Universidade de Aveiro Departamento de Engenharia Mecânica
2012

Raquel Cristina Soares de Carvalho e Silva

Desenvolvimento de metodologias para identificação de parâmetros e otimização de forma em simulações numéricas de processos de conformação plástica

Development of numerical methodologies for parameter identification and shape optimization in metal forming simulations

Universidade de Aveiro Departamento de Engenharia Mecânica
2012

Raquel Cristina Soares de Carvalho e Silva

Desenvolvimento de metodologias para identificação de parâmetros e otimização de forma em simulações numéricas de processos de conformação plástica

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Mecânica, realizada sob a orientação científica do Doutor António Andrade-Campos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro, e do Doutor Robertt Valente, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Apoio financeiro da FCT – Fundação para a Ciência e a Tecnologia, Ministério da Educação e Ciência – através da bolsa SFRH / BD / 45588 / 2008.

I dedicate the present work to my parents, sister and Marco.

My parents and sister, who knew at all the moments of my life to give me, wisely and friendly, the force required to face the challenges of life, making me who I am today. For your endless support and love, my eternal gratitude.

Marco, for all love and friendship shown during this journey. For all patience, calmness and serenity that helped me overcome all obstacles. For everything my eternal gratitude.

o júri

presidente	Prof. Dr. Maria Hermínia Deulonder Correia Amado Laurel professora catedrática da Universidade de Aveiro
	Prof. Dr. Luís Filipe Martins Menezes professor catedrático da Faculdade de Ciências e Tecnologia da Universidade de Coimbra
	Prof. Dr. Paulo António Firme Martins professor catedrático do Instituto Superior Técnico da Universidade Técnica de Lisboa
	Prof. Dr. Sandrine Thuillier professora com habilitação da Université de Bretagne Sud de França
	Prof. Dr. Abel Dias dos Santos professor auxiliar da Faculdade de Engenharia da Universidade do Porto
	Prof. Dr. Marta Cristina Cardoso Oliveira professora auxiliar da Faculdade de Ciências e Tecnologia da Universidade de Coimbra
	Prof. Dr. António Gil D´Orey de Andrade-Campos professor auxiliar convidado da Universidade de Aveiro
	Prof. Dr. Robertt Ângelo Fontes Valente professor auxiliar da Universidade de Aveiro

acknowledgements

Although this work is of an individual nature, it would not have been possible without the collaboration of many people who directly or indirectly contributed to its realization. Therefore, I would like to express my gratitude to all those who helped me during this journey.

Thus, I start thanking Doctor António Andrade-Campos for the excellent opportunity to perform my PhD in the optimization area and for always believed in my work. I thank especially the friendship, dedication and availability, as well as guidelines and teachings have always provided.

To Doctor Robertt Valente for the support, dedication and the constant optimism that made me believe in this work and were essential in its development.

To Doctor Sandrine Thuillier, Engineer Antoine and all that had received me at the Laboratoire de Génie Méchanique et Matériaus (LG2M), where I developed the whole experimental part of this work.

To Doctor José Grácio for the continuous support and for receiving me so well at the Department of Mechanical Engineering at University of Aveiro.

To Engineer António Festas for his availability performing the experimental specimens.

To Doctor Marta Oliveira and Engineer Diogo Neto for the support and teachings in the tool numerical implementation.

To all my friends for their support and especially by joyful moments and relaxation throughout my journey.

In particular to Dora, Joana, Guilhermina, Lé, Marisa, Teresa and Tico for all the support provided at various stages of my life and the memorable moments, with each one I lived. It is with a smile that I thank the special friends they are.

To my friends from GRIDS, in special to João Caseiro for all the support during my journey.

To all my family, who always supported me during all my life.

Palavras-chave

metodologias inversas, processos de conformação plástica, Método dos Elementos Finitos (MEF), algoritmos de otimização, identificação de parâmetros, otimização da forma inicial de chapas, otimização da forma de ferramentas.

resumo

Por parte da indústria de estampagem tem-se verificado um interesse crescente em simulações numéricas de processos de conformação de chapa, incluindo também métodos de engenharia inversa. Este facto ocorre principalmente porque as técnicas de tentativa-erro, muito usadas no passado, não são mais competitivas a nível económico. O uso de códigos de simulação é, atualmente, uma prática corrente em ambiente industrial, pois os resultados tipicamente obtidos através de códigos com base no Método dos Elementos Finitos (MEF) são bem aceites pelas comunidades industriais e científicas

Na tentativa de obter campos de tensão e de deformação precisos, uma análise eficiente com o MEF necessita de dados de entrada corretos, como geometrias, malhas, leis de comportamento não-lineares, carregamentos, leis de atrito, *etc.*. Com o objetivo de ultrapassar estas dificuldades podem ser considerados os problemas inversos. No trabalho apresentado, os seguintes problemas inversos, em Mecânica computacional, são apresentados e analisados: (i) problemas de identificação de parâmetros, que se referem à determinação de parâmetros de entrada que serão posteriormente usados em modelos constitutivos nas simulações numéricas e (ii) problemas de definição geométrica inicial de chapas e ferramentas, nos quais o objetivo é determinar a forma inicial de uma chapa ou de uma ferramenta tendo em vista a obtenção de uma determinada geometria após um processo de conformação.

São introduzidas e implementadas novas estratégias de otimização, as quais conduzem a parâmetros de modelos constitutivos mais precisos. O objetivo destas estratégias é tirar vantagem das potencialidades de cada algoritmo e melhorar a eficiência geral dos métodos clássicos de otimização, os quais são baseados em processos de apenas um estágio. Algoritmos determinísticos, algoritmos inspirados em processos evolucionários ou mesmo a combinação destes dois são usados nas estratégias propostas. Estratégias de cascata, paralelas e híbridas são apresentadas em detalhe, sendo que as estratégias híbridas consistem na combinação de estratégias em cascata e paralelas.

São apresentados e analisados dois métodos distintos para a avaliação da função objetivo em processos de identificação de parâmetros. Os métodos considerados são uma análise com um ponto único ou uma análise com elementos finitos. A avaliação com base num único ponto caracteriza uma quantidade infinitesimal de material sujeito a uma determinada história de deformação. Por outro lado, na análise através de elementos finitos, o modelo constitutivo é implementado e considerado para cada ponto de integração.

Problemas inversos são apresentados e descritos, como por exemplo, a definição geométrica de chapas e ferramentas.

Considerando o caso da otimização da forma inicial de uma chapa metálica a definição da forma inicial de uma chapa para a conformação de um elemento de cárter é considerado como problema em estudo. Ainda neste âmbito, um estudo sobre a influência da definição geométrica inicial da chapa no processo de otimização é efetuado. Este estudo é realizado considerando a formulação de NURBS na definição da face superior da chapa metálica, face cuja geometria será alterada durante o processo de conformação plástica.

No caso dos processos de otimização de ferramentas, um processo de forjamento a dois estágios é apresentado. Com o objetivo de obter um cilindro perfeito após o forjamento, dois métodos distintos são considerados. No primeiro, a forma inicial do cilindro é otimizada e no outro a forma da ferramenta do primeiro estágio de conformação é otimizada. Para parametrizar a superfície livre do cilindro são utilizados diferentes métodos. Para a definição da ferramenta são também utilizados diferentes parametrizações.

As estratégias de otimização propostas neste trabalho resolvem eficientemente problemas de otimização para a indústria de conformação metálica.

keywords

inverse methodologies, metal forming processes, Finite Element Method (FEM), optimization algorithm, parameters identification, initial blank shape optimization, tool shape optimization.

abstract

The interest of the stamping industry in the numerical simulation of sheet metal forming, including inverse engineering approaches, is increasing. This fact occurs mainly because trial and error design procedures, commonly used in the past, are no longer economically competitive. The use of simulation codes is currently a common practice in the industrial forming environment, as the results typically obtained by means of the Finite Element Method (FEM) are well accepted by both the industrial and scientific communities.

In order to obtain accurate stress and strain fields, an effective FEM analysis requires reliable input data such as geometry, mesh, non-linear material behaviour laws, loading cases, friction laws, *etc.*. In order to overcome these difficulties, a possible approach is based on inverse problems. In this work, the following inverse problems in computational Mechanics are presented and analysed: (i) parameter identification problem, that refer to the definition of input parameters to be used in constitutive models for numerical simulations, based on experimental data, and (ii) initial blank and tool design problem, where the aim would be to estimate the initial shape of a blank or a tool in order to achieve the desired geometry after the forming process.

New optimization strategies in parameter identification problems that lead more efficiently to accurate material parameters are introduced and implemented. The aim of these strategies is to take advantage of the strength of each selected algorithm and improve the overall robustness and efficiency of classical optimization methodologies based on single stages. Deterministic algorithms, evolutionary-inspired algorithms or even the combination of these two algorithms are used in the proposed strategies. Strategies such as cascade, parallel and hybrid approaches are analysed in detail. In hybrid strategies, cascade and parallel approaches are integrated.

Two different approaches are presented and analyzed for the evaluation of the objective functions in parameter identification processes. The approaches considered are single-point and FE analyses. The single infinitesimal point evaluation seems to characterize an infinitesimal amount of material subjected to all kind of deformation history. On the other hand, in all FE analysis codes, the constitutive model is implemented and accounted for in each element integration point.

Inverse problems, such as blank and tool design, are presented and described. In the case of the initial blank optimization process the design of a carter is presented. Also related to the initial blank optimization process, a study of the influence of the initial geometry definition in the optimization process is conducted. This study is performed considering the NURBS formulation to model the blank upper surface that will be changed during the optimization process.

In the case of the tool design problem, a two-stage forging process is presented. In order to achieve a straight cylinder after forging, two different approaches are analyzed. In the first one, the initial geometry of the cylinder is optimized and, in the other one, the shape of the first stage tool is optimized. To parameterize the free surface of the cylinder different methods are presented. Furthermore, in order to define the tool in this example, different parameterizations are presented.

The optimisation strategies proposed in this work efficiently solve optimisation problems for the industrial metal forming.

“Every great advance in science has issued
from a new audacity of imagination.”

John Dewey

Contents

List of Figures

List of Tables

I Outline

Chapter 1

Introduction

An introduction to the presented work is performed. This introduction comprises an historic background, the main objectives as well as the importance of the present work in the current reality. Themes like metal forming processes, Finite Element Method, definition of inverse problems and the coupling between FEM and optimization are also introduced. A reading guide and a list of the publications, published in the scope of this thesis, is provided.

1.1 Background

The Industrial Revolution was an important mark that completely changed the course of the Humankind. It was a period from 1750 to 1850, in which a set of technological changes were registered with an intense impact in the social, economic and cultural conditions of those times. It had begun in the United Kingdom and subsequently spread to the Western Europe, North America, Japan and then to the rest of the world. Until the Industrial Revolution, manufacturing was done at a job-shop level. In those times the demanding of manufactured needs was everyday increasing and with the advent of new power sources, came the ability to manufacture on a larger scale. Innovation and automation progressively increased with an accompanying increase in the complexity of products [1]. With the Industrial Revolution, metal forming processes appear and with the passing of times the evolution of the machining and the standardization of the processes lead to a huge increase of the forming process knowledge. However, during those times, the possibility to perform simulations of sheet forming processes was an unattainable desire in the sheet forming industry.

It was in the 1960s that the first attempts of a numerical approximate solution of sheet metal forming processes were performed. This first numerical solution was obtained by Finite Difference Methods. These methods could not establish themselves due to the serious drawback of not applying boundary conditions in a general manner as it could be done elegantly in the Finite Element Method (FEM) [2].

The real breakthrough of metal forming simulation emerges with the appearance of the FEM. The first reference to the well known designation of Finite Element Method emerges in 1960 with the work of Ray Clough. This work was about the analysis of elastic problems in plane stress. In its original paper, Ray Clough [3] defined the nomenclature that is still on use, and established systematic procedures that are in the basis of FEM's computational implementation, in the resolution of discrete problems [4]. In those times, process engineers wanted to minimize the need for expensive modifications of the tool in trial-and-error processes. This minimization was only possible if the process engineers were able to reveal any possible forming defects at an early stage. To accomplish this, engineers needed an accurate metal forming simulation.

In the early days of FEM, in the 1960s and early 1970s, the theoretical perspective of the forming processes was considered a difficult task because modelling sheet metal forming requires accurate characterization of effects, such as nonlinear material behaviour, large deformations and complex contact condition between the tool and the blank. It was in 1970 that the first theoretically accurate FE-formulation of the large deformation problem was presented by Hibbitt *et al.* [5]. This formulation was written in a Total Lagrangian formulation (TL-formulation). In 1975, McMeeking and Rice [6], and Bath *et al.* [7] proposed an Updated Lagrangian formulation (UL-formulation). In those times, forming processes simulation were considered as either plane-strain or axisymmetric problems, due to simplification reasons [8].

It was in 1978 that the first theoretically precise 3D formulation of a sheet metal forming process was proposed by Wang and Budiansky [9]. Also in this year, other pioneering works appear, such as the works of Gotoh and Ishire [10] and Wifi [11]. The following decade had seen a high activity in the field. The first 3D applications are known by Tang *et al.* [12] and Toh and Kobayashi [13]. In those times, a differentiation between different approaches used to apply the FEM analysis was considered based on the choice of motion description, the type of constitutive relations and the solution procedures. Six main groups were considered, such as, the *solid* approach, the *static-implicit* approach, the *rigid-plastic* approach, the *static-explicit* approach and the *flow* approach. All the studies previously referred are of *static-implicit* or *static-explicit* type.

In the 1980s, the *flow* and *rigid-plastic* approaches were more common than the *static-implicit* one. This fact occurred mainly because the *flow* and the *rigid-plastic* formulations were more stable and allowed considerably larger time-steps. During those days, the practical application of the sheet metal forming simulations was computationally very expensive, even when small problems were considered.

Since the beginning of 1990s, it was verified a massive increase of the use of sheet forming simulations in the industrial community. In the middle of that decade most companies within the automotive industry were performing sheet forming simulations on a regular basis. In those times, dynamic and explicit codes were dominating the software market. General-purpose codes like LS-Dyna^{TM} and Abaqus®/Explicit, as well as specialized codes such as PAM-STAMP® and OPTRIS®, are examples of codes used in those times. Other codes based on *static-explicit* approaches were used, in the Japanese industrial environment. In the German automotive industry a few *static-implicit* codes were used, like INDEED®, and in the Ford Motor Company the MTLFORM® *static-implicit* code was considered for the metal forming simulations. The *static-implicit* codes were at those times (and still in the actuality) more used in the academic communities. In the early 1990s, the AutoForm® program was created from a research project at ETH in Zurich. AutoForm® is a highly specialized *static-implicit* code for stamping simulations and used some innovative algorithms to enhance stability and computational efficiency. In its original form, the code adopted bending-enhanced membrane elements and an iterative linear solver. However, with the time, the code was further developed to include conventional shell elements. This code, for forming applications, is competitive or even superior to dynamic and explicit codes when the efficiency and robustness are considered [8].

Since 2009, AutoForm® is the most commonly used code in the metal forming industry. In addition, the software market is still dominated by various dynamic-explicit codes like LS-DYNA^{TM}, Abaqus®, PAM-STAMP 2G® and STAMPACK®. In the industry, the use of other codes is now only marginal [8].

Nowadays the knowledge in the simulation of sheet metal forming is increasing, however the scientific and the industrial communities want more exactness and still have some goals to achieve. These goals can be summarized in three main groups, such as, (i) time reduction, (ii) cost reduction and (iii) increase of product quality. Time reduction encompasses the early checking of producibility of workpieces, the reduction of the development times, the reduction of the try-out times and the quick response to modification wishes. Cost reduction means having cheaper products, reduce the die costs and increase the reliability. The increase of product quality means the production of more complicated parts, the accumulation of know-how for new materials, the process repeatability and the optimization by variants. These are actual goals that will be achieved with the increasing of the accurateness of the FEM and optimization software [14]. Nowadays, the conjugation of FEM software with optimization software is a new trend in metal forming industry in which the companies are expending their time and money. It is based on this idea that the present work is proposed, trying to expose some optimization strategies and new optimization procedures in the sheet metal forming domain.

1.2 Aims and goals of the present work

The stamping industry is increasing more and more the interest in numerical simulation of metal forming processes. This interest appears because trial and error design procedures, based on experience previously obtained by engineers, lead to large economical costs and, consequently, lost of competitiveness. Nowadays it is possible to simulate with FEM software the complexities inherent to plastic forming processes such as large deformations, contact, friction, springback, wrinkles in formed parts, among others. Also, optimization procedures are more robust and the coupling between FEM software and optimization software is receiving a special attention in the last years.

It is based on this idea, and trying to increase the knowledge in these areas, that the present work was conducted. As the title suggests, the work has as the main objective the development of numerical methodologies for parameter identification and shape optimization in metal forming simulations. Considering this, it is possible to understand that “metal forming processes” is assumed as one of the keywords of this work.

In order to correctly simulate metal forming processes, it is imperative to use complex material models and reliable input data. For this reason, one of the present work objectives is the development and implementation of optimization methodologies able to correctly calculate the material parameters of a constitutive model suitable for a specific material.

Other objective of the present work relies on the fact that the use of only one optimization method do not usually lead to an efficient solution of the previous mentioned inverse problems. It is then fundamental to develop new methodologies accounting for more than a single type of optimization method. The study and implementation of these strategies is another target of this work. Special attention is given to gradient-based methods and evolutionary algorithms (EAs). These optimization methods are used in strategies that can faster lead to the final objective, by means of cascade, parallel and hybrid solution procedures. With the combination of different optimization algorithms it is possible to take advantage of the strength of each algorithm and increase the probability to reach the global minimum.

In the parameters optimization processes the evaluation of the objective function is one of the fundamental points that should be carefully considered. In the present work, it is intended to understand how the use of “single-point” or metal forming FE analysis in the evaluation of the objective function influences the optimization process. In the case of single infinitesimal point evaluation, all the material is considered as an infinitesimal point subjected to all kind of deformation history. Using FE analysis codes, the constitutive model is implemented and accounted for each element integration point. These two approaches are presented and analysed in the present work.

The determination of the initial blank shape is one of the most important optimization process, with a direct influence on the quality of the finished part, as well as reducing the final cost and product development time. Based on this idea, a methodology to find the optimized initial blank shape for a carter forming processes is proposed in the present work. In addition, a study that allows to understand the influence of the parametric geometry definition in the optimization process is presented.

The preform tool design is assumed to be the most important step for product quality control. It is considered that a reasonable preform tool shape reduces the raw material cost and improves material flow, reducing manufacturing cost and eliminating following processes. Consequently, preform tool shape optimization is still an interesting theme. However, one way to avoid the preform tool design is to determine an initial shape of the specimen to obtain the final objective. Therefore, these two approaches can be applied to the same goal. The last objective of the present work is to compare two different approaches to solve a specific inverse problem. Both approaches try to find a desirable final shape of a specimen after a forging process. The first approach optimizes the initial shape of the specimen and the second one optimize the preform tool shape. In the present work the results are compared for both approaches.

1.3 Metal forming processes

Nowadays, a diverse range of manufacturing methods are accessible in the industrial environment. The choice of which method will be used in order to produce a specific component should be performed considering the advantages and disadvantages of each method. The chosen method should be the one that provides a product with proper function and properties at the lowest cost [15]. Considering the norm DIN8580 [16], the manufacturing processes are classified into six main groups: primary shaping, material forming, dividing, joining, modifying material property and coating, as it can be seen in Figure 1.1 [17].

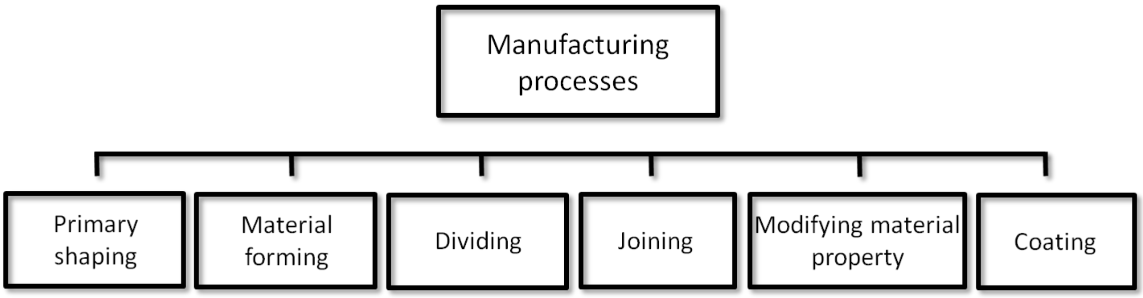


Figure 1.1 - Classification of manufacturing processes according to the norm DIN8580 [16].

