ((Z2(node1(i),S11)*N(1) + ...
48                Z2(node2(i),S11)*N(2) + ...
49                Z2(node3(i),S11)*N(3)) * ...
50                w * detJ(i)))/2;
51  D(2,3)=D(2,3)+(((Z1(node1(i),S22)*N(1) + ...
52                Z1(node2(i),S22)*N(2) + ...
53                Z1(node3(i),S22)*N(3)) * ...
54                w * detJ(i)) + ...
55                ((Z2(node1(i),S22)*N(1) + ...
56                Z2(node2(i),S22)*N(2) + ...
57                Z2(node3(i),S22)*N(3)) * ...
58                w * detJ(i)))/2;
59  D(3,3)=D(3,3)+(((Z1(node1(i),S12)*N(1) + ...
60                Z1(node2(i),S12)*N(2) + ...
61                Z1(node3(i),S12)*N(3)) * ...
62                w * detJ(i)) + ...
63                ((Z2(node1(i),S12)*N(1) + ...
64                Z2(node2(i),S12)*N(2) + ...

```



```
65         Z2 (node3 (i), S12) * N (3) * ...
66         w * detJ (i)) / 2;
67     end
68
69     D = D / A;
70
71 end
```

References

- [1] Atom - atomic bonds | britannica. <https://www.britannica.com/science/atom/Atomic-bonds>. (Accessed on 03/26/2021).
- [2] Introduction. <https://imagej.nih.gov/ij/docs/intro.html>. (Accessed on 04/20/2021).
- [3] Krita | digital painting. creative freedom. <https://krita.org/en/>. (Accessed on 06/08/2021).
- [4] Nondestructive evaluation physics : Materials. <https://www.nde-ed.org/Physics/Materials/Mechanical/Toughness.xhtml>. (Accessed on 04/19/2021).
- [5] Overview. <https://www.ctcms.nist.gov/~langer/oof2man/Chapter-Overview.html#Introduction>. (Accessed on 04/20/2021).
- [6] Propriedades e aplicações | aprodinox. <http://www.aprodinox.org.br/site/propriedades-e-aplicacoes-acos-util-2205.php>. (Accessed on 03/26/2021).
- [7] E3 - hardness. In M.J. NEALE, editor, *Tribology Handbook (Second Edition)*, page E3.1. Butterworth-Heinemann, Oxford, second edition edition, 1995.
- [8] Alfred R Akisanya, Udoka Obi, and Neill C Renton. Effect of ageing on phase evolution and mechanical properties of a high tungsten super-duplex stainless steel. *Materials Science and Engineering: A*, 535:281–289, 2012.
- [9] Salem Saleh Al-Amri, Namdeo V Kalyankar, et al. Image segmentation by using threshold techniques. *arXiv preprint arXiv:1005.4020*, 2010.
- [10] Yousef Al-Kofahi, Wiem Lassoued, William Lee, and Badrinath Roysam. Improved automatic detection and segmentation of cell nuclei in histopathology images. *IEEE Transactions on Biomedical Engineering*, 57(4):841–852, 2009.
- [11] Andrew J Alexander and Philip J Camp. Non-photochemical laser-induced nucleation. *The Journal of chemical physics*, 150(4):040901, 2019.
- [12] S Atamert and JE King. Super duplex stainless steels part 1 heat affected zone microstructures. *Materials Science and Technology*, 8(10):896–912, 1992.
- [13] S Atamert and JE King. Sigma-phase formation and its prevention in duplex stainless steels. *Journal of materials science letters*, 12(14):1144–1147, 1993.

- [14] NR Baddoo. Stainless steel in construction: A review of research, applications, challenges and opportunities. *Journal of constructional steel research*, 64(11):1199–1206, 2008.
- [15] Riad Badji, Mabrouk Bouabdallah, Brigitte Bacroix, Charlie Kahloun, Brahim Belkessa, and Halim Maza. Phase transformation and mechanical behavior in annealed 2205 duplex stainless steel welds. *Materials Characterization*, 59(4):447–453, 2008.