The “primary shaping” consists in the creation of an initial shape from the molten, gaseous or formless solid state. Processes such casting, melt extrusion, die casting and pressing of metal powder are examples of primary shaping. “Dividing” is the local separation of material, and examples of this kind of processes are sawing, turning, milling and broaching. “Joining” is the assembly of individual pieces to create other components and also the filling and saturation of porous pieces. Examples of joining processes are welding, riveting and shrink fitting. “Modifying the material properties” intend to alter material characteristics of a component trying to achieve certain useful properties. Such processes include heat treatment processes such as hardening or recrystallization annealing. “Coating” consists in the application of thin layers on the component. Examples are galvanization, painting and foil wrapping. “Material forming processes” are processes where the material is formed by plastic deformation. Forming processes are classified in accordance with norm DIN 8582 [18] depending on the main direction of the applied stresses. They can be classified as [17]:

- Forming under compressive conditions;
- Forming under combined tensile and compressive conditions;
- Forming under tensile conditions;
- Forming by bending;
- Forming under shear conditions.

The DIN 8582 [18] differentiates 17 distinct forming processes according to the relative movement between die and workpiece, die geometry and workpiece geometry. For the category of forming under compressive conditions 5 processes are considered, such as, rolling, open die forming, closed die forming, coining and forming by forcing through an orifice. In the case of the forming under compressive and tensile conditions 5 processes are considered. The processes are stripping, deep drawing, flanging, spinning, wrinkle, wrinkle bulging. For the forming under tensile condition the processes considered are extending by stretching, expanding and stretch forming. In the case of forming by bending, the bending with linear die movement and the bending with rotary die movement processes are considered. For the forming under shearing conditions the displacement and twisting methods are considered [17]. In Figure 1.2 it is possible to see a graphical summary of these methods. Information about these methods can be found in references [15-18].

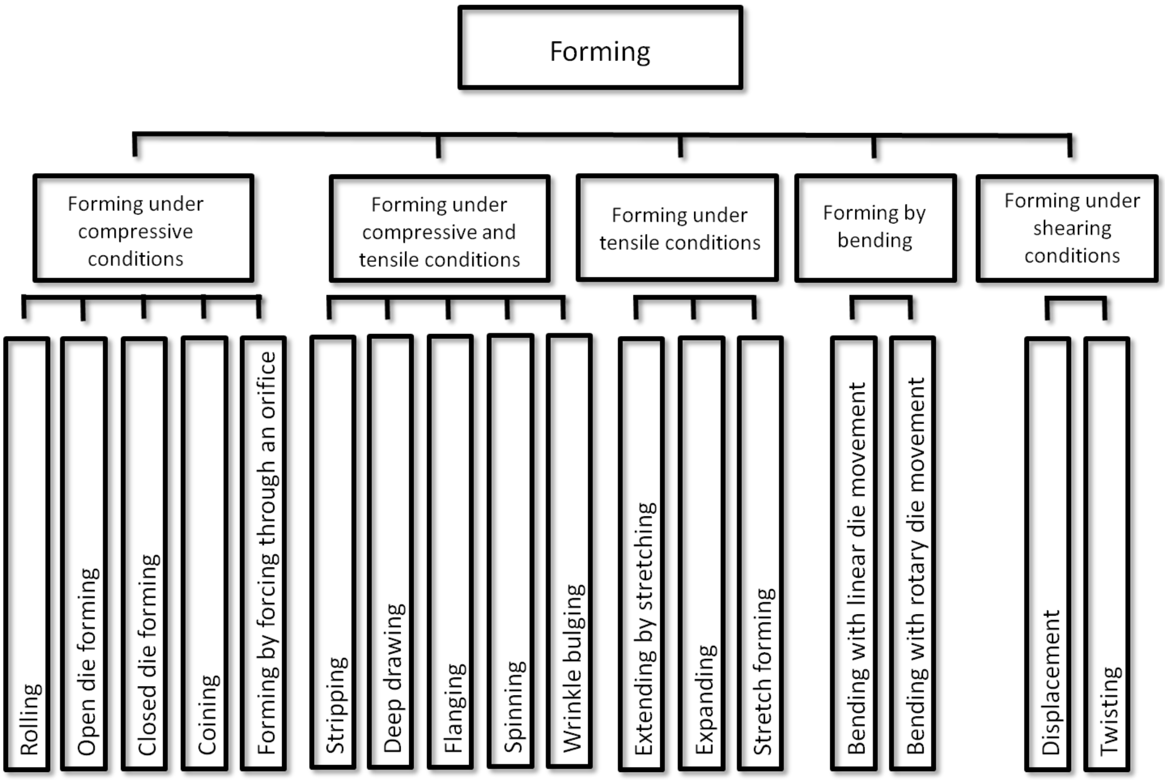

Forming
Forming under compressive conditions
Forming under compressive and tensile conditions
Forming under tensile conditions
Forming by bending
Forming under shearing conditions
Rolling
Open die forming
Closed die forming
Coining
Forming by forcing through an orifice
Stripping
Deep drawing
Flanging
Spinning
Wrinkle bulging
Extending by stretching
Expanding
Stretch forming
Bending with linear die movement
Bending with rotary die movement
Displacement
Twisting

Figure 1.2 - Classification of production processes used in forming, in accordance to DIN 8582 [18].

Metal forming processes have, among all manufacturing processes, a special place since they help to produce metal parts of superior mechanical properties with minimum waste of material [17,19]. For the metal forming processes, the material has a relatively simple initial geometry. This material is then plastically deformed, in one or more operations, into a component of relatively complex shape. However, metal forming usually requires relatively expensive tooling. Therefore, the process is economically attractive only when a large number of parts must be produced and/or when the mechanical properties required in the finished component can be obtained only by forming processes.

Forming processes are normally used together with other manufacturing processes, in order to complete the transformation from the raw material into the final component. The materials have as desirable properties low yield strength and high ductility. These properties are conditioned by the temperature and the rate of deformation. For higher temperatures, the ductility is increased and the yield strength is decreased. The temperature allows to define the forming processes as cold forming (when the component is initially at room temperature), warm forming (when the component is heated above room temperature and below the recrystallization temperature) and hot forming (when the component is heated above the recrystallization temperature) [19].

Metal forming, among other manufacturing processes, represents a highly significant group of processes for producing automotive, industrial, aerospace, packaging, military components, *etc.*. A few components produced by metal forming can be enumerated as following [19]:

- Components for industrial plants and equipments as well as for automobiles and machine;
- Hand tools, such as hammers, pliers and surgical instruments;
- Fasteners, such as screws, nuts and rivets;
- Containers, such as metal boxes, cams and canisters;
- Construction elements used in tunnelling, mining and quarrying;
- Fittings used in building industry, such as doors and windows.

Some examples of formed components are presented in Figure 1.3.

Figure 1.3 - Examples of some components obtained by metal forming processes.

For the design, analysis and optimization of forming processes an analytical knowledge regarding the metal flow, stresses and heat transfer is needed, as well as technological information related to lubrication, heating and cooling techniques, material handling, die design and manufacture and forming equipment. Most of this information can be found in the literature [19].

1.4 The Finite Element Method

As previously mentioned, the Finite Element Method (FEM) emerged in the early 1960s and at the present it is widely used in engineering analysis and, being expected that this use will increase significantly in the years to come [20]. The FEM is an indispensable technology that allows the modelling and simulation of workpieces for every field of engineering. Fields like housing, transportation, communications, packaging, automotive, among others are examples of fields where the FEM analysis is assumed as essential for the achievement of high quality products at low cost. At the beginning, FEM was used to solve problems of stress analysis and since that it has been applied to many other problems such as thermal analysis, fluid flow analysis, piezoelectric analysis, and many others. Therefore, the FEM is used to determine the distribution of some field variables like the displacement in stress analysis, the temperature or heat flux in thermal analysis, the electrical charge in electrical analysis, *etc.* [21].

The FEM is a numerical method and, as a consequence, is an approximation method. It is necessary to not forget that the FEM solves complex problems, however gives only an approximate solution. This kind of methods should be avoided in the case that simpler analytical solutions exist. Therefore, the method is used in the search of complex problems' solution from diverse fields of knowledge for which the exact solution doesn´t exist or cannot be expressed in an analytic way [4].

The way that a system behaves during a specific phenomenon depends upon the geometry or domain of the system, the properties of the material, the boundary and the initial and loading conditions. For a specific engineering system, the geometry as well as the boundary and initial conditions can be very complex. It is then very difficult to solve the governing differential equation with an analytical formulation, being necessary to solve the problem with a numerical procedure, such as the FEM [21].

From a user point of view, the computational application of the FEM consists of four steps:

- Modelling of the geometry;
- Meshing (discretization);
- Specification of the material properties;
- Specification of boundary, initial and loading conditions.

Real structures, components or domains are normally very complex and have to be reduced to a manageable geometry. The curved parts of the geometry are represented by a set of elements that tries to approximate the curves and curved surfaces by piecewise straight lines or flat surfaces if linear elements are used. Therefore, the accuracy of the representation of the curved parts is controlled by the number of elements used. More elements will allow a more accurate definition of the curved parts. Unfortunately, more elements means longer computational time to solve the problem. Therefore, the optimum number of elements should always be selected based in a good relation between the CPU time and the geometric accuracy given by the number of elements [21].

The geometry model can be created considering points, lines and curves, surfaces and solids. The points are created by keying the coordinates; the lines and curves can be created by connecting points; surfaces can be created by connecting, translating or rotating the existent lines or curves; and the solids can be created by connecting, translating or rotating the existent surfaces [21].

There are numerous Computer Aided Design (CAD) software packages that have graphic interfaces that help in the creation and manipulation of geometrical objects. For the modelling step the knowledge, experience and engineering judgment are keywords allowing the correct definition of the system. In some cases, finely detailed geometrical features play only an aesthetic role and have negligible effects on the performance of the engineering system. In these cases these features can be deleted, ignored or simplified. However, for other cases these detailed geometrical features are essential in the simulation results and it is the engineer that should evaluate these cases based on his experience [21].

Another important step is the meshing, which is performed in order to discretize the geometry created into small pieces called *elements*. The solution of a mechanical problem can be very complex and varies in a way that can be unpredictable using functions across the whole domain of the problem. However, the problem can be divided (meshed) into small elements that are connected by *nodes*. In this case the solution within an element can be approximated very easily using simple functions. The solution of the whole problem domain will be the combination of all solutions for all elements. An example of this discretization is presented in Figure 1.4, where a carter is divided in small elements.

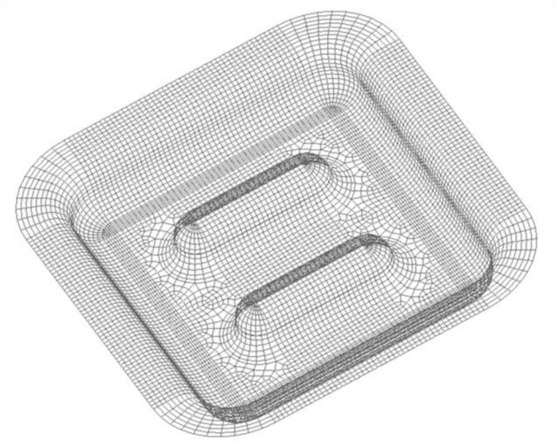


Figure 1.4 - Carter discretization in elements [22].

The mesh generation is an important task in the pre-process and can be very time consuming. During the mesh generation step, all domain is divided into small elements of specific shapes, such as for example in 2D, triangles and quadrilaterals. During this step also the information about the *element connectivity* must be created for a future use in the construction of the FEM equations. The use of triangular elements is commonly available in most of the pre-processors and this technique is flexible in the modelling of complex geometries and its boundaries. However, it have one disadvantage which is related to the fact that the results obtained are less accurate than the ones obtained with quadrilateral elements. Quadrilateral meshes, on the other hand, are generally more difficult to generate in an automated manner when compared with triangular meshes [21].

Most of the engineering systems consist of more than one material. The materials properties can be defined for a group of elements or for each individual element. The input of material's properties is normally straightforward and the user only needs to insert the material properties and associate them to a specific group of elements. The definition of these properties is, however, not always easy and there are commercially available material databases to choose from [21].

The correct specification of the boundary, initial and loading conditions is a decisive step for the correct application of the FEM. These conditions normally are easy to input in commercial pre-processors and often interfaced with graphics. The user can specify these conditions either to geometrical identities or to elements of the mesh. This is a step that requires experience, knowledge and proper engineering judgments and differs from problem to problem [21].

In summary, the application of the FEM to realistic engineering problems is conditioned by the capacity of the available computational devices. As a consequence, the development of the FEM is growing at the same time that the computers are getting computationally more powerful [4].

1.5 Definition of inverse problems

In order to obtain accurate stress and strain fields, an effective FEM analysis requires secure input data such as geometry, mesh, non-linear material behaviour laws, loading cases, friction laws, *etc*. This sort of problems can be defined as direct problems, in which the quality of the results relies on the quality of the input data that are not always available. In order to overcome these difficulties, a possible approach are the inverse problems, for instance, with the definition of input parameters to be used in geometric or constitutive models for numerical simulations, based on experimental data. The interest of the forming industry in inverse engineering approaches is increasing. This fact occurs mainly because trial and error design procedures, commonly used in the past, are no longer competitive [3]. Considering the need to evaluate the input data, distinct inverse problems can be formulated.

One category of inverse problems is called “parameter identification”. The aim of these problems, for instance, is to estimate material parameters for constitutive models. The development of new materials and the effort to characterize the existent materials lead to the formulation of new complex constitutive models. However, many of these constitutive models demand the determination of a large number of parameters adjusted to the material whose behaviour is to be simulated. In cases where the number of parameters is high, it might be necessary to solve the problem as a non-linear optimization problem. The parameters’ determination should always be performed confronting mathematical and experimental results.

The experimental data allows the determination of the different parameters needed for the mathematical formulation of the model. This can be accomplished solving an inverse problem which consists of searching for a set of parameter values for which the experimental reality and the numerical simulation are similar. The comparison between the mathematical model and the experimental data results in a function that must be evaluated and minimized. This will be the objective function of these problems [24-27].

The parameters identification problem can be reduced to a curve-fitting problem if physical constraints were not taken into account. However, most material constitutive models have physical constraints such as material parameter boundary values and mathematical relations between them, guaranteeing the physical meaning of the material parameters [28]. The formulation for the solution of the constitutive model parameters identification is as follows. First, a physical system whose behaviour can be described by a numerical model and which experimental results are available should be considered. A set of measurable variables that can be experimentally determined should also be considered. This set of variables is defined as $\mathbf{Z}^{\mathrm{T}} = [z_1, z_2, \dots, z_m]$. In the case that simple mechanical tests are considered (such as tensile and shear tests), these measurable variables would be the stresses or strains. Considering this information, then it is possible to formulate the solution of the identification problem as the minimization of a function that measures the difference between theoretical predictions $\mathbf{Z}^{\mathrm{num}}$

(obtained by the numerical model) and experimental data. This function is known as the objective function $\mathcal{L}(\mathbf{A})$, and can be formulated as [29,30]:

$$\mathcal{L}(\mathbf{A}) = \sum_{q=1}^{N} \mathcal{L}_q(\mathbf{A}), \tag{1.1}$$

with

$$\mathcal{L}_q(\mathbf{A}) = \frac{1}{(t_1 - t_0)} \int_{t_0}^{t_1} [\mathbf{Z}^{\mathrm{num}}(\mathbf{A}) - \mathbf{Z}^{\mathrm{exp}}]^{\mathrm{T}} \mathbf{D}_q [\mathbf{Z}^{\mathrm{num}}(\mathbf{A}) - \mathbf{Z}^{\mathrm{exp}}] \, \mathrm{d}t. \tag{1.2}$$

In equation 1.2, $\mathbf{A} = [A_1, A_2, \dots, A_r]^{\mathrm{T}}$ is the set of the $r \in \mathbb{N}$ constitutive model parameters and $\mathbf{Z}^{\mathrm{exp}}$ is the known experimental value of $\mathbf{Z}$ in the N experimental tests. The time period of the generic test q is given by (t_0, t_1). $\mathbf{D}_q$ is a given weight matrix associated to the test q and can be formulated as [28]:

$$\mathbf{D}_q = \frac{1}{W_q^{\mathrm{abs}} + W_q^{\mathrm{rel}} \mathbf{Z}^{\mathrm{exp}}}. \tag{1.3}$$

In equation 1.3, W_q^{abs} and W_q^{rel} are the absolute and relative weights for the q test, respectively. These coefficients should be chosen considering the uncertain nature of the observed variables and the different magnitudes of the measured variables. It is important to have a special attention when $W_q^{\mathrm{abs}} = 0$ and $W_q^{\mathrm{rel}} > 0$ for values of $\mathbf{Z}^{\mathrm{exp}}$ near zero because in that case $\mathbf{D}_q \to \infty$ and this will conduct to unfeasible results [28].

Equation 1.2 can be simplified with the approximation of the integral to a finite sum where the difference between the model and the experimental values is only evaluated at the M_q instants of observation, *i.e.* [28]:

$$\mathcal{L}_q(\mathbf{A}) = \frac{1}{M_q} \sum_{i=1}^{M_q} \left[\frac{\mathbf{Z}^{\mathrm{num}}(\mathbf{A}, t_i) - \mathbf{Z}^{\mathrm{exp}}(t_i)}{W_q^{\mathrm{abs}} + W_q^{\mathrm{rel}} \mathbf{Z}^{\mathrm{exp}}(t_i)} \right]^2. \tag{1.4}$$

For this specific case, the optimization problem can be formulated as:

$$\begin{aligned} \min_{\mathbf{A}} \quad & \mathcal{L}(\mathbf{A}) & \\ \text{s.t.:} \quad & g_m(\mathbf{A}) \le 0, & m = 1, \dots, M \\ & h_l(\mathbf{A}) = 0, & l = 1, \dots, L \\ & A_i^{\min} < A_i < A_i^{\max}, & i = 1, \dots, r. \end{aligned} \tag{1.5}$$

In the formulation 1.5, M are the number of inequalities $g_m(\mathbf{A})$ and L are the number of equalities $h_l(\mathbf{A})$ [31]. The search-space is limited and, as a consequence, the optimization parameters should fill in it. In the case that direct search optimization methods are considered, the concern of the variables limits can be done directly in the generation of new parameters. In the case of gradient-base methods this limitations should be considered as constraints for the case of constrained optimization problems or as interior penalties for the case of unconstrained optimization methods. In the case of unconstrained optimization problems the parameters can be transformed as follows [28]:

$$A \geq A_0 \left[1 + (A_{\max} - 1) \left(1 - e^{\frac{1-A}{A_{\max}-1}} \right) \right]. \tag{1.6}$$

In equation 1.6, $A_0 \geq 1$ and A_0 is the initial value of A for an existing $A_{\max}$ and A normalized. Following the same logic, a similar expression can be formulated for $A_0 < 1$ with $A_{\min}$.

Inequalities constraints can also be considered for the case of unconstrained problems. In this case the modified objective function must obey the conditions of continuity and existence of derivative in any point of its universe and for any direction. As a consequence, the problem can be minimizing the following equation [28]:

$$\mathcal{L}^p(\mathbf{A}) = \sum_{q=1}^{N} \mathcal{L}_q(\mathbf{A}) + \sum_{m=1}^{M} [\alpha'_m \max\{0, g_m(\mathbf{A})\}]^2 + \sum_{l=1}^{L} [\alpha'_l h_l(\mathbf{A})]^2, \tag{1.7}$$

where $\alpha \supseteq \{\alpha', \alpha''\}$ are the penalty coefficients that lead to the evolution of the optimization process. The second and third terms of the equation 1.7 are called exterior penalty functions [28].

The other category of inverse problems considered in the present work is the “initial shape optimization”. This class of inverse problems intends to determine the initial geometry/shape of a body when the final geometry or shape is already known. When applied to sheet metal forming processes, its aim would be to estimate the initial shape of a specimen (or a blank) in order to achieve the desired geometry after the forming process. In tool optimization problems, the objective is to find the initial shape of the tool knowing the final shape of the specimen that will be deformed by the tool. It is important not to confuse the initial shape optimization, which is an inverse problem, with the well known “shape optimization problem”. In the case of shape optimization problems, the optimization process tries to find a geometry or shape that is optimal, in the sense that it minimizes a certain objective while satisfying given constraints [23].

The initial shape optimization problem can be similar to the parameter identification problem if the shape to be optimized is defined by a finite number of parameters. Therefore, both problems can be solved by the same approach. The major difference between these two approaches is that, for the constitutive model parameters identification, the variables can have different magnitudes. However, in the case of initial shape optimization problems, all the variables define geometrical shapes and contours, being all with the same magnitude. The objective function for the initial shape optimization problems can be written as equation 1.1 suggests, however, $\mathcal{L}_q(\mathbf{A})$ is now formulated as [28]:

$$\mathcal{L}_q(\mathbf{A}) = \frac{1}{M_q} \sum_{i=1}^{M_q} [F_i(\mathbf{A}, \boldsymbol{\sigma}, \boldsymbol{\varepsilon}, \dots)]^2, \tag{1.8}$$

where F_i is a function that characterizes the shape of the solid and it can be a function of structural properties, such as the stress $\boldsymbol{\sigma}$ and the strain $\boldsymbol{\varepsilon}$ fields. The general formulation of the initial shape optimization problem is similar to the formulation considered in 1.5 for the constitutive model parameters identification.