- [16] Saber Mirzaee Bafti, Chee Siang Ang, Md Moinul Hossain, Gianluca Marcelli, Marc Alemany-Fornes, and Anastasios D Tsaousis. A crowdsourcing semi-automatic image segmentation platform for cell biology. *Computers in Biology and Medicine*, 130:104204, 2021.
- [17] Craig R Barrett, William D Nix, William D Nix, AS Tetelman, et al. *The principles of engineering materials*. Prentice hall, 1973.
- [18] Lamia Jaafar Belaid and Walid Mourou. Image segmentation: a watershed transformation algorithm. *Image Analysis & Stereology*, 28(2):93–102, 2009.
- [19] Serge Beucher and Fernand Meyer. The morphological approach to segmentation: the watershed transformation. *Mathematical morphology in image processing*, 34:433–481, 1993.
- [20] Harry Bhadeshia and Robert Honeycombe. *Steels: microstructure and properties*. Butterworth-Heinemann, 2017.
- [21] Reidar Bjorhovde. Development and use of high performance steel. *Journal of Constructional Steel Research*, 60(3-5):393–400, 2004.
- [22] Irene Calliari, Marco Breda, Claudio Gennari, Luca Pezzato, Massimo Pellizzari, and Andrea Zambon. Investigation on solid-state phase transformations in a 2510 duplex stainless steel grade. *Metals*, 10(7):967, 2020.
- [23] William D Callister and David G Rethwisch. *Materials science and engineering: an introduction*, volume 9. Wiley New York, 2018.
- [24] Andrew Campbell, Paul Murray, Evgenia Yakushina, Stephen Marshall, and William Ion. New methods for automatic quantification of microstructural features using digital image processing. *Materials & Design*, 141:395–406, 2018.
- [25] Kai Wang Chan and Sie Chin Tjong. Effect of secondary phase precipitation on the corrosion behavior of duplex stainless steels. *Materials*, 7(7):5268–5304, 2014.
- [26] Jacques Charles. Duplex stainless steels-a review after dss ‘07 held in grado. *Steel Research International*, 79(6):455–465, 2008.
- [27] Kuan-Neng Chen, Andy Fan, and Rafael Reif. Microstructure examination of copper wafer bonding. *Journal of Electronic Materials*, 30(4):331–335, 2001.
- [28] TH Chen, KL Weng, and JR Yang. The effect of high-temperature exposure on the microstructural stability and toughness property in a 2205 duplex stainless steel. *Materials Science and Engineering: A*, 338(1-2):259–270, 2002.
- [29] Heng-Da Cheng, X_ H_ Jiang, Ying Sun, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern recognition*, 34(12):2259–2281, 2001.
- [30] Joseph R Davis et al. *Stainless steels*. ASM international, 1994.

- [31] Omer Demirkaya, Musa H Asyali, and Prasanna K Sahoo. *Image processing with MATLAB: applications in medicine and biology*. CRC Press, 2008.
- [32] Petra Erdely, Peter Staron, Emad Maawad, Norbert Schell, Joachim Klose, Helmut Clemens, and Svea Mayer. Design and control of microstructure and texture by thermomechanical processing of a multi-phase tial alloy. *Materials & Design*, 131:286–296, 2017.
- [33] DM Escriba, E Materna-Morris, RL Plaut, and AF Padilha. Chi-phase precipitation in a duplex stainless steel. *Materials Characterization*, 60(11):1214–1219, 2009.
- [34] DA Fadare, TG Fadara, and OY Akanbi. Effect of heat treatment on mechanical properties and microstructure of nst 37-2 steel. 2011.
- [35] King-Sun Fu and JK Mui. A survey on image segmentation. *Pattern recognition*, 13(1):3–16, 1981.
- [36] Elisa Drelie Gelasca, Jiyun Byun, Boguslaw Obara, and BS Manjunath. Evaluation and benchmark for biological image segmentation. In *2008 15th IEEE International Conference on Image Processing*, pages 1816–1819. IEEE, 2008.
- [37] N Haghddadi, D Abou-Ras, P Cizek, PD Hodgson, AD Rollett, and H Beladi. Austenite-ferrite interface crystallography dependence of sigma phase precipitation using the five-parameter characterization approach. *Materials Letters*, 196:264–268, 2017.