1.6 FEM and optimization coupling

In general terms, sheet metal forming is a complex deformation process controlled by parameters such as blank shape, tools' geometry, sheet thickness' values, blank holding force, friction, *etc.* [32]**.** Due to its complexity, and the higher combination level of all these input variables, optimization procedures are fundamental tools in the proper design of the process parameters, useful in the prediction and correction of undesirable forming defects such as fracture, springback, wrinkling, shape deviations and unbalanced residual stresses [33]. Due to the high level and robustness of commercial FEM codes, it can be attractive to use FEM simulation results as objective function estimators in a more comprehensive optimization procedure [23].

It was in 1960 that Lucien Schmit recognized the potential for combining optimization techniques and structural design [34]. Since then, shape optimization problems have been intensively studied in the literature. Some works performed in this area are [35-41].

The initial shape design optimization problem is generally solved using a methodology that couples a FEM software with an optimization algorithm. If a commercial FEM software is used, and considering that usually these software codes are *black-box* featured, an interface program can be developed in order to (i) send the information to the FEM software, (ii) execute the FEM program, (iii) retrieve the results, (iv) calculate the objective function and constraints, (v) send the information to the optimization algorithm and (vi) obtain new and improved optimization variables that would be sent again, directly or indirectly, to the FEM software [28,42].

This 6-step typical methodology seems to be straightforward. However, usually FEM software codes do not directly accept optimization variables as input information. These optimization variables can define a geometry (or a shape of a specimen) that must be previously discretized into elements (and nodes) in order to be properly used in the numerical simulation. This task, called *geometry parameterization and discretization*, is not straightforward and can influence the entire optimization process as well as its success [28]. In Figure 1.5, it is possible to see how a FEM program (in the present work the Abaqus® software) and an optimization algorithm can be integrated. As it was mentioned before, an interface algorithm should be developed, which will allow the correct connection between the optimization algorithm and the Abaqus® program. In this case, the initial set of optimization parameters should be firstly written in the *.coe file, which only contains the optimization variables values. The interface will then read this file and write the *.inp file, that is a file with all information needed for the correct simulation of the specific mechanical problem. Then, the FEM program Abaqus® reads the *.inp file and performs the correct simulation. After the simulation, the results are written in the *.dat file. The interface will then compare the results from the *.dat file with an *.exp file that contains only the experimental results. This comparison is performed evaluating the objective function value. The objective function value will be after analysed by the optimization algorithm, which will verify if the stopping criteria are achieved and, in the positive case, all the optimization process stops. In negative case the optimization algorithm will evolve for a new set of parameters.

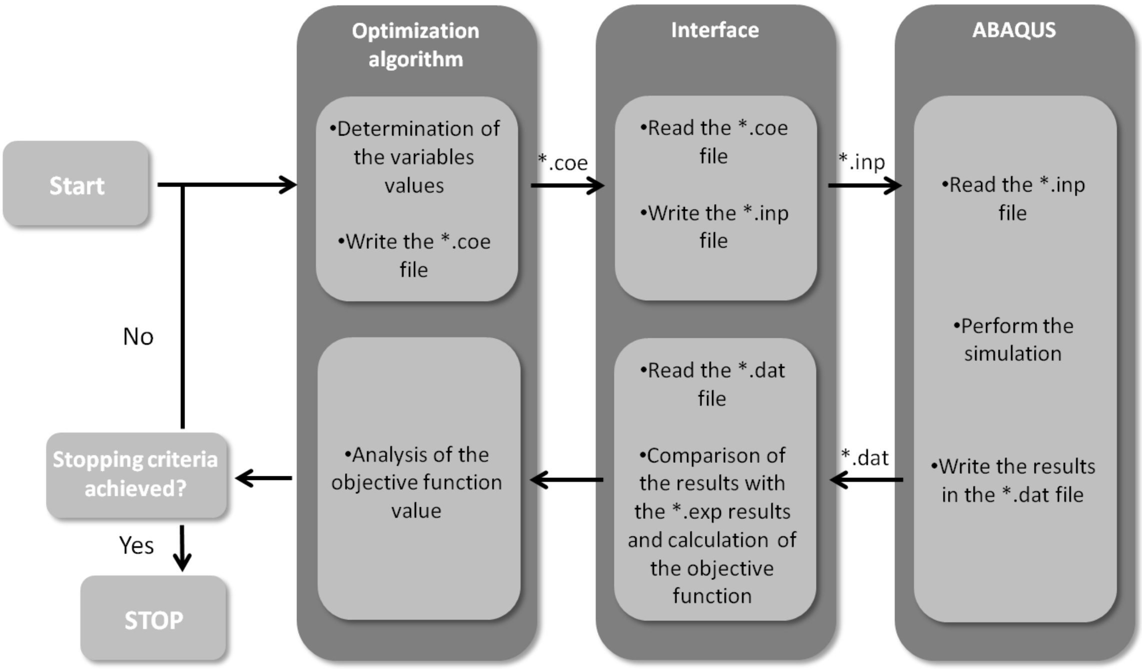

Optimization algorithm
Interface
ABAQUS
Start
•Determination of the variables values
•Write the *.coe file
*.coe
•Read the *.coe file
•Write the *.inp file
*.inp
•Read the *.inp file
•Perform the simulation
•Write the results in the *.dat file
*.dat
•Read the *.dat file
•Comparison of the results with the *.exp results and calculation of the objective function
•Analysis of the objective function value
No
Stopping criteria achieved?
Yes
STOP

Figure 1.5 - Scheme of the FEM and optimization coupling [43], as example.

In the shape optimization performed in the present work the coupling between the optimization program and the FEM software was made as it is explained in Figure 1.5.

1.7 Reading guide

The present work is organized in five main parts. The first part is the *Outline*, which is composed of one generic chapter that gives a background for a better understanding of the proposed work. The second part is the *Mathematical formulation and implementation*. In this part all the fundamental mathematical concepts that the present work comprises are introduced. The third part refers to the *Parameter identification* which presents an experimental material characterization, optimization strategies for non-linear material parameters identification in metal forming processes and a study where the evaluation of the objective function, for parameter identification of material constitutive models, is performed considering a single-point analysis or a FE analysis. In the fourth part, the *Inverse problems of blank and tool design* and methodologies for blank shape design and for tool shape optimization are presented. The part five deals with the *Final Remarks* where some generic conclusions are underlined and discussed, and the future works, where some ideas of how this work can be continued in the future, are presented. The description of each chapter is as follows:

Chapter 1 - An introduction to the presented work is performed. This introduction comprises an historic background, the main objectives as well as the importance of the present work in the current reality. Themes like metal forming processes, Finite Element Method, definition of inverse problems and the coupling between FEM and optimization are also introduced. A reading guide and a list of the publications, published in the scope of this thesis, is provided.

Chapter 2 - Some concepts related to parametric curves and surfaces are introduced. A special focus on the parametric Bézier, B-spline and NURBS curves and surfaces is considered. The main properties of each parameterization are underlined, allowing to understand the advantages and disadvantages of each representation.

Chapter 3 - A review of the main topics in the kinematics of nonlinear continuum mechanics is presented. Concepts such motion, deformation gradient, strain, velocity gradient, rate of deformation and continuum spin are considered. The polar decomposition theorem is described.

Chapter 4 - The fundamentals of constitutive modelling are introduced. The elastic, as well as the plastic behaviour of a material, is formulated. In order to mathematically express a material behaviour, the concepts of yield surface, associated flow rule and hardening law are established. Finally, a summary of the main isotropic and anisotropic yield functions is presented.

Chapter 5 - The Abaqus® user subroutine RSURFU is extensively described, and the formulation presented in chapter 2 is considered for the Bézier and NURBS' curves implementation. The Bézier and NURBS RSURFU are validated for the cases of a linear and a spherical tool.

Chapter 6 - The optimization problem is described and presented. Several one-dimensional and multidimensional unconstrained methods are presented as well as multidimensional constrained problems.

Chapter 7 - A brief introduction underlining the importance of the SDL Optimization Lab is performed. The general structure of the program is explained as well as the problem definition for the correct implementation of optimization problems. Issues like integration methods and the gradient calculation are also presented.

Chapter 8 - The parameter identification problem consists in the comparison between experimental results (coming from mechanical tests) and a constitutive model that characterizes the studied material. Concerning this, in the present chapter, the mechanical characterization of the aluminium alloy AA6082 is presented. This is a generic procedure in the characterization of sheet metal forming materials. Tensile, shear and bulge tests are conducted. The mechanical characterization is performed using the ARAMIS system.

Chapter 9 - Parameter identification inverse problems are studied in order to achieve the best material parameters for specific constitutive models. A non-linear elastic-plastic hardening model, a hyperelastic model, and an elasto-viscoplastic model with isotropic and kinematic work-hardening were considered. Two different optimization algorithms were used: (i) the gradient-based Levenberg-Marquardt algorithm, and (ii) a real search-space evolutionary algorithm (EA). Strategies such as cascade, parallel and hybrid approaches are analysed in detail.

Chapter 10 - In the present chapter two different approaches are presented and analysed: the single-point and FE analysis. The use of these different methodologies for the evaluation of objective function in the parameter identification process is still an open question and the interest in this field has been increasing among the metal forming community. To discuss this issue, two different constitutive models suitable for metals were used, i.e. an non-linear elasto-plastic hardening model and an elasto-viscoplastic model with isotropic and kinematic hardening. The determined material parameters for the two models, the respective objective function values and the CPU time required to perform the simulations are presented and discussed.

Chapter 11 - A numerical procedure for the blank shape design is described and studied. Considering the proposed methodology, the design of a carter blank is presented. The other main objective is the study of the influence of the initial geometry definition in the optimization process. This study is performed considering the NURBS formulation to model the blank upper surface that will be changed during the optimization process.

Chapter 12 - The shape optimization of a two-stage forging process is presented. In order to achieve a straight cylinder after the forging, two different approaches are analyzed. In the first one, the initial geometry of the cylinder is optimized and, in the other one, the shape of the first stage tool is optimized. To parameterize the free surface of the cylinder different methods are presented. Furthermore, in order to define the tool in the last example, also different parameterizations are presented.

Chapter 13 - The main conclusions of the present work are underlined. Some future works in the domain of the present work are presented.

1.8 List of publications in the scope of this thesis

During the work presented in this thesis some publications and oral presentations were conducted. The scientific work performed, in the scope of this thesis, is following described:

International journal papers

- **de-Carvalho R**, Valente R, Andrade-Campos A (2010) On the objective function evaluation in parameter identification of material constitutive models- Single-point or FE analysis. International Journal of Material Forming 3:33-36.

- **de-Carvalho R**, Valente R, Andrade-Campos A (2011) Optimization Strategies for Non-Linear Material Parameters Identification in Metal Forming Problems. Computers and Structures 89:246-255.

- Andrade-Campos A, **de-Carvalho R**, Valente R (2012) Novel criteria for determination of material model parameters. Journal of Mechanical Sciences 54:294-305.

- **de-Carvalho R**, Silva S, Valente R, Andrade-Campos A (2012) Blank optimization in a stamping process – Influence of the geometry definition. Finite Elements in Analysis and Design 61:75-84.

International conference proceeding papers

- **de-Carvalho R**, Valente R, Andrade-Campos A (2009) Optimization Strategies for Non-Linear Material Parameters Identification in Metal Forming Problems. Proceedings of ESMC2009 – 7th EUROMECH Solid Mechanics Conference, Lisboa, Portugal.

- **de-Carvalho R**, Valente R, Andrade-Campos A (2010) On the objective function evaluation in parameter identification of material constitutive models - Single-point or FE analysis. Proceedings of 13th ESAFORM Conference on Material Forming, Brescia, Italy.

- **de-Carvalho R**, Silva S, Valente R, Andrade-Campos A (2011) The Geometry Definition Influence in Inverse Analysis – Application to Carter Forming Process; Proceedings of 14th ESAFORM Conference on Material Forming, Belfast, Ireland.

- **de-Carvalho R**, Andrade-Campos A, Caseiro J, Valente R (2011) On the use of optimization methodologies in sheet metal forming and mechanical characterization. Proceedings of the CMNE 2011, Coimbra, Portugal.

- **de-Carvalho R**, Andrade-Campos A, Valente R (2012) Defining analytical rigid curves/surfaces in tool optimization problems. Proceedings of the 1st ECCOMAS Young Investigators Conference, Aveiro, Portugal.

- **de-Carvalho R**, Andrade-Campos A, Valente R (2012) Defining analytical rigid curves and surfaces in tool optimization problems. Proceedings of the 3rd International Conference on Engineering Optimization, Rio de Janeiro, Brazil.

References

[1] Jeswiet J, Geiger M, Engel U, Kleiner M, Schikorra M, Duflou J, Neugebauer R, Bariani P, Bruschi S (2008) Metal forming progress since 2000. CIRP Journal of Manufacturing Science and Technology 1:2-17.

[2] Woo D (1968) On the complete solution of the deep-drawing problem. International Journal of Mechanical Sciences 10:83-9.

[3] Clough R (1960) The Finite Element Method in plane stress analysis. Proceedings of the 2nd ASCE Conference on Electronic Computation, Pittsburgh, United States of America.

[4] Teixeira-Dias F, Pinho-da-Cruz J, Valente R, Sousa R (2010) Método dos Elementos Finitos – Técnicas de simulação numérica em engenharia. LIDEL, Lisbon (in Portuguese).

[5] Hibbitt H, Marcal P, Rice J (1970) A finite element formulation for problems of large strain and large displacement. International Journal of Solids and Structures 6:1069-1086.

[6] McMeeking R, Rice J (1975) Finite element formulations for problems of large elastic-plastic deformations. International Journal of Solids and Structures 11:601-616.

[7] Bathe K, Ramm E, Wilson E (1975) Finite element formulations for large deformation dynamic analysis. International Journal for Numerical Methods in Engineering 9:353-386.

[8] Banabic D (2010) Sheet metal forming processes - Constitutive modelling and numerical simulation. Springer-Verlag.

[9] Wang N, B Budiansky B (1978) Analysis of sheet metal stamping by finite element method. Journal of Applied Mechanics 45:73-82.

[10] Gotoh M, Ishise F (1978) A finite element analysis of rigid-plastic deformation of the flange in a deep-drawing process based on a fourth-degree yield function. International Journal of Mechanical Sciences 20:423-435.

[11] Wifi A (1976) An incremental complete solution of the stretch-forming and deep-drawing of a circular blank using a hemispherical punch. International Journal of Mechanical Sciences 18: 23-31.

[12] Tang S, Chu E, Samanta S (1982) Finite element prediction of the deformed shape of an automotive body panel during preformed stage. Proceedings of NUMIFORM'82 International Conference, Swansea, U.K..

[13] Toh C, Kobayashi S (1983) Finite element process modelling of sheet metal forming of general shapes. Proceedings of Grundlagen der Umformtechnik I, Berlin, Germany.

[14] Tekkaya A (2000) State-of-the-art of simulation of sheet metal forming. Journal of Materials Processing Technology 103:14-22.

[15] Valberg H (2010) Applied metal forming – Including FEM analysis. Cambridge University Press, New York.

[16] (1974) DIN 8580 Fertigungsverfahren Einteilung (In German).

[17] SCHULER (1998) Metal Forming Handbook. Springer-Verlag.

[18] (1971) DIN 8582, Fertigungsverfahren Umformen (In German).

[19] Altan T, Ngaile G, Shen G (2005) Cold and hot forging – Fundamentals and applications. ASM International®, United States of America.

[20] Bathe K (2006) Finite Element procedures. Prentice Hall, New Jersey.

[21] Liu G, Quek S (2003) The Finite Element Method – A practical course. Elsevier Science Ltd..

[22] de-Carvalho R, Silva S, Valente R, Andrade-Campos A (2011) The geometry definition influence in inverse analysis -Application to carter forming process. Proceedings of the 14th International ESAFORM Conference on Material Forming, Belfast, Ireland.

[23] de-Carvalho R, Silva S, Valente R, Andrade-Campos A (2012) Blank optimization in a stamping process – Influence of the geometry definition. Finite Elements in Analysis and Design 61:75-84.

[24] de-Carvalho R, Valente R, Andrade-Campos A (2011) Optimization strategies for non-Linear material parameters identification in metal forming problems. Computers and Structures 89:246-255.

[25] Ponthot J, Kleinermann J (2006) A cascade optimization methodology for automatic parameter identification and shape/process optimization in metal forming simulation. Computer Methods in Applied Mechanis and Engineering 195:5472-5508.

[26] Andrade-Campos A, Pilvin P, Teixeira-Dias F, Simões J (2007) Determinação dos parâmetros constitutivos através de metodologias de optimização em cascata. Proceedings of Congresso de Métodos Numéricos em Engenharia/Congresso Ibero Latino-Americano sobre Métodos Computacionais em Engenharia, Porto, Portugal.

[27] Andrade-Campos A, Pilvin P, Simões J, Teixeira-Dias F (2009) Software development for inverse determination of constitutive model parameters. Software Engineering: New Research, Nova Science Publishers, Inc..

[28] Andrade-Campos A (2011) Development of an Optimization Framework for Parameter Identification and Shape Optimization Problems in Engineering. International Journal of Manufacturing, Materials, and Mechanical Engineering 1:57-79.

[29] Cailletaud G, Pilvin P (1994) Identification and inverse problems related to material behaviour. Proceedings of the International Seminar on Inverse Problems, Clamart, France.

[30] Andrade-Campos A, Thuiller S, Pilvin P, Teixeira-Dias F (2007) On the determination of material parameters for internal variable thermoelastic-viscoplastic constitutive models. International Journal of Plasticity 23:1349–1379.

[31] Polak E (1997) Optimization – Algorithms and consistent approximations. Springer-Verlag.

[32] Padmanabhan R, Oliveira M, Baptista A, Alves J, Menezes L (2009) Blank design for deep drawn parts using parametric NURBS surfaces. Journal of Materials Processing Technology 209:2402-2411.

[33] Azaouzi M, Belouettara S, Rauchsa G (2011) A numerical method for the optimal blank shape design. Materials & Design 32:756-765.

[34] Belegundu A, Chandrupatla T (1999) Optimization concepts and applications in engineering. Prentice-Hall.

[35] Pironneau O (1983) Optimal shape design for elliptic systems. Springer-Verlag.

[36] Sokolowski J, Zolesio J (1992) Introduction to shape optimization. Springer-Verlag.

[37] Haslinger J, Neitaanmäki P (1996) Finite element approximation for optimal shape, material and topology design. John Wiley & Sons.

[38] Fourment L, Chenot J (1996) Optimal design for non-steady-state metal forming processes-I. Shape optimization method. International Journal for Numerical Methods in Engineering 39:33-50.

[39] Khludnev A, Sokolowski J (1997) Modelling and control in solid mechanics. Springer-Verlag.

[40] Zhou M, Pagaldipti N, Thomas H, Shyy Y (2004) An integrated approach to topology, sizing, and shape optimization. Structural and Multidisciplinary Optimization 26:308–317.

[41] Corriveau G, Guilbault R, Tahan A (2010) Genetic algorithms and finite element coupling for mechanical optimization. Advances in Engineering Software 41:422–426.

[42] Caseiro J (2009) Estratégias Evolucionárias de Optimização de Parâmetros Reais. MSc Thesis. University of Aveiro, Portugal (in Portuguese).

II Mathematical Formulation and Implementation

Chapter 2

Parametric Curves and Surfaces

Some concepts related to parametric curves and surfaces are introduced. A special focus on the parametric Bézier, B-spline and NURBS curves and surfaces is considered. The main properties of each parameterization are underlined, allowing to understand the advantages and disadvantages of each representation.

2.1 Introduction

During the past decades, computer aided design (CAD) tools have played an increasingly important role for a variety of applications in many fields such as: industrial design and manufacture, electrical and mechanical engineering, robotics, computer vision, image processing, computer graphics, biomedicine, *etc*., either for functional or aesthetic reasons [1-3]. It was noted that when different scientific communities were treating very similar problems, they applied different approaches that had lead to different developments in that areas of research. Recently, the increasing complexity of applications leads to the necessity of interconnection between different areas of knowledge to solve these new problems. This interconnection lead to novel technologies in the geometric design field, and simultaneous an intensive research in mathematical theory, representation and analysis of curves and surfaces [2,3].

It is considered that the evolution of CAD had its origins in the work of two French automotive engineers, Pierre Bézier (of Renault) and Paul de Faget de Casteljau (of Citroën) [4]. In his first works [5-7], Bézier used the Bernstein polynomial basis [8] to generate curves and surfaces. The Bézier works were published in 1966, 1967 and 1972. Casteljau, in 1959, had developed similar ideas although these have never been published. The term *spline* was introduced earlier in the mathematical literature by Schoenberg at 1946. His work drew attention to the possibilities of spline approximations and was published in 1966 [9]. In these years, the Coons patch [10] had been putted aside, being the methods of Bézier and Casteljau the methods chosen for CAD representations [4]. During the 1970s, the CAD representation had significantly evolved with the publications of Reisenfeld´s Ph.D. thesis on B-splines in 1972 [11] and the publication in 1975 of the Versprille´s Ph.D. thesis [12] on rational B-splines, which gave rise to the well known NURBS.

Recently, a new class of splines was introduced, the T-splines. They are a generalization of NURBS and allow to describe the same NURBS surface by removing unnecessary control vertices. The T-spline model and the NURBS model are geometrically equivalent but the first one doesn´t consider redundant control vertices. Redundant control vertices can lead also to ripples in the NURBS surface [13].

Regarding the parametric representations, splines are mainly used in the geometric design of curves and surfaces. A spline curve is defined by a set of two or more coordinate positions called control vertices. These control vertices are then the vertices of a piecewise-linear control polygon. In Figure 2.1 it is possible to see the control polygon, with four control vertices, and the final resultant spline.

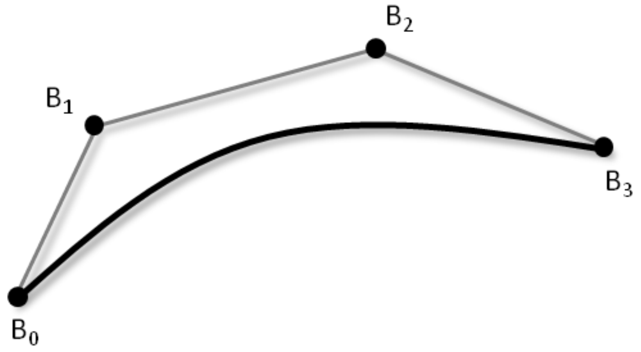


Figure 2.1 - Control vertices in a spline.

Splines can be considered as interpolation splines or as approximation splines if the curve/ surface passes through all the control vertices or if the curve/surface passes near the control vertices, respectively. There are different sort of splines including Cardinal Splines, Kochanec-Bartlet Splines, Bezier Splines, B-Splines and NURBS. Bézier, B-Splines and NURBS curves and surfaces are the most popular and used parametric representations among these.

The nonuniform rational B-spline (NURBS) curves and surfaces are the ones with greater flexibility and precision, that allows an accurate description of complex curves and surfaces. Therefore, NURBS are assumed as the standard in the computer graphics industry and in computer aided design [13]. Chronologically the first ones to appear were the Bézier curves/surfaces, followed by the B-spline curves/surfaces and then the NURBS curves/surfaces, in the search for greater flexibility and precision. B-splines and NURBS curves/surfaces are generalizations of Bézier curves/surfaces and are manipulated in similar ways as Bézier surfaces [1].

The mathematical formulation of these curves/surfaces is based in the so-called *basis functions*. The Bézier surfaces are based on Bernstein basis, which limit their flexibility. This reduction of flexibility happens because the degree of the surface in each parametric direction is one less than the number of control vertices in that direction, and due to the global nature of the Bernstein basis. This means that the change of a control vertex will affect all the shape of the surface. There is another class of basis functions, called the B-spline basis, which contains the Bernstein basis as a special case. These basis are generally nonglobal, which means that each vertex is associated with a unique basis (support) function. As a consequence, each vertex affects the shape of a curve only over a range of parameter values. In the B-spline basis, the maximum possible order of the surface in each parametric direction is equal to the number of control polygon vertices in that direction [1].

The NURBS curves and surfaces formulation is based in the rational B-spline basis functions, which are a generalization of the nonrational B-spline basis functions. These curves and surfaces have more flexibility by the introduction of weights that allows to control how a control vertices influences the whole curve or surface.

In the present chapter only the Bézier, B-spline and NURBS curves and surfaces are considered regarding the main scope of the presented work.

2.2 Parametric curves and surfaces formulation

Curves and surfaces are mathematically represented explicitly, implicitly or parametrically. Explicit surfaces can be written in the form $z = f(x, y)$ and are useful in many applications. However, explicit surfaces are axis dependent, cannot adequately represent multiple-valued functions and cannot be used where a constraint involves an infinite derivative. Implicit surfaces can be written in the form $f(x, y, z) = 0$ and are able to represent multiple-valued functions. However, these are still axis dependent. Parametric surfaces, on the other hand, can be represented as $\boldsymbol{x}(u, w) = [x(u, w), y(u, w), z(u, w)]^{\mathrm{T}}$, with u and w being the function parameters. Therefore, the surface is said to be biparametric. Parametric and implicit surface definitions are predominant in the academic, industrial and commercial fields [1,2].

In the parametric form, each of the coordinates of a point on the curve is represented separately as an explicit function of an independent parameter [15]. Therefore, parametric curves can be represented as [1]:

$$x = f(t);\;\; y = g(t);\;\; z = h(t) \quad \text{with} \quad a \leq t \leq b\,. \tag{2.1}$$

In this equation, t is the independent parameter and although the interval $[a, b]$ is arbitrary, normally it is normalized to $[0,1]$ [14]. Due to its intrinsic formulation, parameter representations are extremely flexible, are axis independent and these have additional degrees of freedom compared to either explicit or implicit formulations [1].

Even very powerful, the parametric form has some disadvantages. In parametric form some typical operations are more difficult, such as: determining the intersection of two parametric curves, finding the distance from a point to a curve or specifying an unbounded geometry [1,14].

Following the same logic, a surface can be parametrically represented as:

$$x = f(u, w);\;\; y = g(u, w);\;\; z = h(u, w), \tag{2.2}$$

where the surface is said to be biparametric, being u and w independent parameters. An isoparametric curve can be formed in the surface if one of the parameters value is held constant while the other is varied. If both parameters are held constant then a point is represented. To form the edges of the surface it is necessary to hold constant the minimum or maximum value of one of the parameters and vary the other parameter [1].

2.3 Bézier curves

2.3.1 Bézier curve definition

Letting $P(t)$ be the position vector along the Bézier curve, a parametric Bézier curve can be mathematically defined as follows [1,14]:

$$P(t) = \sum_{i=0}^{n} B_i \; J_{n,i}(t) \qquad \text{with} \qquad 0 \leq t \leq 1, \tag{2.3}$$

where $\mathbf{B}$ is the position vector of the control vertices, and $J_{n,i}(t)$ is the i^{th} Bernstein basis function with order n, defined as:

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad \text{with} \quad \binom{n}{i} = \frac{n!}{i!\,(n-i)!}\,. \tag{2.4}$$

Considering the equation 2.3, it is possible to formulate the first Bézier curve derivative as:

$$\frac{dP(t)}{dt} = P'(t) = \sum_{i=0}^{n} \mathbf{B}_i \, J'_{n,i}\,(t). \tag{2.5}$$

In an equivalent way, the second derivative can be given as:

$$\frac{d^2P(t)}{d^2t} = P''(t) = \sum_{i=0}^{n} \mathbf{B}_i \, J''_{n,i}\,(t)\,. \tag{2.6}$$

2.3.2 Bézier curve properties

The shape of a parametric curve is determined by a control polygon. This control polygon is a set of control vertices, varying the number of control vertices with the desired degree of the curve. The properties of the Bézier curves are determined by the basis functions properties that formulate the Bézier curves. These functions are known as Bernstein basis. The properties can be summarized as [1]:

- The basis functions are real;
- The degree of the curve is one less than the number of control polygon points;
- The curve usually follows the shape of the control polygon;

- The first and last points on the curve are the same of the first and last points of the control polygon;
- Considering the last point, it is possible to assume that the tangent vectors at the ends of the curve have the same direction as the first and the last polygon spans, respectively.

2.4 B-spline curves

2.4.1 B-spline curve definition

Mathematically, a parametric B-spline curve can be defined as follows [1]:

$$P(t) = \sum_{i=1}^{n+1} \mathbf{B}_i \, N_{i,k}(t) \qquad \text{with} \qquad t_{\min} \leq t \leq t_{\max} \quad \text{and} \quad 2 \leq k \leq n+1. \tag{2.7}$$

In this equation, $\mathbf{B}$ is once more the position vector of the $n+1$ control vertices, and $N_{i,k}(t)$ are the normalized B-spline basis functions of order k and degree $k-1$. The B-spline basis functions are defined by the Cox-de-Boor recursion formula as follows:

$$N_{i,1}(t) = \begin{cases} 1 & \text{if } x_i \leq t < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \text{ and } N_{i,k}(t) = \frac{(t - x_i)\, N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)\, N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}}. \tag{2.8}$$

In this equation, x_i are the elements of a knot vector that satisfies the relation $x_i \leq x_{i+1}$ and t varies from $t_{\min}$ to $t_{\max}$ along the B-spline. Further information about this topic can be found, for instance in references [1,2].

Regarding equation 2.7, it is possible to formulate the first B-spline curve derivative as

$$\frac{dP(t)}{dt} = P'(t) = \sum_{i=1}^{n+1} \mathbf{B}_i \, N'_{i,k}(t), \tag{2.9}$$

and the second derivative as:

$$\frac{d^2P(t)}{d^2t} = P''(t) = \sum_{i=1}^{n+1} \mathbf{B}_i \, N''_{i,k}(t). \tag{2.10}$$

2.4.2 B-spline curve properties

Considering that a B-spline basis is the base of B-spline curve definition, several properties can be underlined, such as [1]:

- The sum of the B-spline basis functions for any parameter value t is 1;
- Each basis function is positive or zero for all parameters values;

- The maximum order of the curve equals the number of control polygon vertices and the maximum degree is less one;
- The curve usually follows the shape of the control polygon;
- The curve is transformed by transforming the control polygon vertices.

2.5 NURBS curves

2.5.1 NURBS curve definition

A non-uniform rational B-spline (NURBS) curve can be mathematically defined as follows [1]:

$$P(t) = \sum_{i=1}^{n+1} \mathbf{B}_i \, R_{i,k}(t), \tag{2.11}$$

where $\mathbf{B}$ is the position vector of the $n + 1$ control vertices, and $R_{i,k}(t)$ are the normalized rational B-spline basis functions of order k and degree $k - 1$. The rational B-spline basis functions are defined as follows:

$$R_{i,k}(t) = \frac{h_i \, N_{i,k}(t)}{\sum_{i=1}^{n+1} h_i \, N_{i,k}(t)} \, . \tag{2.12}$$

In the previous formula, h_i are the weights for each control vertices. In the case all weight are equal to 1 this curve simplifies to a B-spline curve.

Considering the equations 2.11 and 2.12, it is possible to formulate the first rational B-spline curve derivative as:

$$\frac{dP(t)}{dt} = P'(t) = \sum_{i=1}^{n+1} \mathbf{B}_i \, R'_{i,k} \, (t), \tag{2.13}$$

where

$$R'_{i,k} \, (t) = \frac{h_i \, N'_{i,k} \, (t)}{\sum_{i=1}^{n+1} h_i \, N_{i,k}(t)} - \frac{h_i \, N_{i,k} \, (t) \sum_{i=1}^{n+1} h_i \, N'_{i,k} \, (t)}{\left(\sum_{i=1}^{n+1} h_i \, N_{i,k}(t)\right)^2} \, , \tag{2.14}$$

and the second derivative as

$$\frac{d^2P(t)}{d^2t} = P''(t) = \sum_{i=1}^{n+1} \mathbf{B}_i \, R''_{i,k} \, (t), \tag{2.15}$$

where

$$R''^{(t)} = \frac{h_i\, N''_{i,k}\,(t)}{\sum_{i=1}^{n+1} h_i\, N_{i,k}(t)} - \frac{\sum_{i=1}^{n+1} h_i\, N'_{i,k}\,(t)\; h_i\, N'_{i,k}\,(t)}{\left(\sum_{i=1}^{n+1} h_i\, N_{i,k}(t)\right)^2} - \frac{\left(h_i\, N_{i,k}\right)^3 \sum_{i=1}^{n+1} h_i\, N''_{i,k}\,(t)}{\left(\sum_{i=1}^{n+1} h_i\, N_{i,k}(t)\right)^4}$$
$$- \frac{\left(h_i\, N_{i,k}\right)^2 h_i\, N'_{i,k}\,(t) \sum_{i=1}^{n+1} h_i\, N'_{i,k}\,(t)}{\left(\sum_{i=1}^{n+1} h_i\, N_{i,k}(t)\right)^4} + \frac{2h_i\, N_{i,k} \sum_{i=1}^{n+1} h_i\, N'_{i,k}\,(t) h_i\, N'_{i,k}\,(t)}{\left(\sum_{i=1}^{n+1} h_i\, N_{i,k}(t)\right)^3}. \tag{2.16}$$

2.5.2 NURBS curve properties

Since rational B-splines basis functions and curves are a generalization of nonrational B-spline basis functions and curves, their properties are similar to the nonrational B-splines. In particular [1]:

- Each basis function is positive or zero for all parameters values;
- The sum of the rational B-spline basis functions for any parameter value t is 1;
- A rational B-spline curve of order k (degree k-1) is C^{k-2} continuous;
- The maximum order of the rational B-spline curve is equal to the number of control polygon vertices;
- The rational B-spline curve generally follows the shape of the control polygon;
- The rational B-spline curve is transformed by transforming the control polygon vertices.

2.5.3 An example of a NURBS curve

An example of a second degree rational B-spline curve (NURBS) is presented. The control vertices are:

$$B_1 = (0.0,0.0),$$
$$B_2 = (1.0,0.0),$$
$$B_3 = (1.0,1.0),$$
$$B_4 = (2.0,1.0),$$
$$B_5 = (2.0,0.0),$$
$$B_6 = (3.0,0.0),$$

with weights of 1.0, 1.0, 2.0, 2.0, 1.0, 1.0, respectively. In Figure 2.2 it is shown the polygon control vertices and the correspondent NURBS curve. In Figure 2.3 the respective basis functions are also presented.

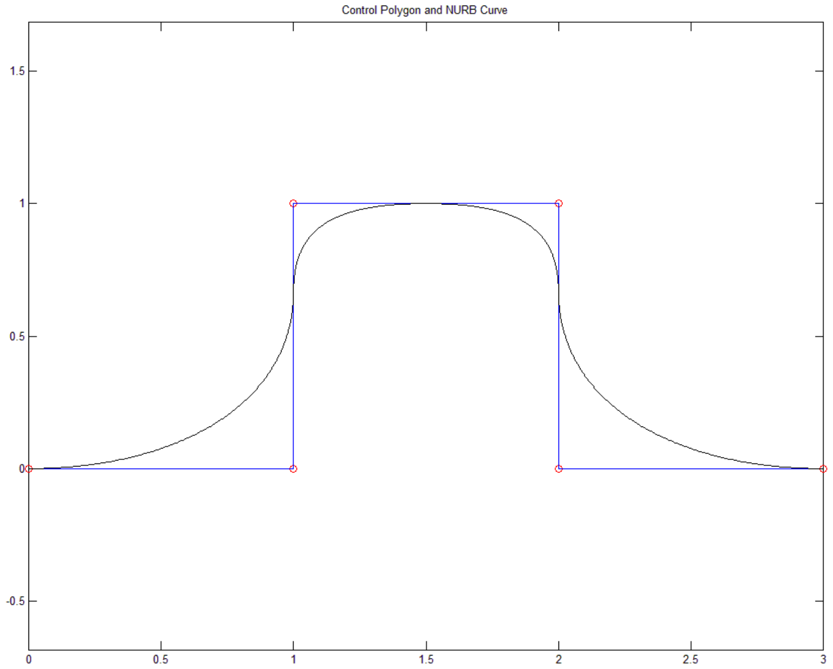


Figure 2.2 - Example of rational B-spline curve.

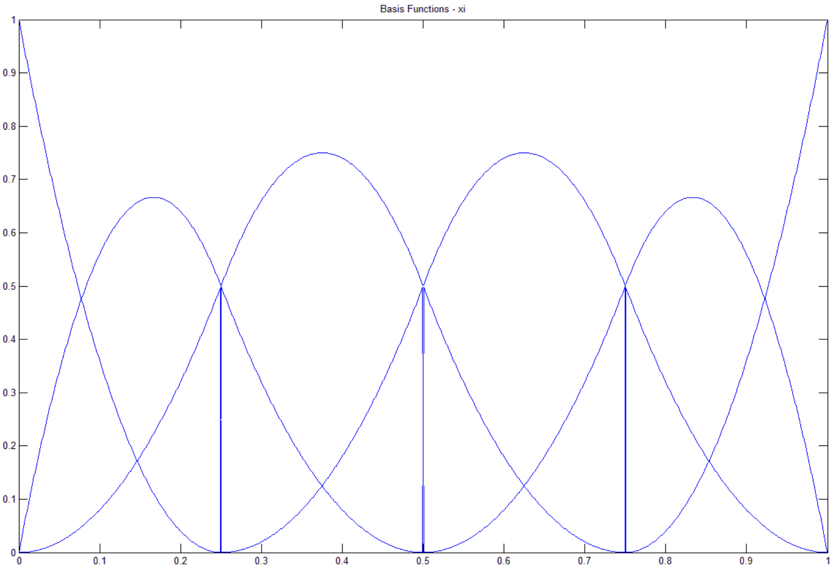


Figure 2.3 - Example of rational B-spline basis functions.

2.6 Bézier surfaces

2.6.1 Bézier surface definition

A Bézier surface can be defined as [1]:

$$Q(u,w) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{B}_{i,j}\, J_{n,i}(u)\, K_{m,j}(w). \tag{2.17}$$

In the equation, $\mathbf{B}_{i,j}$ are the control vertices and the indices n and m are less one than the number of control vertices in the u and w directions, respectively. The $J_{n,i}(u)$ and $K_{m,j}(w)$ are the Bernstein basis functions in the u and w parametric directions, and can be computed as:

$$J_{n,i}(u) = \binom{n}{i} u^{i}(1-u)^{n-i} \quad \text{with } \binom{n}{i} = \frac{n!}{i!\,(n-i)!} \tag{2.18}$$

and

$$K_{m,j}(w) = \binom{m}{j} w^{i}(1-w)^{m-j} \quad \text{with } \binom{m}{j} = \frac{m!}{j!\,(m-j)!}\,. \tag{2.19}$$

Considering equation 2.17, the first and second parametric partial derivatives can be calculated as:

$$\frac{\partial Q}{\partial u} = Q_u(u,w) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,j}\, J'_{n,i}(u)\, K_{m,j}(w)\,; \tag{2.20}$$

$$\frac{\partial Q}{\partial w} = Q_w(u,w) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,j}\, J_{n,i}(u) K'_{m,j}(w)\,; \tag{2.21}$$

$$\frac{\partial^2 Q}{\partial u \partial w} = \frac{\partial^2 Q}{\partial w \partial u} = Q_{uw}(u,w) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,j}\, J'_{n,i}(u)\;\, K'_{m,j}(w); \tag{2.22}$$

$$\frac{\partial^2 Q}{\partial^2 u} = Q_{uu}(u,w) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,j}\, J''_{n,i}(u)\, K_{m,j}(w)\,; \tag{2.23}$$

$$\frac{\partial^2 Q}{\partial^2 w} = Q_{ww}(u,w) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,j}\, J_{n,i}(u) K''_{m,j}(w). \tag{2.24}$$

2.6.2 Bézier surface properties

As it was mentioned for Bézier curves, Bézier surfaces properties are determined by the Bernstein basis functions that formulate the Bézier curves. These properties can be summarized as [1]:

- The basis functions are real;
- The degree of the surface in each parametric direction is less one than the number of control net points in that direction;
- The continuity in each parametric direction is two less than the number of control vertices in that direction;
- The surface usually follows the shape of the control net;
- The corner points of the control net are coincident with the resulting Bézier surface.

2.7 B-spline surfaces

2.7.1 B-spline surface definition

A B-spline surface can be defined as [1]:

$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j}\, N_{i,k}(u)\, M_{j,l}(w)\,, \tag{2.25}$$

where $N_{i,k}(u)$ and $M_{j,l}(w)$ are the B-spline basis functions in the biparametric u and w directions, respectively, and can be calculated as follows:

$$N_{i,1}(u) = \begin{cases} 1 & \text{if } x_i \leq u < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$
$$\text{with } N_{i,k}(u) = \frac{(u - x_i)\, N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)\, N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}, \tag{2.26}$$

and

$$M_{j,1}(w) = \begin{cases} 1 & \text{if } y_j \leq w < y_{j+1} \\ 0 & \text{otherwise} \end{cases}$$
$$\text{with } M_{j,l}(w) = \frac{(w - y_j)\, M_{j,l-1}(w)}{y_{j+l-1} - y_j} + \frac{(y_{j+l} - w)\, M_{j+1,l-1}(w)}{y_{j+l} - y_{j+1}}. \tag{2.27}$$

The B-spline derivatives are given by the expressions:

$$\frac{\partial Q}{\partial u} = Q_u(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j}\, N'_{i,k}(u)\, M_{j,l}(w), \tag{2.28}$$

$$\frac{\partial Q}{\partial w} = Q_w(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j}\, N_{i,k}(u) M'_{j,l}(w), \tag{2.29}$$

$$\frac{\partial^2 Q}{\partial u \partial w} = \frac{\partial^2 Q}{\partial w \partial u} = Q_{uw}(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j}\, N'_{i,k}(u)\; M'_{j,l}(w), \tag{2.30}$$

$$\frac{\partial^2 Q}{\partial^2 u} = Q_{uu}(u,w) = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} B_{i,j}\, N''_{i,k}(u)\, M_{j,l}(w), \tag{2.31}$$

$$\frac{\partial^2 Q}{\partial^2 w} = Q_{ww}(u,w) = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} B_{i,j}\, N_{i,k}(u) M''_{j,l}(w). \tag{2.32}$$

2.7.2 B-spline surface properties

Considering that a B-spline basis is the base of B-spline surface definition, several properties can be underlined, such as [1]:

- The maximum order of the curve in each direction is equal to the number of control vertices in the respective direction;
- The continuity in each parametric direction is less two than the number of control vertices in that direction;
- The surface is transformed by transforming the control net vertices;
- The influence of each control vertex is limited to $\pm{}^{\mathrm{k}}/_{2}$, $\pm{}^{\mathrm{l}}/_{2}$ spans in each parametric direction.
- If the number of control vertices is equal to the order in each parametric direction and there are no interior knot values the B-spline surface reduces to a Bézier surface.