- [38] A Zarei Hanzaki, R Pandi, Peter D Hodgson, and S Yue. Continuous cooling deformation testing of steels. *Metallurgical and Materials Transactions A*, 24(12):2657–2665, 1993.
- [39] Yan-Lin He, Na-Qiong Zhu, Xiao-Gang Lu, and Lin Li. Experimental and computational study on microstructural evolution in 2205 duplex stainless steel during high temperature aging. *Materials Science and Engineering: A*, 528(2):721–729, 2010.
- [40] John P Hirth. The influence of grain boundaries on mechanical properties. *Metallurgical Transactions*, 3(12):3047–3067, 1972.
- [41] M. F. Horstemeyer. *Multiscale Modeling: A Review*, pages 87–135. Springer Netherlands, Dordrecht, 2010.
- [42] Chi-Shang Huang and Chia-Chang Shih. Effects of nitrogen and high temperature aging on σ phase precipitation of duplex stainless steel. *Materials Science and Engineering: A*, 402(1-2):66–75, 2005.
- [43] Fish Jacob and Belytschko Ted. *A first course in finite elements*. Wiley, 2007.
- [44] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.
- [45] Shehroz S Khan and Amir Ahmad. Cluster center initialization algorithm for k-means clustering. *Pattern recognition letters*, 25(11):1293–1302, 2004.
- [46] Y Kim. Phase transformation in cast duplex stainless steels [thesis]. Ames: Iowa State University, 2004.
- [47] Yoon-Jun Kim, L Scott Chumbley, and Brian Gleeson. Continuous cooling transformation in cast duplex stainless steels cd3mn and cd3mwcun. *Journal of materials engineering and performance*, 17(2):234–239, 2008.

- [48] Satendra Kumar, Mamta Kathuria, Alok Kumar Gupta, and Monika Rani. Fuzzy clustering of web documents using equivalence relations and fuzzy hierarchical clustering. In *2012 CSI Sixth International Conference on Software Engineering (CONSEG)*, pages 1–5. IEEE, 2012.
- [49] GL Leone and HW Kerr. The ferrite to austenite transformation in stainless steels. *Welding Journal*, 61(1):13S–21S, 1982.
- [50] MH Lewis. Precipitation of (fe, cr) sigma phase from austenite. *Acta Metallurgica*, 14(11):1421–1428, 1966.
- [51] Xiuyun Li, Jie Yang, Qing Wang, Jinjin Fan, and Peng Liu. Research and application of improved k-means algorithm based on fuzzy feature selection. In *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 1, pages 401–405. IEEE, 2008.
- [52] Thaddeus B. Massalski. *Theorie zu Phase Diagrams*. pages 6842–6850, 1992.
- [53] Raymond T Ng and Jiawei Han. Efficient and effective clustering methods for spatial data mining. In *Proceedings of VLDB*, pages 144–155. Citeseer, 1994.
- [54] J-O Nilsson, P Kangas, A Wilson, and T Karlsson. Mechanical properties, microstructural stability and kinetics of σ -phase formation in 29cr-6ni-2mo-0.38 n superduplex stainless steel. *Metallurgical and materials Transactions A*, 31(1):35–45, 2000.
- [55] Z Oksiuta, P Hosemann, SC Vogel, and N Baluc. Microstructure examination of fe–14cr ods ferritic steels produced through different processing routes. *Journal of Nuclear Materials*, 451(1-3):320–327, 2014.
- [56] G Palumbo, A Piccininni, and P Guglielmi. Prediction of the residual state of stress in a superduplex stainless steel produced by sand casting (using a coupled thermo-mechanical approach). *The International Journal of Advanced Manufacturing Technology*, 107(7):3011–3022, 2020.
- [57] Ann Kristin Petersen and Birger Solberg. Greenhouse gas emissions, life-cycle inventory and cost-efficiency of using laminated wood instead of steel construction.: Case: beams at gardermoen airport. *Environmental Science & Policy*, 5(2):169–182, 2002.
- [58] Dzung L Pham, Chenyang Xu, and Jerry L Prince. Current methods in medical image segmentation. *Annual review of biomedical engineering*, 2(1):315–337, 2000.
- [59] Michael Pohl, Oliver Storz, and Thomas Glogowski. Effect of intermetallic precipitations on the properties of duplex stainless steel. *Materials characterization*, 58(1):65–71, 2007.