2.8 NURBS surfaces

2.8.1 NURBS surface definition

A NURBS surface can be defined as [1]:

$$P(t) = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} \mathbf{B}_{i,j}\, S_{i,j}(u,w), \tag{2.33}$$

where $\mathbf{B}$ are once more the position vector of the control vertices. $S_{i,j}(u,w)$ are the bivariate rational B-spline basis functions of order k and degree $k-1$. The rational B-spline basis functions are defined as follows:

$$S_{i,j}(u,w) = \frac{h_i\, N_{i,k}(u)\, M_{j,l}(w)}{\sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j}\, N_{i,k}(u)\, M_{j,l}(w)}\,. \tag{2.34}$$

In the previous formula, h_i, as instance, are the weights for each control vertices. Further information about this topic can be found, for instance, in references [1,14].

Considering the equation 2.33 and 2.34, it is possible to formulate the derivatives of the rational B-spline surface as:

$$\frac{\partial Q}{\partial u} = Q_u(u,w) = \frac{\bar{N}}{\bar{D}}\left(\frac{\bar{N}_u}{\bar{N}} - \frac{\bar{D}_u}{\bar{D}}\right), \tag{2.35}$$

$$\frac{\partial Q}{\partial w} = Q_w(u,w) = \frac{\bar{N}}{\bar{D}}\left(\frac{\bar{N}_w}{\bar{N}} - \frac{\bar{D}_w}{\bar{D}}\right), \tag{2.36}$$

$$\frac{\partial^2 Q}{\partial u \partial w} = Q_{uw}(u,w) = \frac{\bar{N}}{\bar{D}}\left(\frac{\bar{N}_{uw}}{\bar{N}} - \frac{\bar{N}_u}{\bar{N}}\frac{\bar{D}_w}{\bar{D}} - \frac{\bar{N}_w}{\bar{N}}\frac{\bar{D}_u}{\bar{D}} + 2\frac{\bar{D}_u}{\bar{D}}\frac{\bar{D}_w}{\bar{D}} - \frac{\bar{D}_{uw}}{\bar{D}}\right), \tag{2.37}$$

$$\frac{\partial^2 Q}{\partial^2 u} = Q_{uu}(u,w) = \frac{\bar{N}}{\bar{D}}\left(\frac{\bar{N}_{uu}}{\bar{N}} - 2\frac{\bar{N}_u}{\bar{N}}\frac{\bar{D}_u}{\bar{D}} + 2\frac{\bar{D}_u^2}{\bar{D}^2} - \frac{\bar{D}_{uu}}{\bar{D}}\right), \tag{2.38}$$

$$\frac{\partial^2 Q}{\partial^2 w} = Q_{ww}(u,w) = \frac{\bar{N}}{\bar{D}}\left(\frac{\bar{N}_{ww}}{\bar{N}} - 2\frac{\bar{N}_w}{\bar{N}}\frac{\bar{D}_w}{\bar{D}} + 2\frac{\bar{D}_w^2}{\bar{D}^2} - \frac{\bar{D}_{ww}}{\bar{D}}\right), \tag{2.39}$$

where

$$\bar{N}_u = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j}\, B_{i,j}N'_{i,k}\,(u)M_{j,l}(w), \tag{2.40}$$

$$\bar{N}_w = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j}\, B_{i,j}N_{i,k}\,(u)M'_{j,l}(w), \tag{2.41}$$

$$\bar{N}_{uw} = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j}\, B_{i,j}N'_{i,k}\,(u)M'_{j,l}(w), \tag{2.42}$$

$$\bar{N}_{uu} = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j}\, B_{i,j}N''_{i,k}\,(u)M_{j,l}(w)\ , \tag{2.43}$$

$$\bar{N}_{ww} = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j}\, B_{i,j}N_{i,k}\,(u)M''_{j,l}(w), \tag{2.44}$$

$$\bar{D}_u = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j}\, N'_{i,k}\,(u)M_{j,l}(w), \tag{2.45}$$

$$\bar{D}_w = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j}\, N_{i,k}\,(u)M'_{j,l}(w), \tag{2.46}$$

$$\bar{D}_{uw} = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j}\, N'_{i,k}\,(u)M'_{j,l}(w), \tag{2.47}$$

$$\bar{D}_{uu} = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j}\, N''_{i,k}\,(u)M_{j,l}(w), \tag{2.48}$$

$$\bar{D}_{ww} = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j}\, N_{i,k}\,(u)M''_{j,l}(w). \tag{2.49}$$

2.8.2 NURBS surface properties

Since rational B-splines basis functions and surfaces are a generalization of nonrational B-spline basis functions and surfaces, their properties are similar to the nonrational B-splines. In particular [1]:

- Each basis function is positive or zero for all parameters values;
- The sum of the rational B-spline basis functions for any parameter value of u and w is 1;
- A rational B-spline surface of order k, l (degree k-1, l-1) is C^{k-2} and C^{l-2} continuous respectively;
- The maximum order of the rational B-spline surface in each parametric direction is equal to the number of control vertices in that direction;
- The rational B-spline surface generally follows the shape of the control net;
- The rational B-spline surface is transformed by transforming the control net vertices;
- The influence of each control vertex is limited to $\pm{}^{\mathrm{k}}/_{2}$, $\pm{}^{\mathrm{l}}/_{2}$ spans in each parametric direction;
- If the number of control vertices is equal to the order in each parametric direction and there are no duplicate interior knot values the NURBS surface reduces to a rational Bézier surface.

The present chapter assumes as fundamental for the mathematical formulation of the curves and surfaces used in the presented work. In Chapter 11 the NURBS surface formulation is considered in the initial blank shape definition. In Chapter 12, the Bézier and NURBS curves formulation is considered in order to define the geometry of a tool used in a forging process.

References

[1] Rogers D (2001) An introduction to NURBS – With historical perspective, Academic Press.

[2] Dimas E, Briassoulis D (1999) 3D geometric modeling based on NURBS: a review. Advances in Engineering Software 30:741–751.

[3] Pottmann H, Leopoldseder S, Hofer M, Steiner T, Wang W (2005) Industrial Geometry: Recent Advances and Applications in CAD. Computer-Aided Design 37:751-766.

[4] Austin J, Hughes T, Bazilevs Y (2009) Isogeometric Analysis – Toward Integration of CAD and FEA. John Wiley & Sons.

[5] Bézier P (1966) Définition numérique des courbes et surfaces I. Automatisme 6:625-632.

[6] Bézier P (1967) Définition numérique des courbes et surfaces II. Automatisme 12:17-21.

[7] Bézier P (1972) Numerical Control: Mathematics and Applications. John Wiley & Sons.

[8] Bernstein S (1912) Démonstration du théorem de Weierstrass fondeé sur le calcul des probabilités. Harkov Soobs. Matem ob-va, 13:1-2.

[9] Curry B, Schoenberg I (1966) On Pólya frequency functions IV: The fundamental spline functions and their limits. Journal d´Analyse Mathematique 17:71-107.

[10] Coons S (1967) Surfaces for computer-aided design of space forms. Project MAC-TR-41 MIT.

[11] Reisenfeld R (1972) Appliccation of B-spline Approximation to Geometric Problems of Computer Aided Design. PhD thesis. Syracuse University.

[12] Versprille K (1975) Computer-aided Design Applications of the Rational B-spline Approximation Form. PhD thesis. Syracuse University.

[13] Ozturk A (2005) State-of-the Art Hardware Based Rendering, Parametric Surfaces and Data Reduction – T-splines.

[14] Piegle L, Tiller W (1997) The Nurbs Book, Springer-Verlag.

Chapter 3

Continuum Kinematics

A review of the main topics in the kinematics of nonlinear continuum mechanics is presented.
Concepts such motion, deformation gradient, strain, velocity gradient, rate of deformation and continuum spin are considered.
The polar decomposition theorem is described.

3.1 Introduction

The main focus of the present work is the parameter identification of constitutive models and the initial shape optimization of forming processes. In both cases the process simulation needs a good description of the phenomena that take place in an infinitesimal amount of material. These phenomena are described considering the kinematics mathematical formulation for continuum mechanics. Considering this fact, in the present chapter a review of the main topics in the kinematics of nonlinear continuum mechanics is performed.

In forming processes, the body typically undergoes large deformations. Deformation usually comprises stretch, rigid body rotation and translation. The stretch contributes for the shape change, while the rigid body rotation and translation does not contribute to shape change or internal stress [1]. Considering a continuum mechanics point of view, the large deformations in solids are analysed considering a referential coupled to the body and that deforms with it [2-4].

This chapter begins with the description of motion and the deformation gradient. Strain field can be defined in many ways in nonlinear continuum mechanics, and some of these definitions are presented, with the polar decomposition theorem being described. Further information about continuum mechanics can be obtained, for instance, in references [5-7].

3.2 The motion's description

To perform the motion's description, and for a better understanding of kinematics, it is necessary to introduce the terms "point" and "particle". The term "point" is used to refer a spatial position, while "particle" refers to a small part of a material continuum. The motion of a particle can be defined considering either a Lagrangian or an Eulerian descriptions [8].

In the Lagrangian (or material) description, the independent variables are the position $\mathbf{X}$ of the particle at time t =0 and the time t. Therefore, the motion can be expressed as [8]:

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t). \tag{3.1}$$

In the Eulerian (or spatial) description, the independent variables are the present time t and the present position $\mathbf{x}$ of the particle that occupied the point $\mathbf{X}$ at time $t = 0$. Therefore, the attention is fixed to a given point instead of a certain particle of a continuum. The motion may be expressed as [8]:

$$\mathbf{X} = \mathbf{X}(\mathbf{x}, t). \tag{3.2}$$

The Eulerian or spatial description is best suited for fluid mechanics problems, as it focuses attention on a certain region in space, which enables the analysis of a flow in a considered point. Solid and structural mechanics problems usually adopt the Lagrangian formulation, since at some stage of a formulation the constitutive behaviour of the material particle must be taken into account, which involves a material description [5,8]. Considering this, in the following a particular focus will be put on the Lagrangian approach, being this one of the most suitable for the forming processes simulation.

3.3 Deformation gradient

In the description of the motion of a body (and deformation), it is necessary to have a configuration which the equations are referred to, known as the reference configuration. Considering Figure 3.1, it is possible to observe the initial undeformed configuration Ω_0 and the current deformed configuration Ω_d of an element of material [6].

It is considered that the element of material Ω undergoes combined stretch, rigid body rotation and translation. The deformation gradient $\mathbf{F}$ is the key quantity in finite deformation analysis, appearing in all equations relating quantities before deformation to corresponding quantities after or during deformation [6]. The deformation vector $\mathbf{F}$ maps the infinitesimal vector $\mathrm{d}\mathbf{X}$ into its deformed state $\mathrm{d}\mathbf{x}$ as follows [1]:

$$\mathrm{d}\mathbf{X} = \mathbf{F}\mathrm{d}\mathbf{x}. \tag{3.3}$$

The relation 3.3 can be written, in the component form, as [8]:

$$\begin{pmatrix} \mathrm{d}x \\ \mathrm{d}y \\ \mathrm{d}z \end{pmatrix} = \begin{pmatrix} F_{xx} & F_{xy} & F_{xz} \\ F_{yx} & F_{yy} & F_{yz} \\ F_{zx} & F_{zy} & F_{zz} \end{pmatrix} \begin{pmatrix} \mathrm{d}X \\ \mathrm{d}Y \\ \mathrm{d}Z \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathrm{x}}{\partial \mathrm{X}} & \frac{\partial \mathrm{x}}{\partial \mathrm{Y}} & \frac{\partial \mathrm{x}}{\partial \mathrm{Z}} \\ \frac{\partial \mathrm{y}}{\partial \mathrm{X}} & \frac{\partial \mathrm{y}}{\partial \mathrm{Y}} & \frac{\partial \mathrm{y}}{\partial \mathrm{Z}} \\ \frac{\partial \mathrm{z}}{\partial \mathrm{X}} & \frac{\partial \mathrm{z}}{\partial \mathrm{Y}} & \frac{\partial \mathrm{z}}{\partial \mathrm{Z}} \end{pmatrix} \begin{pmatrix} \mathrm{d}X \\ \mathrm{d}Y \\ \mathrm{d}Z \end{pmatrix}. \tag{3.4}$$

From equation 3.4, it is possible to define the deformation gradient as [8]:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \cdot \tag{3.5}$$

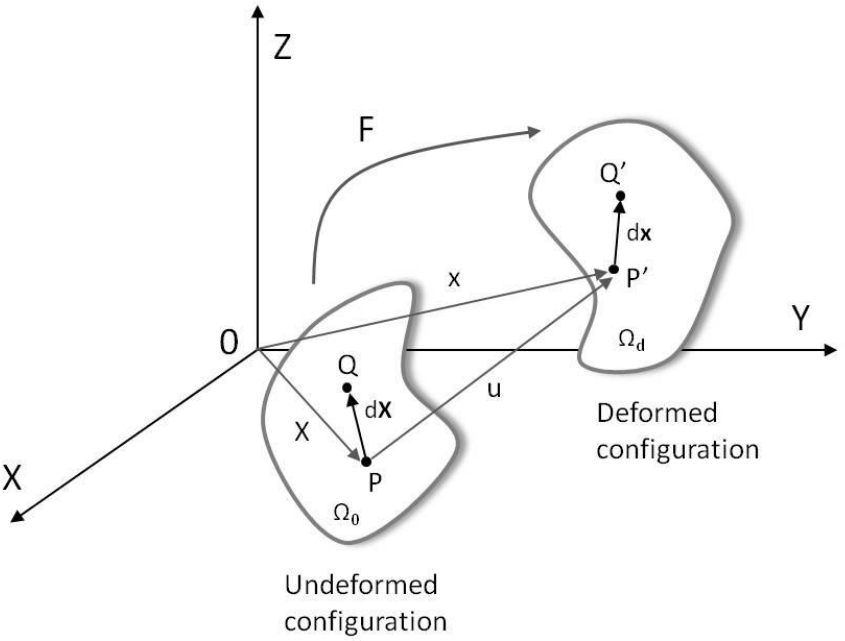


Figure 3.1 - An element of material represented in the undeformed and deformed configuration [8].

By means of the deformation gradient, a complete description of deformation (excluding rigid body translations) is possible, including stretch as well as rigid body rotation. Pure translation is not considered because does not change any vector or its components. In the case of pure translation the deformation gradient is equal to identity. The shape, size change or internal stresses of the element of material Ω are not influenced by the rigid body rotation. Therefore, in solving problems it is necessary to separate the stretch from the rigid body rotation contained within the deformation gradient [1,8].

An useful form of the strain tensor is obtained when it is formulated as a function of displacement gradients. To do so, it is first necessary to represent the deformation gradient as a function of displacement gradients. To this end, the displacement can be defined as:

$$\mathbf{x} = \mathbf{X} + \mathbf{u}, \tag{3.6}$$

and then the deformation gradient can be written as:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}}, \tag{3.7}$$

Where $\mathbf{I}$ is the second order identity tensor.

3.4 Strain measures

In order to determine the different strain tensors, it is necessary to first define the left Cauchy-Green tensor $\mathbf{C}$ and the right Cauchy-Green tensor $\mathbf{B}$. The left Cauchy-Green tensor can be in the form [8]

$$\mathbf{C} = \mathbf{F}^T\mathbf{F}, \tag{3.8}$$

and the right Cauchy-Green tensor as

$$\mathbf{B} = \mathbf{F}\mathbf{F}^T. \tag{3.9}$$

With these auxiliary tensors, different formulations can be used to express the strain tensors. For instance, the Eulerian(or Almansi) strain tensor can be defined as [8]:

$$\boldsymbol{e} = \frac{1}{2}(\mathbf{I} - \mathbf{B}^{-1}). \tag{3.10}$$

Another measure of strain is the logarithmic (or true) strain, that can be defined as [8]:

$$\boldsymbol{\epsilon} = \frac{1}{2}\ln \mathbf{B}^{-\mathbf{1}}. \tag{3.11}$$

Finally, the material Green-Lagrange strain tensor is defined as [8]:

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I}) = \frac{1}{2}\left(\frac{\partial \mathbf{u}}{\partial \mathbf{X}} + \left(\frac{\partial \mathbf{u}}{\partial \mathbf{X}}\right)^{\mathrm{T}} + \left(\frac{\partial \mathbf{u}}{\partial \mathbf{X}}\right)^{\mathrm{T}} \frac{\partial \mathbf{u}}{\partial \mathbf{X}}\right) \cdot \tag{3.12}$$

3.5 Velocity gradient, rate of deformation and continuum spin

Plasticity is an incremental process, and to be implemented in a finite element code plasticity models are written in rate form. In this context, it is possible to define the increment of velocity that occurs over an incremental change in position, $\mathrm{d}\mathbf{x}$, in the form [1]:

$$\mathrm{d}\mathbf{v} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathrm{d}\mathbf{x}. \tag{3.13}$$

The gradient considered in this equation is the velocity gradient, and describes the spatial rate of change of the velocity, in the form [1]:

$$\mathbf{L} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}} = \dot{\mathbf{F}}\mathbf{F}^{-1}. \tag{3.14}$$

The velocity gradient can be decomposed into a symmetric (stretch-related) and an antisymmetric (rotation-related) parts, as follows:

$$\mathbf{L} = \mathbf{D} + \mathbf{W}. \tag{3.15}$$

The symmetric part $\mathbf{D}$ is called the rate of deformation, and is formulated as:

$$\mathbf{D} = \frac{1}{2}\left(\mathbf{L} + \mathbf{L}^{\mathrm{T}}\right), \tag{3.16}$$

while the antisymmetric part represents the continuum spin, being given by [1]:

$$\mathbf{W} = \frac{1}{2}\left(\mathbf{L} - \mathbf{L}^{\mathrm{T}}\right). \tag{3.17}$$

3.6 Polar decomposition

Considering that a deformation comprises a translation, a rigid body rotation and a stretch, and taking into account the considerations stated before, it is possible to formulate the so-called “polar decomposition theorem”. This theorem states that any non-singular, second-order tensor can be uniquely decomposed into the product of an orthogonal tensor (rotation) and a symmetric tensor (stretch) [5]. The polar decomposition of the deformation gradient can be expressed as follows:

$$\mathbf{F} = \mathbf{VR} \tag{3.18}$$

or, alternatively, as

$$\mathbf{F} = \mathbf{RU}, \tag{3.19}$$

where $\mathbf{V}$ and $\mathbf{U}$ are the left and right symmetric stretch tensors, respectively [6].

In Figure 3.2 it is possible to understand how the polar decomposition acts from the undeformed configuration to the deformed one. The represented axes are attached to the material point P.

3.7 Elastic and plastic deformation gradient decomposition

Considering Figure 3.3, it is possible to understand that if the deformable body have an elasto-plastic behaviour it is important to separate the elastic from the plastic contributions for the deformation gradient.

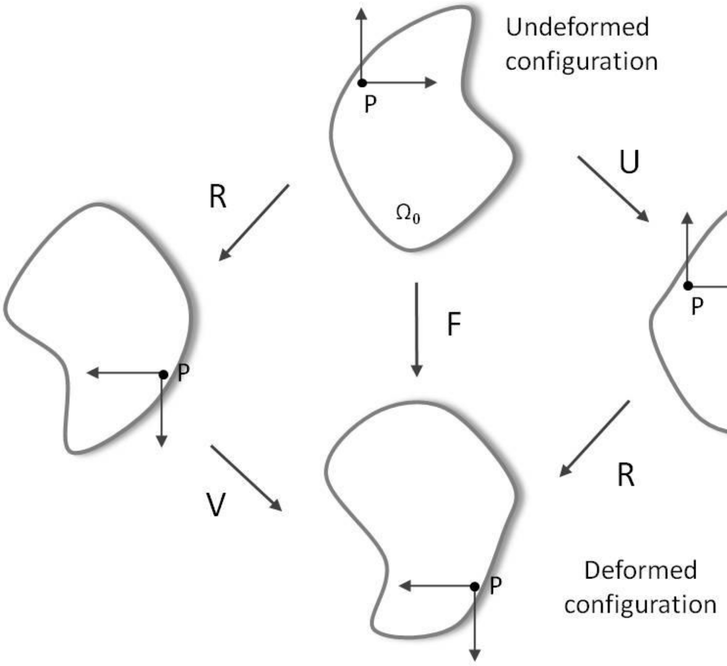


Figure 3.2 - Schematic representation of both methods for the polar decomposition [6].

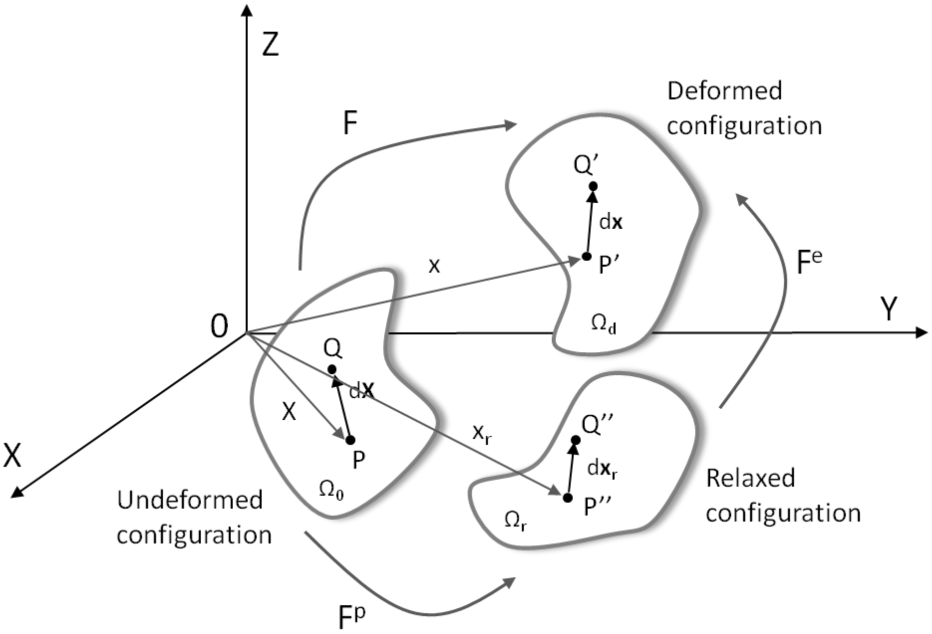


Figure 3.3 - Elastic and plastic decomposition of the deformation gradient [5].

In order to achieve this separation, an intermediate configuration (known as the relaxed configuration) is considered, corresponding t the domain Ω_r in Figure 3.3 [8]. This intermediate configuration is obtained considering that the deformed body has undergone a purely plastic deformation, and that $\mathrm{d}\mathbf{X}$ is now $\mathrm{d}\mathbf{x}_\mathrm{r}$. For each material point belonging to Ω_r a position vector $\mathbf{x}_\mathrm{r}$ is considered [9]. It is now possible to define an elastic and an plastic deformation gradients as follows:

$$\mathbf{F}^{\mathrm{e}} = \frac{\partial \mathbf{x}}{\partial \mathbf{x}_{\mathrm{r}}} \quad \text{and} \quad \mathbf{F}^{\mathrm{p}} = \frac{\partial \mathbf{x}_{\mathrm{r}}}{\partial \mathbf{X}} \cdot \tag{3.20}$$

Finally, the deformation gradient can be expressed as a function of these two tensors, leading to:

$$\mathbf{F} = \mathbf{F}^{\mathrm{e}}\mathbf{F}^{\mathrm{p}}. \tag{3.21}$$

This is known as the classical multiplicative decomposition [8] of the deformation gradient.

References

[1] RodrÍguez-Ferran A, Huerta A (1998) Comparing Two Algorithms to Add Large strains to a Small Strain Finite Element Code. Journal of Engineering Mechanics 124:939-948.

[2] Belytschko T, Liu W, Moran B (2001) Nonlinear Finite Elements for Continua and Structures. John Wiley & Sons.

[3] Banabic D (2010) Sheet Metal Forming Processes – Constitutive Modeling and Numerical Simulation. Springer-Verlag.

[4] Oliveira M (2005) Algoritmos e estratégias de gestão do problema de contacto com atrito em grandes deformações. PhD Thesis. University of Coimbra (in Portuguese).

[5] Andrade-Campos A (2005) Numerical Analysis and Modelling of the Mechanical and Thermal Behaviour of Aluminium Alloys. PhD Thesis. University of Aveiro (in Portuguese).

[6] Simo J, Hughes T (1998) Computational Inelasticity. Springer-Verlag.

[7] Basar Y, Weichert D (2000) Nonlinear Continuum Mechanics of Solids – Fundamental Mathematical and Physical Concepts. Springer-Verlag.

[8] Dunne F, Petrinic N (2005) Introduction to computational plasticity. Oxford University Press.

[9] Bonet J, Wood R (2008) Nonlinear Continuum Mechanics for Finite Element Analysis. Cambridge University Press.

Chapter 4

Constitutive Modeling

The fundamentals of constitutive modelling are introduced. The elastic, as well as the plastic behaviour of a material, is formulated. In order to mathematically express a material behaviour, the concepts of yield surface, associated flow rule and hardening law are established. Finally, a summary of the main isotropic and anisotropic yield functions is presented.

4.1 Introduction

The present chapter has the purpose to make an introduction to constitutive modelling. It is assumed throughout the chapter that only small strains are considered in the absence of large rigid body rotations. As mentioned in chapter 3, after the kinematic transformation have eliminated the rotation effects in the tensor quantities (by the use of corotational framework), the procedures are similar to the ones of small strain formulations.

The material behaviour can be mathematically described by a constitutive equation which gives the stress as a function of the deformation history of the body. Different constitutive relations allow to distinguish different kind of materials [1]. Generally, in metal forming processes, the materials have a linear behaviour at the early stages of deformation. In these cases, it is said that the stress-strain relation is elastic, being given by the Hooke´s law. However, if a certain criterion is reached, it is assumed that the material undergoes irreversible deformations [2]. In this case, it is said that the material had deformed plastically, not being capable of recovering its initial shape. The transition from the elastic to the plastic behaviour is determined by the yield surface or the yield stress value in 1D problems.

The plasticity of materials that exhibit rate effects is known as viscoplasticity. It is considered that viscoplasticity describes rate-dependent plasticity in which crystallographic slip is the dominant deformation process. For the case of viscoplasticity the elastic-plastic strain decomposition still holds, and yield is determined as for the time-independent plasticity with a yield function [3].

In general terms, the yield function can be defined by means of a micromechanical or a phenomenological approach. In the case of the micromechanical approach, the determination of plastic process and the related variables is done at the atomic, molecular or crystalline levels. In the case of the phenomenological (macroscopic) approach, on the other hand, the experimental data is approximated by an analytical function.

It is possible to enumerate some advantages for the use of phenomenological approaches instead the micromechanical ones [4]. Usually, a phenomenological methodology:

- Presents a simpler mathematical form;
- Is computationally easier to be implemented in Finite Element codes;
- Can be generalized to describe anisotropic behaviour of materials.

However, the phenomenological approaches have, as the main disadvantage, a relatively poor accuracy under multi-axial and non-proportional loading conditions [4].

Phenomenological models comprise a yield surface in conjunction to isotropic/kinematic hardening laws, and can be of associate and non-associate types. Other models, known as unified phenomenological models, are also available and, in this case, there is no yield surface [5].

The constitutive formulation presented in the following allows for a better understanding of the material formulation considered in the mechanical problems presented in chapters 9, 10, 11 and 12.

4.2 Linear elasticity

The simplest constitutive model is the linear elasticity. Although the existence of nonlinear elasticity, in the present work only elastic linearity is considered. For certain values of strain, the stress-strain relation is considered to be linear. In this case it is said that if a body is submitted to a specific external solicitation and then this solicitation is removed, it will retrieves its initial shape. In this case the body only had suffered elastic deformations [2]. The stress-strain linear evolution is formulated considering the Hooke´s law, which is represented in the matricial form as [3]:

$$\boldsymbol{\sigma} = \mathbf{D} : \boldsymbol{\varepsilon}^{\mathrm{e}} = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix}. \tag{4.1}$$

In the equation, λ and μ are the Lamé constants given by:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \tag{4.2}$$

and

$$\mu = \frac{E}{2(1+\nu)} \cdot \tag{4.3}$$

4.3 Plasticity

The present work deals mainly with metals, which are made of crystals where atoms are stacked in a regular array. The origin of plasticity in crystalline materials, such as the metals, is related to crystal slip. From this, a number of very important phenomena in macroscopic plasticity become apparent, such as: the plastic slip does not lead to volume change (this is known as the incompressibility condition of plasticity), the plastic slip is a shearing process (*i.e.* the hydrostatic stress can often be assumed not to influence slip), and in a polycristal, the plastic yielding is often considered too be an isotropic process [3].

In Figure 4.1 it is shown the classical stress-strain curve obtained for a purely uniaxial tensile test [3].

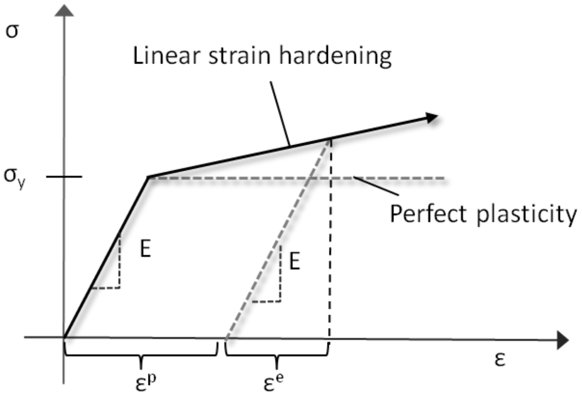


Figure 4.1 - The classical stress-strain representation [3].

As it is possible to observe in Figure 4.1, first the material has a linear elastic behaviour until it reaches the yield stress (σ_y). After this point, the yield stress can increase with higher values of plastic strains, a phenomenon known as strain hardening, since the stress is increasing relative to perfect plastic behaviour [3]. If the material is unloaded, making the applied stress equal to zero, the remaining strain is the plastic strain (ε^p), the recovered strain is the elastic strain (ε^e), and the real strain is the sum of these two components:

$$\varepsilon = \varepsilon^e + \varepsilon^p. \tag{4.4}$$

The plastic behaviour comprises some particular phenomena, such as the strain hardening, the plastic anisotropy, the elastic hysteresis and the Bauschinger effect. To define the plastic behaviour of a material in a general stress state, three elements are required [4]:

- A yield criterion, that express the relation between the stress components at the moment that plastic yielding occurs;
- An associated flow rule, that expresses the relation between the components of the strain-rate and stress-rate;
- An hardening law, that describes the evolution of the initial yield stress during the forming process.

In the following sections these three points are explained in detail.

4.3.1 Yield criterion

As previously mentioned, the material transits from the elastic to the plastic state when the equivalent stress value equals the yield stress coming from the tension test. In the uniaxial case it is easy to find the yield point, however in multiaxial stress state it is more difficult, being necessary to define a criterion that model the transition from the elastic to the plastic state [4]. This criterion defines a relation between the principal stresses, and should specify the conditions under which plastic flow occurs. Such relation is named yield function, and it is normally defined in the form of an implicit function such as [6]:

$$f(\boldsymbol{\sigma}, \boldsymbol{\alpha}) = 0, \tag{4.5}$$

where $\boldsymbol{\sigma}$ is the stress tensor and $\boldsymbol{\alpha}$ is a set of variables that influence the hardening behaviour [7]. The equation 4.5 mathematically represents a surface in three dimensional stress space. This surface must be closed, smooth and convex. All the points located inside the surface ($f < 0$) correspond to elastic stress levels. Points located on the surface ($f = 0$) are related to a plastic state, and the points located outside of the surface ($f > 0$) have no physical meaning [4].

The yield surface function is formulated on the basis of some phenomenological considerations concerned to the transition from the elastic to the plastic state. The yield function may be defined considering two different ways such as: (i) assuming that the plastic yield begins when some physical quantity attains a critical value or (ii) approximating experimental data by an analytical function. The last formulation is purely phenomenological not considering the crystallographic structure of the material.

4.3.2 Associated flow rule

The yield criterion and the hardening law define the yield surface. However, there is no information about the evolution of the plastic deformation [7]. After the yield limit, the material deforms plastically and the normality hypothesis of plasticity enables to determine the “direction” of flow. This hypothesis states that the increment in the plastic strain tensor is in a direction (*i.e.* relative to the principal stress directions) which is normal to the tangent to the yield surface at the load point. This is schematically represented in Figure 4.2 (for the specific case of von Mises yield surface) [3]:

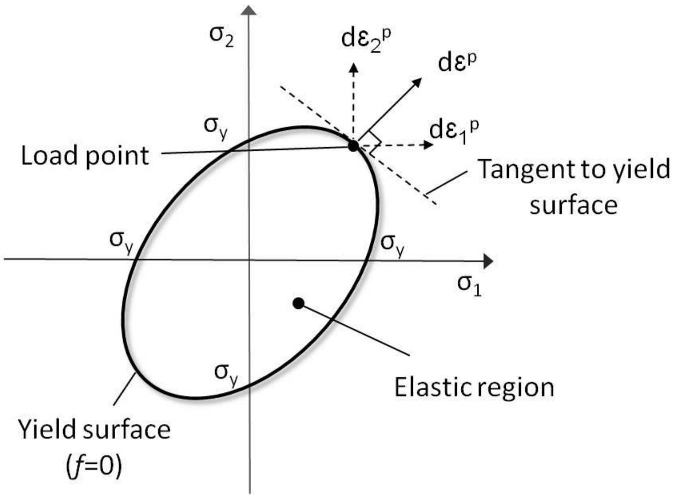


Figure 4.2 - The plastic strain increment for the specific case of von Mises yield surface [3].

The normality hypothesis can be formulated as:

$$\mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{p}} = \mathrm{d}\lambda\frac{\partial f}{\partial \boldsymbol{\sigma}} \qquad \text{or} \qquad \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} = \dot{\lambda}\frac{\partial f}{\partial \boldsymbol{\sigma}}, \tag{4.6}$$

where $\partial f/\partial \boldsymbol{\sigma}$ gives the direction of plastic strain increment (or, equivalently, the plastic strain rate) while the magnitude is determined for the plastic strain rate by $\dot{\lambda}$, that is known as the plastic multiplier.

4.3.3 Hardening rule

Several metals show hardening when deformed plastically. This means that the stress required to cause further plastic deformation increases. This increase is usually a function of the accumulated plastic strain [3]. This function is known as the strain hardening law and represents the evolution of the yield surface. There are three main groups of hardening laws, such as isotropic hardening laws, kinematic hardening laws and combined hardening laws. In the last one, isotropic and kinematic hardening occurs [6]. If there is no hardening after yielding, for further plastic deformation, the yield surface remains unaltered and the equivalent stress remains equal to σ_{y}, as can be seen in Figure 4.3. In this case it is said that the material is perfectly plastic.

In Figure 4.4, it is represented the yield surface evolution and the respective stress-strain curve in the case of isotropic hardening. It is possible to observe that the yield surface expand uniformly in all directions of the stress space.

In the case of isotropic materials, the yield function can be given as:

$$f(\boldsymbol{\sigma}, \bar{\varepsilon}^{\mathrm{p}}) = \sigma_{\mathrm{e}}(\boldsymbol{\sigma}) - \sigma_{\mathrm{y}}(\bar{\varepsilon}^{\mathrm{p}}) = 0, \tag{4.7}$$

where $\sigma_{\mathrm{e}}(\boldsymbol{\sigma})$ is a convex function of the stress tensor and is known as the yield function. $\sigma_{\mathrm{y}}(\bar{\varepsilon}^{\mathrm{p}})$ is the hardening function, being dependent on the equivalent plastic strain ($\bar{\varepsilon}^{\mathrm{p}}$), and establishes the dimension of the yield surface.

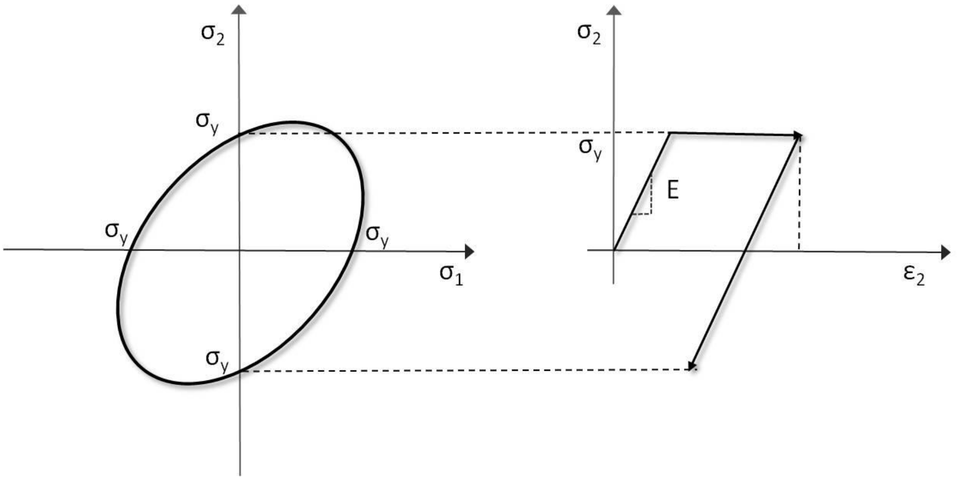


Figure 4.3 - Yield surface and respective stress-strain curve, when no hardening is present [3].

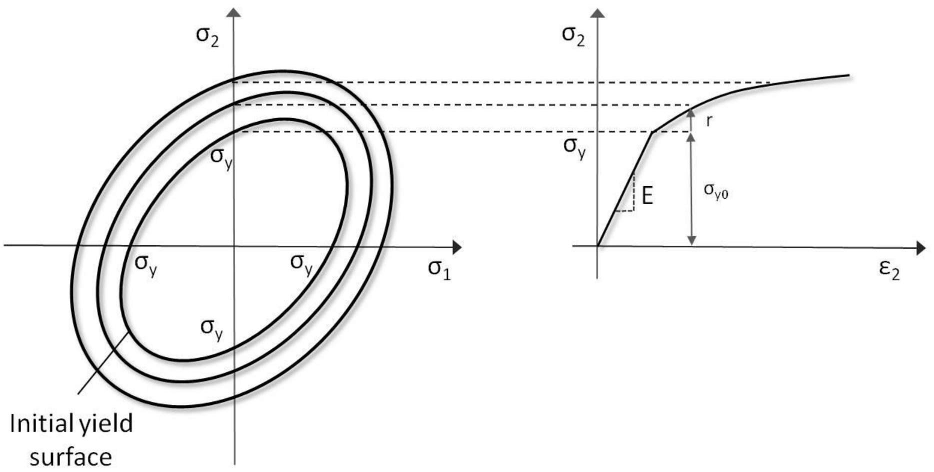


Figure 4.4 - Evolution of the yield surface with isotropic hardening [3].

There are many formulations for the isotropic hardening function. Common examples are the the Ludwick (1909), the Prager (1938), the Hollomon (1944), the Swift (1947), the Voce (1948) and the Fernandes *et al.* (1998) laws. These hardening laws can be formulated as [7]:

Ludwick $$\sigma_{\mathrm{y}} = \sigma_{\mathrm{y}_0} + H\bar{\varepsilon}^n \tag{4.8}$$

Prager $$\sigma_{\mathrm{y}} = \sigma_{\mathrm{y}_0} \tanh\left(\frac{E\bar{\varepsilon}}{\sigma_{\mathrm{y}_0}}\right) \tag{4.9}$$

Hollomon $$\sigma_{\mathrm{y}} = H\bar{\varepsilon}^n \tag{4.10}$$

Swift $$\sigma_{\mathrm{y}} = C(\varepsilon_0 + \bar{\varepsilon}^{\mathrm{p}})^n \tag{4.11}$$

Voce $$\sigma_{\mathrm{y}} = \sigma_{y_0} + (\sigma_{\mathrm{y}_{\mathrm{sat}}} - \sigma_{\mathrm{y}_0})\big[1 - \exp(-C_{\mathrm{y}}\bar{\varepsilon}^{\mathrm{p}})\big] \tag{4.12}$$

Fernandes *et al.* $$\sigma_{\mathrm{y}} = C\big[g\big(\varepsilon_0 + \bar{\varepsilon}_{\mathrm{ps}}\big) + h\bar{\varepsilon}^{\mathrm{p}})\big]^n. \tag{4.13}$$

In these equations, σ_y, σ_{y_0}, and $\sigma_{y_{sat}}$ are respectively the flow stress, the initial yield stress and the saturation stress under uniaxial tension. Also $\bar{\varepsilon}$ and $\bar{\varepsilon}^p$ are the logarithmic equivalent stress strain and the equivalent plastic stress respectively. The other parameters are material constants that need to be determined experimentally [7].