- [60] Bernhard Preim and Charl Botha. Chapter 4 - image analysis for medical visualization. In Bernhard Preim and Charl Botha, editors, *Visual Computing for Medicine (Second Edition)*, pages 111–175. Morgan Kaufmann, Boston, second edition edition, 2014.
- [61] AJ Ramirez, JC Lippold, and SD Brandi. The relationship between chromium nitride and secondary austenite precipitation in duplex stainless steels. *Metallurgical and materials transactions A*, 34(8):1575–1597, 2003.
- [62] Robert E Reed-Hill, Reza Abbaschian, and Reza Abbaschian. *Physical metallurgy principles*, volume 17. Van Nostrand New York, 1973.

- [63] Jadwiga Rogowska. 5 - overview and fundamentals of medical image segmentation. In ISAAC N. BANKMAN, editor, *Handbook of Medical Imaging*, Biomedical Engineering, pages 69–85. Academic Press, San Diego, 2000.
- [64] Bharath Sathya and R Manavalan. Image segmentation by clustering methods: performance analysis. *International Journal of Computer Applications*, 29(11):27–32, 2011.
- [65] Y.S. Sato, T.W. Nelson, C.J. Sterling, R.J. Steel, and C.-O. Pettersson. Microstructure and mechanical properties of friction stir welded saf 2507 super duplex stainless steel. *Materials Science and Engineering: A*, 397(1):376–384, 2005.
- [66] Peter Smith. Chapter 3 - metallic materials for piping components. In Peter Smith, editor, *The Fundamentals of Piping Design*, pages 115–136. Gulf Publishing Company, 2007.
- [67] John M Sosa, Daniel E Huber, Brian Welk, and Hamish L Fraser. Development and application of mipar™: a novel software package for two-and three-dimensional microstructural characterization. *Integrating Materials and Manufacturing Innovation*, 3(1):123–140, 2014.
- [68] RO Sousa, P Lacerda, PJ Ferreira, and LMM Ribeiro. On the precipitation of sigma and chi phases in a cast super duplex stainless steel. *Metallurgical and Materials Transactions A*, 50(10):4758–4778, 2019.
- [69] J Tiley, T Searles, E Lee, S Kar, Russ Banerjee, JC Russ, and HL Fraser. Quantification of microstructural features in α/β titanium alloys. *Materials Science and Engineering: A*, 372(1-2):191–198, 2004.
- [70] Orlando José Tobias and Rui Seara. Image segmentation by histogram thresholding using fuzzy sets. *IEEE transactions on Image Processing*, 11(12):1457–1465, 2002.
- [71] Santina Topolska and Jerzy Łabanowski. Effect of microstructure on impact toughness of duplex and superduplex stainless steels. *Journal of Achievements in Materials and Manufacturing Engineering*, 36(2):142–149, 2009.
- [72] P Vasantharaja, M Vasudevan, and P Palanichamy. Effect of welding processes on the residual stress and distortion in type 316ln stainless steel weld joints. *Journal of Manufacturing Processes*, 19:187–193, 2015.
- [73] Dan Wang, Penghuai Fu, Liming Peng, Yingxin Wang, and Wenjiang Ding. Development of high strength sand cast mg–gd–zn alloy by co-precipitation of the prismatic β and β_1 phases. *Materials Characterization*, 153:157–168, 2019.
- [74] Zhiping Xiong, Ilana Timokhina, and Elena Pereloma. Clustering, nano-scale precipitation and strengthening of steels. *Progress in Materials Science*, page 100764, 2020.
- [75] Shimon D Yanowitz and Alfred M Bruckstein. A new method for image segmentation. *Computer Vision, Graphics, and Image Processing*, 46(1):82–95, 1989.
- [76] Yu Jin Zhang. A survey on evaluation methods for image segmentation. *Pattern recognition*, 29(8):1335–1346, 1996.
- [77] Ji-Cheng Zhao and Michael R Notis. Continuous cooling transformation kinetics versus isothermal transformation kinetics of steels: a phenomenological rationalization of experimental observations. *Materials Science and Engineering: R: Reports*, 15(4-5):135–207, 1995.