It is often assumed that any hardening that occurs is isotropic. However, in the cases that there is reverse loading, isotropic hardening leads to a very large elastic region. This phenomenon doesn´t occurs experimentally, and is called the Bauschinger effect [3]. In this case it is said that the material hardens kinematically. In Figure 4.5 it is represented the kinematic hardening, where the yield surface suffers a translation in the stress space [6].

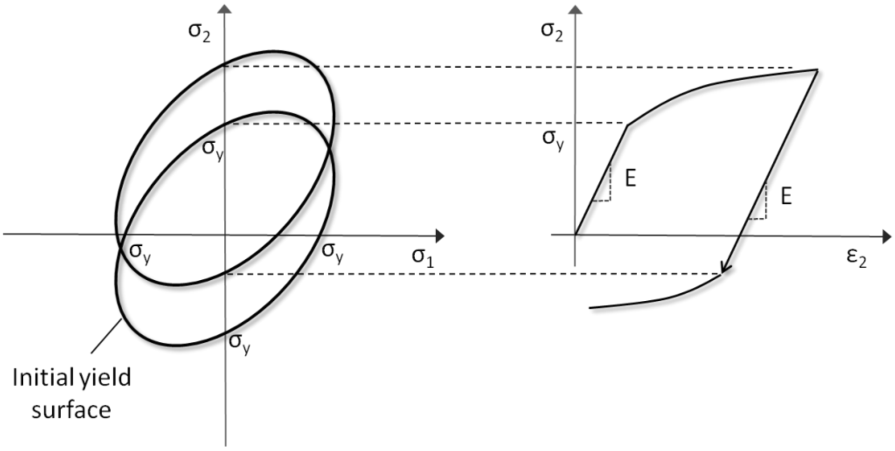


Figure 4.5 - Evolution of the yield surface with kinematic hardening [3].

Considering the yield function in equation 4.7 it is possible to formulate the same function with kinematic hardening as:

$$f(\boldsymbol{\sigma}, \bar{\varepsilon}^p) = \sigma_e\big(\boldsymbol{\sigma} - \boldsymbol{X}(\bar{\varepsilon}^p)\big) - \sigma_{y_0} = 0, \tag{4.14}$$

where $\boldsymbol{X}(\bar{\varepsilon}^p)$ is often called "backstress". This variable is defined in the stress space having the same number of components as the stress tensor [3].

In 1955, Prager [8] proposed the following kinematic hardening law:

$$\mathrm{d}\mathbf{X} = c\mathrm{d}\boldsymbol{\varepsilon}^p, \tag{4.15}$$

where c is a material constant that should be determined experimentally. Considering this law, the yield surface translate in the normal direction. In 1959, Ziegler [9] proposed a modification to the Prager´s law as follows:

$$\mathrm{d}\mathbf{X} = (\boldsymbol{\sigma} - \mathrm{X})\mathrm{d}\mu, \tag{4.16}$$

where μ is a scalar determined with the consistency condition. Instead of moving the yield surface in the normal direction, this law makes the yield surface move in the radial direction defined by the tensor $\boldsymbol{\sigma} - \mathbf{X}$.

The previous hardening law only models linear kinematic hardening. Considering this, Lemaitre and Chaboche [10] had proposed in 1985 a kinematic hardening law that is non-linear with saturation, and formulated as follows:

$$d\mathbf{X} = C_{\mathrm{X}}\left[\frac{X_{\mathrm{sat}}}{\overline{\sigma}}(\mathbf{S}-\mathbf{X})-\mathbf{X}\right]d\bar{\varepsilon}^{p}, \quad \text{with } \mathbf{X}(0)=\mathbf{0}, \tag{4.17}$$

where C_{X} and X_{sat} are material parameter that should again be determined experimentally and $\mathbf{S}$ is the deviatoric stress tensor [7].

Other materials show hardening effects both kinematically and isotropically (see Figure 4.6), which usually happens in cyclic plasticity. In these cases, for an individual cycle the kinematic hardening is the dominant process. However, over a quite large number of cycles the material also hardens isotropically, which is represented by an expansion of the yield surface [3].

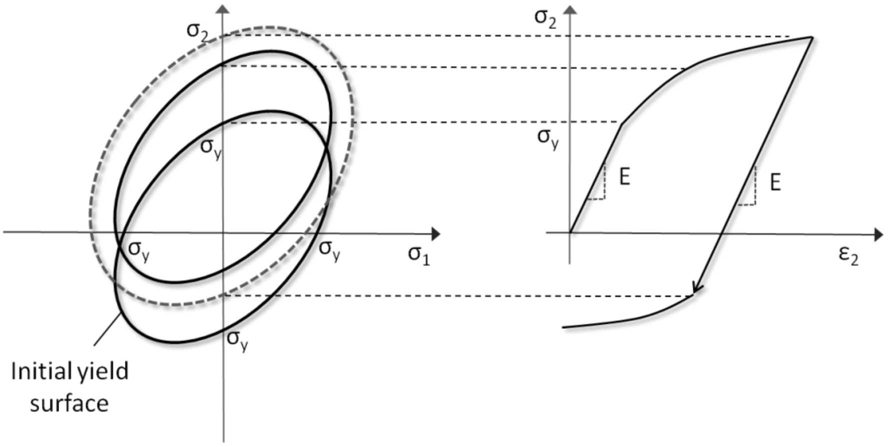


Figure 4.6 - Evolution of the yield surface with combined isotropic and kinematic hardening [3].

In these cases, the yield function of equation 4.6 can be formulated in the form [3]:

$$f(\boldsymbol{\sigma},\bar{\varepsilon}^{\mathrm{p}}) = \sigma_e\big(\boldsymbol{\sigma}-\boldsymbol{X}(\bar{\varepsilon}^{\mathrm{p}})\big)-\sigma_{\mathrm{y0}}+r(\bar{\varepsilon}^{\mathrm{p}}). \tag{4.18}$$

4.4 Yield criteria for isotropic materials

There are two main groups of yield criteria, based on their capability to represent the material anisotropy. One group is named isotropic yield criteria, when under a specific loading the material properties are equal in all directions. If these properties depend on the direction considered, the material is then said to be anisotropic.

The first isotropic yield criterion was proposed by Tresca in 1864 [11], based on the observation that plastic strains appear by crystallographic gliding under shear stresses. This criterion has as main idea that a material passes from elastic to plastic state when the maximum shear stress reaches a critical value [4]. Generically the criterion can be formulated as follows:

$$f(\boldsymbol{\sigma})=\max\{|\sigma_1-\sigma_2|,|\sigma_2-\sigma_3|,|\sigma_3-\sigma_1|\}=\sigma_{\mathrm{y}}. \tag{4.19}$$

In this equation σ_1, σ_2 and σ_3 are the principal stresses and σ_{y} is the uniaxial yield stress from the tension test.

Later on, the Huber-Mises-Hencky isotropic yield criterion was proposed independently by Huber in 1904 [12] and von Mises in 1913 [13], being further developed by Hencky [14]. This criterion is usually known as von Mises yield criterion and is based on the idea that a hydrostatic pressure cannot cause plastic yielding of the material. The criterion can be formulated as follows: the material passes from elastic to plastic state when the accumulated elastic energy of distortion reaches a critical value that is independent of the type of the stress state [4]. The von Mises criterion can be expressed as:

$$f(\boldsymbol{\sigma})=(\sigma_1-\sigma_2)^2+(\sigma_2-\sigma_3)^2+(\sigma_3-\sigma_1)^2=2\sigma_{\mathrm{y}}^2. \tag{4.20}$$

Drucker, in 1949 [15], proposed an isotropic yield criterion in order to represent the experimental data located between the Tresca and von Mises yield surfaces. The criterion can be formulated as:

$$f(\boldsymbol{\sigma})=J_2^3-C_{\mathrm{D}}J_3^2, \tag{4.21}$$

where J_2 and J_3 are the second and third invariants of the stress tensor, respectively, and C_{D} is a constant.

The Hershey [16] isotropic yield criterion is a generalization of these three yield criteria and was proposed in 1954 [4,6]. The criterion can be formulated as:

$$f(\boldsymbol{\sigma})=(\sigma_1-\sigma_2)^a+(\sigma_2-\sigma_3)^a+(\sigma_3-\sigma_1)^a=2{\sigma_{\mathrm{y}}}^a. \tag{4.22}$$

In the previous equation, σ_{y} is the uniaxial yield stress and a is an exponent which is determined based on the crystallographic structure of the material. This criterion reduces to the von Mises criterion if $a=2$, whereas for $a=1$ (and in the limit case that $a\to\infty$) the criterion reduces to the Tresca yield condition [4]. In the case that $2<a<4$, the corresponding surface lies outside the von Mises circular cylinder, whereas for $1<a<2$ and for $a>4$ it lies between Tresca and von Mises.

4.5 Yield criteria for anisotropic materials

In 1984, Hill [17] proposed an anisotropic yield criterion that is a generalization of the Huber-Mises-Hencky isotropic yield criterion. It is one of the most used yield criterion in the material behaviour definition (particularly for steels), mostly because of its simplicity. In the last decades the increasing use of anisotropic alternative materials in metal forming has lead to the development of new, and more complex, yield criteria, that, when integrated in FEM, allow to represent more accurately the material behaviour. In Table 4.1 a resume of the main yield criteria developed as well as the parameters needed for the identification process is presented. The possibility of 3D extension of the yield criterion is also represented. In the table, σ_0, σ_{30}, σ_{45}, σ_{75} and σ_{90}, are respectively the uniaxial yield stresses at 0, 30, 45, 75 and 90° from the rolling direction, while r_0, r_{30}, r_{45}, r_{75} and r_{90}, are respectively the Lankford's coefficients at 0, 30, 45, 75 and 90° from the rolling direction. Also, r_b is the biaxial anisotropy coefficient. The Lankford's coefficients r_θ can be defined as [4]:

$$r_\theta = \frac{\dot{\varepsilon}_{\theta+90}}{\dot{\varepsilon}_{33}}, \tag{4.23}$$

where $\dot{\varepsilon}_{\theta+90}$ is the plastic strain rate associated to the width (inclined at the angle $\theta + 90°$ with respect to the rolling direction) and $\dot{\varepsilon}_{33}$ is the plastic strain rate associated to the thickness.

Nowadays, the most frequently used yield criteria are the Hill 1948, the Hill 1990 and the Barlat 1989 [4]. Considering this fact, the formulation of these three models is presented in detail in the following.

In the Hill 1948 criterion the material is supposed to show anisotropy over three orthogonal symmetry planes. The Hill 1948 yield criterion is expressed by the following quadratic function [4]:

$$2f(\sigma_{ij}) = F(\sigma_{22} - \sigma_{33})^2 + G(\sigma_{33} - \sigma_{11})^2 + H(\sigma_{11} - \sigma_{22})^2 + 2L\sigma_{23}^2 + 2M\sigma_{31}^2 + 2N\sigma_{12}^2 = 1, \tag{4.24}$$

where f is the yield function and F, G, H, L, M and N are constants specific to the anisotropy state of the material. For the case of sheet metals, the axis 1 is usually parallel to the rolling direction, 2 is parallel to the transverse direction and 3 is collinear with the normal direction [4]. The six anisotropy constants in equation 4.24 can be computed as functions of the Lankford's coefficients, as:

$$F = \frac{H}{r_{90}}; \quad G = \frac{1}{r_0 + 1}; \quad H = r_0 G; \quad N = \frac{1}{2}\frac{(r_0 + r_{90})(2r_{45} + 1)}{r_{90}(r_0 + 1)} \cdot \tag{4.25}$$

For the specific case of sheet metal it is not possible to determine experimentally the anisotropic constants L and M. They can be considered equal to N, which corresponds to the situation where the yield stresses in shear are equal for the planes O_{12}, O_{13} and O_{23}. Alternatively, they can be considered equal to 1.5, which corresponds to the isotropic situation [18].

The Hill 1990 yield surface is formulated as follows [4]:

$$\begin{aligned} f(\sigma_{ij}) = {} & |\sigma_{11} + \sigma_{22}|^m + (\sigma_b^m/\tau^m)\left|(\sigma_{11} - \sigma_{22})^2 + 4\sigma_{12}^2\right|^{m/2} \\ & + \left|\sigma_{11}^2 + \sigma_{22}^2 + 2\sigma_{12}^2\right|^{(m/2)-1}\{-2a(\sigma_{11}^2 - \sigma_{22}^2) + b(\sigma_{11} - \sigma_{22})^2\} = (2\sigma_b)^m. \end{aligned} \tag{4.26}$$

In this equation σ_b is the yield stress in equibiaxial tension, τ is the yield stress in pure shear deformation ($\sigma_1 = -\sigma_2$), and a and b are material constants. The value of the exponent m can be achieved by the function [4]:

$$m = \frac{\ln[2(r_{45} + 1)]}{\ln\dfrac{2\sigma_b}{\sigma_{45}}} \cdot \tag{4.27}$$

Table 4.1 - Summary of the main yield criteria and its parameters [4].

Yield criterion	σ_0	σ_{30}	σ_{45}	σ_{75}	σ_{90}	σ_b	r_0	r_{30}	r_{45}	r_{75}	r_{90}	r_b	3D
Hill´s family													
Hill 1948	x						x		x		x		x
Hill 1979	x					x	x						x
Hill 1990	x		x		x	x			x				
Hill 1993	x				x	x	x				x		
Lin, Ding 1996	x				x	x	x		x		x		
Hu 2005					x	x	x		x		x		x
Leacock 2006					x	x	x		x		x		
Hershey´s family													
Hosford 1979	x						x				x		x
Barlat 1989	x						x				x		
Barlat 1991	x		x		x	x							x
Karafillis Boyce 1993	x		x		x		x		x		x		x
Barlat 1997	x		x		x	x	x		x		x		x
BBC 2000	x		x		x	x	x		x		x		x
Barlat 2000	x		x		x	x	x		x		x		
Bron, Besson 2003	x		x		x	x	x		x		x	x	x
Barlat 2004	x		x		x	x	x		x		x	x	x
BBC 2005	x		x		x	x	x		x		x	x	x
Drucker´s family													
Cazacu-Barlat 2001	x	x	x	x	x	x	x	x	x	x	x		x
Cazacu-Barlat 2003	x	x	x	x	x	x	x	x	x	x	x		x
C-P-B 2006	x	x	x	x	x	x	x	x	x	x	x		x
Polinomial criteria													
Comsa 2006	x		x		x	x	x		x		x	x	x
Soare 2007(Poly 4)	x	x	x		x	x	x		x		x	x	x

The constants a and b can be determined as follows [4]:

$$a = \frac{1}{4}\left|\left(\frac{2\sigma_b}{\sigma_{90}}\right)^m - \left(\frac{2\sigma_b}{\sigma_0}\right)^m\right| \tag{4.28}$$

and

$$b = \frac{1}{2}\left[\left(\frac{2\sigma_b}{\sigma_0}\right)^m - \left(\frac{2\sigma_b}{\sigma_{90}}\right)^m\right] - \left(\frac{2\sigma_b}{\sigma_{45}}\right)^m . \tag{4.29}$$

The ratio (σ_b^m/τ^m) can also be formulated as a function of σ_{45}, in the form:

$$\left(\frac{\sigma_b}{\tau}\right)^m = 1 + 2r_{45}. \tag{4.30}$$

The formulation of the Barlat 1989 yield criterion is given by [4]:

$$f(\sigma_{ij}) = a|k_1 + k_2|^M + a|k_1 - k_2|^M + c|2k_2|^M = 2\sigma_e^M, \tag{4.31}$$

where σ_e is the equivalent stress and the coefficients k_1 and k_2 can be calculated as [4]:

$$k_1 = \frac{\sigma_{11} + h\sigma_{22}}{2}; \quad k_2 = \left[\left(\frac{\sigma_{11} - h\sigma_{22}}{2}\right)^2 + p^2\sigma_{12}^2\right]^{1/2}. \tag{4.32}$$

Here a, c, h and p are material parameters identified by [4]:

$$a = 2 - c = \frac{2\left(\frac{\sigma_e}{\tau_{s2}}\right)^M - 2\left(1 + \frac{\sigma_e}{\sigma_{90}}\right)^M}{1 + \left(\frac{\sigma_e}{\sigma_{90}}\right)^M - \left(1 + \frac{\sigma_e}{\sigma_{90}}\right)^M}; \; h = \frac{\sigma_e}{\sigma_{90}}; \; p = \frac{\sigma_e}{\tau_{s1}}\left(\frac{2}{2a + 2^M c}\right)^{1/M}. \tag{4.33}$$

In the previous equation τ_{s1} and τ_{s2} are yield stresses for two different types of shear tests: $\sigma_{12} = \tau_{s1}$ for $\sigma_{11} = \sigma_{22} = 0$ and $\sigma_{12} = 0$ for $\sigma_{22} = -\sigma_{11} = \tau_{s2}$.

4.6 Considerations in the choice of anisotropic yield criteria

As it was previously referred, there are a large number of anisotropic yield criteria in use. In order to select a specific yield criterion, some important factors can be summarized currently [4]:

- The yield criterion should predict accurately the yield locus, the uniaxial yield stress and the uniaxial coefficients of plastic anisotropy;
- It should be computational efficient and easy to implement;
- The yield criterion should be flexible and general;
- It is important the number of mechanical parameters needed for the identification process;
- It should be easy to integrate in an identification process;
- It should be user-friendly and well accepted by the scientific and the industrial communities.

References

[1] Belytschko T, Liu W, Moran B (2001) Nonlinear Finite Elements for Continua and Structures. John Wiley & Sons.

[2] Dodgri I (2000) Mechanics of Deformable Solids – Linear and Nonlinear Analytical and Computational Aspects. Springer-Verlag.

[3] Dunne F, Petrinic N (2005) Introduction to Computational Plasticity. Oxford University Press.

[4] Banabic D (2010) Sheet Metal Forming Processes – Constitutive Modeling and Numerical Simulation. Springer-Verlag.

[5] Andrade-Campos A (2005) Numerical Analysis and Modelling of the Mechanical and Thermal Behaviour of Aluminium Alloys. PhD Thesis. University of Aveiro (in Portuguese).

[6] Teixeira-Dias F, Pinho-da-Cruz J, Valente R, Sousa R (2010) Método dos Elementos Finitos – Técnicas de Simulação Numérica em Engenharia. LIDEL Lisbon (in Portuguese).

[7] T. Grilo (2011) Estudo de Modelos Constitutivos Anisotrópicos Para Chapas Metálicas, MSc Thesis. University of Aveiro (in Portuguese).

[8] W. Prager (1955) The Theory of Plasticity: a Survey of Recent Achievements. Proceedings of The Institution of Mechanical Engineers 169:141-157.

[9] H. Ziegler (1959) A Modification of the Prager´s Hardening Rule. Quarterly Applied Mathematics 17:55-65.

[10] J Lemaître, J. Chaboche (1985) Mechanics of Solids Materials. Cambridge University Press Cambridge.

[11] Tresca H (1864) On the yield of solids at high pressures. Comptes Rendus Academie dês Sciences 59:754 (in French).

[12] Huber M (1904) Przyczynek do podstaw wytorymalosci. Czasopismo Techniczne 22:34–81.

[13] Mises R (1913) Mechanics of solids in plastic state. Göttinger Nachrichten Mathematical Physics 4:582–592 (in German).

[14] Hencky H (1924) On the theory of plastic deformations. Zeitschrift für Angewandte Mathematik und Mechanik 4:323–334 (in German).

[15] Drucker D (1949) Relations of experiments to mathematical theories of plasticity. Journal of Applied Mechanics 16:349–357.

[16] Hershey A (1954) The plasticity of an isotropic aggregate of anisotropic face centered cubic crystals. Journal of Applied Mechanics 21:241–249.

[17] Hill R (1948) A theory of the yielding and plastic flow of anisotropic metals. Proceedings of the Royal Society London 193:281–297.

[18] Oliveira M (2005) Algoritmos e estratégias de gestão do problema de contacto com atrito em grandes deformações. PhD Thesis. University of Coimbra (in Portuguese).

Chapter 5

User Subroutine RSURFU

The Abaqus® user subroutine RSURFU is extensively described, and the formulation presented in chapter 2 is considered for the Bézier and NURBS' curves implementation. The Bézier and NURBS RSURFU are validated for the cases of a linear and a spherical tool.

5.1 Introduction

The present chapter is proposed in order to allow the definition of complex rigid tools, such as the ones considered in the mechanical problem of chapter 12. The geometric definition of tools can be performed by means of analytical functions, parametric surfaces, point clouds or finite elements [1,2]. All these approaches shows advantages and disadvantages. In a comparative way, the main advantages and disadvantages can be described as [2-5]:

- The analytical description simplifies the numeric calculus, however doesn´t allow for the description of tools with complex geometry;
- The parametric definition allows for the description of exact complex tools, resulting in a smoother surface description, which can reduce contact noise;
- The point data process allows a quick simulation, however can´t define some geometric details due to the intrinsic nature of this method;
- The finite element discretization is the most common strategy in use, however the accuracy of the discretization depends on the number of finite elements and, being the

FEM an approximate method in terms of geometry, it is only theoretically exact for an infinite number of elements in the mesh.

It is considered that, when one of the contact bodies is rigid, the use of analytical or parametric discretization improves the solution, since this kind of surfaces presents C^1 continuity, avoiding finite variations of the surface normal [2, 5].

Based on this discussion and considering that only parametric surfaces allow complex discretization, in this work, the parametric tool approach will then be considered.

5.2 RSURFU, an user subroutine to define rigid surfaces

For some applications, the capabilities provided in Abaqus® [3] for defining a rigid surface are too restrictive, not allowing the formulation of complex curves/surfaces. In these cases, a user subroutine RSURFU can be implemented in order to define a more complex analytical rigid surface. This user subroutine is called for each point on the slave surface of a contact pair. The user subroutine should perform the following calculations for each deformable point A [3]:

- A point A’ must be found on the rigid surface (see Figure 5.1), whose normal to the surface passes through A’. However, if there is not a unique point A’, the user subroutine must choose the most suitable point, that usually is the closest point to A;
- The distance H must be calculated, by which A has penetrated into the surface below A’. If H has a negative value, it means that A is outside the surface of the rigid body;
- When the surfaces are in contact, RSURFU must define the local surface geometry.

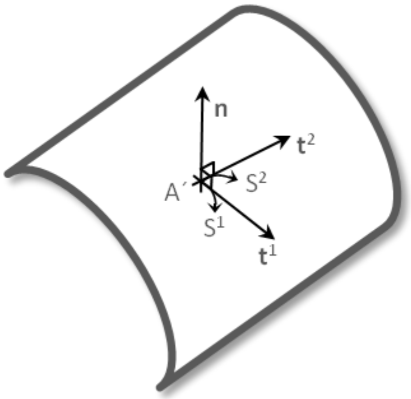


Figure 5.1 - Local geometry on a rigid surface [3].

The local surface geometry mentioned in the last point is specified by two orthogonal tangents in the rigid surface at A’, and the rates of change of the normal at A’ with respect to local surface coordinates S^1 and S^2. These rates of change of the normal vector are required to define the local curvature of the surface. The two tangents to the surface at A’ must be defined so that their positive cross product is the outward normal to the surface [3]. The generic code for the RSURFU implementation is presented in Table 5.1.

Table 5.1 - RSURFU code [3].

```
SUBROUTINE RSURFU(H,P,TGT,DNDS,X,TIME,U,CINAME,SLNAME,
     1 MSNAME,NOEL,NODE,LCLOSE)

     INCLUDE 'ABA_PARAM.INC'

     CHARACTER*80 CINAME,SLNAME,MSNAME

     DIMENSION P(3),TGT(3,2),DNDS(3,2),X(3,3),TIME(2),U(6,2)

        user coding to define H, P, TGT, and DNDS

     RETURN
     END
```

Regarding the correct implementation of the RSURFU user subroutine, it is necessary to define the variables H, $\mathbf{P}$, $\mathbf{TGT}$ and $\mathbf{DNDS}$, which are output variables. Considering a sequential calculation, first it must be determined the position vector $\mathbf{P}$ of the point A' on the surface of the rigid body, where A' should be the closest point to point A on the surface of the deformable body. These calculations will direct or indirectly conduct to the other three variables, as it will be explained in the following.

5.2.1 Position vector of point A´($\mathbf{P}$)

The vector $\mathbf{P}$ is the position of the point A' closest to point A. The determination of the position of point A' is considered the most important and difficult task in the RSURFU implementation. The correct determination will influence all the curve determination as well as the convergence of Abaqus®.

For the determination of the point A', and its coordinates, it was considered that the desired point has a 90 degrees angle between their $\mathbf{t}^1$ tangent and the outward normal, as can be seen in Figure 5.2 [6].

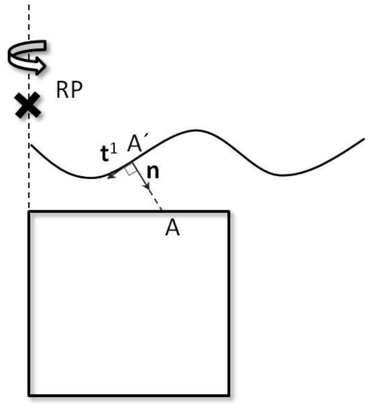


Figure 5.2 - Localization of the first tangent and the normal to the surface [6].

For this case the following relation is valid:

$$\mathbf{t}^1.\mathbf{n} = 0\,. \tag{5.1}$$

Here, the inner product between the first tangent and the normal vector is 0. In the case that more than one point is found, the algorithm calculates the distance between A´ and A and chooses the nearest point [6].

It is important to add the dislocation of the tool (y_r) to the *P*(2) $P(2)$ value, to give motion to the rigid body as follows [6]:

$$\begin{gathered} P(1) = P(t)_x\,, \\ P(2) = P(t)_y + y_r\,, \\ P(3) = P(t)_z\,, \end{gathered} \tag{5.2}$$

where $\mathrm{P(t)_x}$, $\mathrm{P(t)_y}$ and $\mathrm{P(t)_z}$ are the coordinates of the cubic Bézier/NURBS curve obtained by the equations 2.3 and 2.11, respectively.

5.2.2 Direction cosine matrix (TGT)

$\mathbf{TGT}$ is the direction cosines matrix of the two unit tangents, $\mathbf{t}^1$ and $\mathbf{t}^2$, to the surface at point A′. For the two-dimensional cases, only the first two components of $\mathbf{t}^1$ are needed to be given since in this case Abaqus®/Standard assumes that $\mathbf{t}^2$ is equal to (0, 0, −1) and this is the case of the present work. After finding the point A´, it is possible to calculate the tangents to the surface at this point. To find the $\mathbf{TGT}$ values it is only needed normalize the tangent values and take this values as negative due to the intrinsic nature of this problem. This calculation is performed as follows [6]:

$$TGT(1{,}1) = -\frac{P'(t)_x}{\sqrt{{P'(t)_x}^2 + {P'(t)_y}^2}}, \tag{5.3}$$

$$TGT(2{,}1) = -\frac{P'(t)_y}{\sqrt{{P'(t)_x}^2 + {P'(t)_y}^2}}, \tag{5.4}$$

where $P'(\mathrm{t})_\mathrm{x}$ and $P'(\mathrm{t})_\mathrm{y}$ are the x and y components of the first derivative of the cubic Bézier/NURBS curve, depending on the parametric definition considered.

5.2.3 Matrix of rates of change of the surface normal (DNDS)

$\mathbf{DNDS}$ is the matrix with the rates of change of the surface normal, $\mathbf{n}$, at A′, with respect to the distance measuring coordinates S^1 and S^2, along $\mathbf{t}^1$ and $\mathbf{t}^2$. For the two-dimensional case only the first two entries in the first column of $\mathbf{DNDS}$ are required [3]. To perform this calculation, the Frenet-Serret formulation was applied. The *Frenet* formulas are valid exclusively for unit-speed curves. However, the speed v of the curve can be considered as the correlation factor to adapt the *Frenet* formulation for the formulation applied to regular not unit-speed curves, that is the present case [7,8].

In this case, if $\boldsymbol{\alpha}$ is a regular curve in $\mathbb{R}^3$ with $k > 0$, then [7]:

$$\mathbf{n}' = -\mathrm{kv}\mathbf{T} + \tau\mathrm{v}\mathbf{B}, \tag{5.5}$$

where k is the curvature function, τ is the torsion function, $\mathbf{T}$ is the unit tangent, $\mathbf{n}$ is the principal normal and $\mathbf{B}$ is the Binormal.

The RSURFU user subroutine requires the calculation of the $\frac{d\mathbf{n}}{ds}$, which can be defined as follows:

$$\mathbf{n}' = \frac{\mathrm{d}\mathbf{n}}{\mathrm{d}t} = \frac{\mathrm{d}\mathbf{n}}{\mathrm{d}s} \cdot \frac{\mathrm{ds}}{\mathrm{d}t}, \tag{5.6}$$

and $v = \frac{\mathrm{ds}}{\mathrm{d}t}$, which leads to

$$\frac{\mathrm{d}\mathbf{n}}{\mathrm{d}s} = \frac{\mathbf{n}'}{v} = -k\mathbf{T} + \tau\mathbf{B}. \tag{5.7}$$

Considering $\boldsymbol{\alpha}$ a regular curve in $\mathbb{R}^3$ with speed v, it is possible to compute the following variables [8]:

$$\begin{gathered}
\boldsymbol{\alpha}' = \frac{d\boldsymbol{\alpha}}{dt}, \\
\mathbf{T} = \frac{\boldsymbol{\alpha}'}{\|\boldsymbol{\alpha}'\|}, \\
\mathbf{B} = \frac{\boldsymbol{\alpha}' \times \boldsymbol{\alpha}''}{\|\boldsymbol{\alpha}' \times \boldsymbol{\alpha}''\|}, \\
k = \frac{\|\boldsymbol{\alpha}' \times \boldsymbol{\alpha}''\|}{\|\boldsymbol{\alpha}'\|^3}, \\
\tau = \frac{(\boldsymbol{\alpha}' \times \boldsymbol{\alpha}'') \cdot \boldsymbol{\alpha}'''}{\|\boldsymbol{\alpha}' \times \boldsymbol{\alpha}''\|^2}, \\
v = \|\boldsymbol{\alpha}'\|,
\end{gathered} \tag{5.8}$$

In the present problem, however, there is no torsion of the curve, which simplifies to $\frac{d\mathbf{n}}{ds} = -k\mathbf{T}$. Further information on this topic can be found in references [7,8]. Thus, this calculation is performed as follows [6]:

$$DNDS(1,1) = -kT_x = \frac{(\alpha'_x \times \alpha''_x) \cdot \alpha'''_x}{\|\alpha'_x \times \alpha''_x\| \cdot \|\alpha'_x\|^3}, \tag{5.9}$$

$$DNDS(2,1) = -kT_y = \frac{(\alpha'_y \times \alpha''_y) \cdot \alpha'''_y}{\|\alpha'_y \times \alpha''_y\| \cdot \|\alpha'_y\|^3}. \tag{5.10}$$

5.2.4 Penetration value (H)

After defining the position of the point A´, it is easy to compute H, which corresponds to the penetration value of the point A into the surface of the rigid body, measured down the outward normal to the rigid surface. This value must be calculated in a way that H must have a negative value when A is outside the rigid surface, and can be computed as [6]:

$$H = \overrightarrow{\mathbf{AA}'}.\,\mathbf{n}, \tag{5.12}$$

with

$$\overrightarrow{\mathbf{AA}'} = \mathbf{A}' - \mathbf{A} = \mathbf{P} - \mathbf{A} = \big(P(1) - A(1), P(2) - A(2), P(3) - A(3)\big). \tag{5.11}$$

5.3 Validation of the RSURFU user subroutine for the Bézier curve implemented

The RSURFU formulation here presented is used in chapter 12 to define two different parameterization of the considered tool, such as a cubic Bézier curve and a cubic NURBS curve. The formulation of these two curves was described in chapter 2. One important step before the use of these RSURFU user subroutines is their validation. In the present section this validation is performed in a compression test. For this test two situations are considered: one where a linear tool compress the specimen and another where a spherical tool compress the specimen. For each situation three approaches are considered. The first approach is called "linear/spherical tool with Bézier RSURFU", in which the tool is formulated considering the RSURFU user subroutine and the Bézier/NURBS implementation. Other approach is called "linear/spherical tool with geometric RSURFU", in which the tool is formulated considering geometric relations implemented in the RSURFU user subroutine. The last one is called "linear/spherical tool", in which the tool is formulated without the RSURFU. The obtained results are compared for the three approaches [6].

5.3.1 Description of the compression test

The present example consists in the compression of an axisymmetric cylindrical billet to 60% of its initial height This initial value is equal to 100 mm, and the example is schematically shown in Figure 5.3 [9,10].

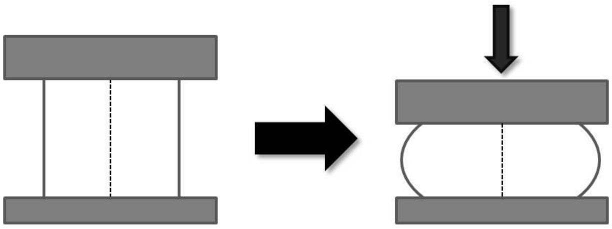


Figure 5.3 - Schematic representation of the compression test, before and after deformation [9].

In this example, the billet is defined as a deformable axysimetric body, while the tools were modeled as rigid bodies. The von Mises elasto-plastic material model was used for the constitutive formulation, considering a linear hardening model. The classical isotropic Coulomb friction law was used to model the contact conditions between the billet and the tool [10]. In order to reduce the computational time, only one-quarter of the cross section was considered, since the deformation is symmetric about the vertical and the horizontal axes. The FEM simulations were performed in the Abaqus® program [3], and the material properties used as input are presented in Table 5.2.

Table 5.2 - Mechanical properties of the billet [10].

Mechanical properties	
Density [kg/m^3]	2710
Young´s module [MPa]	71000
Plastic yield [MPa]	100
Poisson´s ratio [-]	0.33
Friction coefficient [-]	0.1

The studied mesh was based on the work of Grešovnik [10] and the element used was a CAX4R, a 4-node bilinear axisymmetric quadrilateral element with reduced integration and hourglass control [3]. In Figure 5.4 it is possible to observe the mesh used in this work.

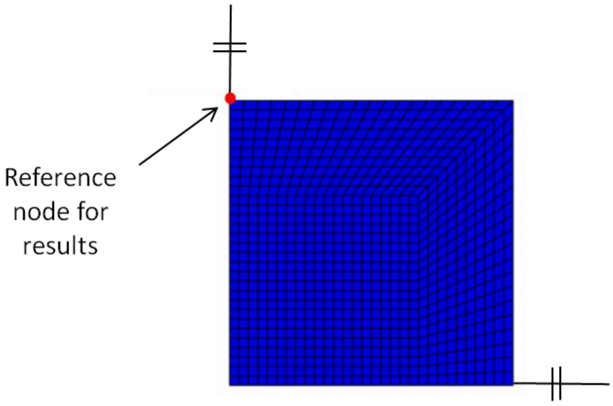


Figure 5.4 - Mesh area considered in the validation test.

5.3.2 Linear tool validation

The first validations are performed considering a linear tool and a spherical punch. For both cases the comparisons are carried out considering the results obtained for the node identified in Figure 5.4. For both cases the initial position of the tool is 5 mm above the billet. In Table 5.3, the code for the implementation of the linear tool with geometric RSURFU is presented [6].

Table 5.3 - RSURFU core code for the linear tool [6].

```
C
C     A(3) is the node position of the deformable body
C
      A(1) = x(1,1)
      A(2) = x(2,1)
      A(3) = x(3,1)
C
C     yr is the position of the reference point that is
C     associated to the tool
C
      yr = x(2,2)

      P(1)= A(1)
      P(2)= yr
      P(3)= A(3)

      TGT(1,1)  =  -1.
      TGT(2,1)  =  0.
      TGT(3,2)  =  -1.

      DNDS(1,1) =  0.
      DNDS(2,1) =  0.

      H = A(2)-P(2)
```

In the Appendix the code that formulates the tool considering a Bézier and a NURBS curve is presented. The linear tool doesn´t use the RSURFU user subroutine in the tool implementation, needing only the *.inp file from Abaqus®. In this case the tool definition can be performed considering the LINE function from Abaqus®. Considering the von Mises equivalent stresses and the billet deformations, the obtained results are the same for the three different approaches, being presented in Figure 5.5. In Table 5.4, the main results, obtained for the three approaches, are shown. The step time number until establish contact is also presented. The $TGT\ (1,1)$, the $TGT(2,1)$, the $DNDS(1,1)$ and the $DNDS(2,1)$ are the values obtained during the calculation using the equations 5.3, 5.4, 5.9 and 5.10, respectively [6].

In Table 5.4 it is also shown the average of the distance between nodes after the forming for the three approaches. Considering the similarity that is shown in both von Mises stresses and the RSURFU internal variables, the Bézier RSURFU user subroutine was considered to be validated for the linear case.

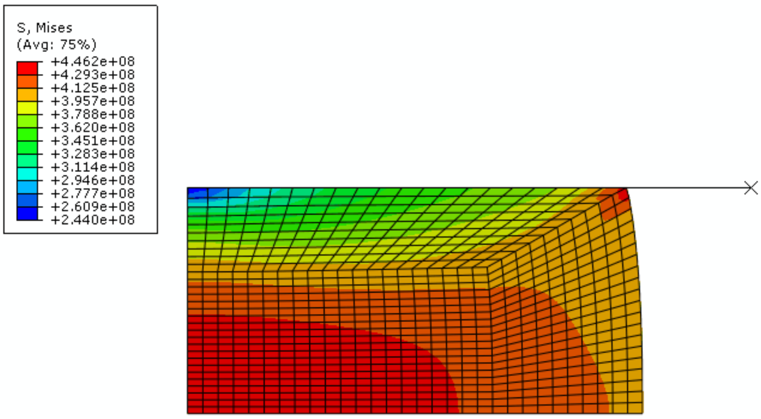


Figure 5.5 - von Mises results for the three different approaches.

Table 5.4 - Results for the linear tool validation.

	Linear tool with Bézier RSURFU	Linear tool with geometric RSURFU	Linear tool
Step time number until the contact is establish	60	60	-
TGT (1,1) [m]	-1.0 ± 0.0	-1.0 ± 0.0	-
TGT (1,2) [m]	0.0 ± 0.0	0.0 ± 0.0	-
$DNDS$ (1,1) [m^{-1}]	0.0 ± 0.0	0.0 ± 0.0	-
$DNDS$ (1,2) [m^{-1}]	0.0 ± 0.0	0.0 ± 0.0	-
Average of the final distance between nodes for the linear tool with Bézier RSURFU and the… [m]	-	0.0 ± 0.0	0.0 ± 0.0
Average of the final distance between nodes for the linear tool with geometric RSURFU and the… [m]	-	-	0.0 ± 0.0

5.3.3 Spherical tool validation

After the linear validation stage, it is still needed to validate the implemented RSURFU with a more complex problem. It is in this context that the following validation was performed. To validate the Bézier RSURFU, the three different approaches considered in the previous section were considered, however in this case a spherical tool was used instead of the previous linear tool. The geometry considered is schematically represented in Figure 5.6.

The "spherical tool with geometric RSURFU" was performed considering the geometric RSURFU given in Abaqus® documentation, with the code being presented in Table 5.5 [3].

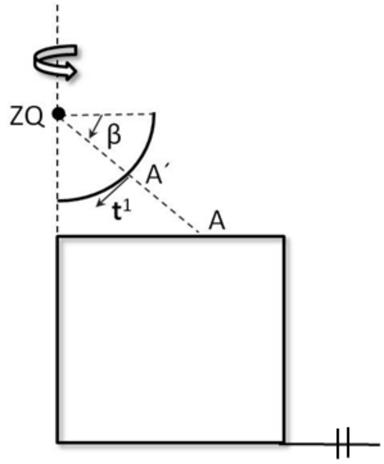


Figure 5.6 - Schematic representation of the modeled (one-quarter) mesh area [6].

Table 5.5 - Spherical punch RSURFU code [3].

```
C     A  = RADIUS 'A' OF THE SPHERICAL HEAD
C     Z0 = ORIGINAL 'Z' COORDINATE OF POINT 'Q'
      A=0.04
      Z0=0.09
      ZQ=Z0 + U(2,2)
C
C     SPHERE
C
        B=SQRT(X(1,1)**2 + (X(2,1)-ZQ)**2)
        H=A-B
        COSB=X(1,1)/B
        SINB=(ZQ-X(2,1))/B
        P(1)=A*COSB
        P(2)=ZQ-A*SINB
        TGT(1,1)=-SINB
        TGT(2,1)=-COSB
        DNDS(1,1)=-SINB/A
        DNDS(2,1)=-COSB/A
```

Considering that it is impossible to draw an analytically exact circle with one Bézier curve, an approximation was therefore considered to simulate the spherical punch with the Bézier RSURFU [10]. To this end, an approximation of a unit quarter of circle (90 degree) by a cubic Bézier curve with an error of 1,96x10^{-4} in the radius [11] was considered. For the case of a quarter of circle (as the one in Figure 5.6) with radius r and centre in (0,0), it is possible to infer that the control points of the cubic Bézier curve are given by the expressions [12]:

$$B_0 = [0, -r],$$

$$B_1 = [r * k, -r],$$
$$B_2 = [r, -r * k],$$
$$B_3 = [r, 0],$$

where k is a constant[1] and is equal to 0.5522847498. More details about this approximation can be found in [12].

The three approaches were implemented and simulated, and in Figure 5.7 the simulation results in terms of von Mises stresses for each one of them are presented. Considering the von Mises equivalent stresses for the three approaches, it can be seen that these are not the same, however they are extremely close. To understand what was going inside the RSURFU, the output variables were compared for the two approaches that need the RSURFU implementation. The point where this analysis was performed is indicated in the Figure 5.4, having an initial value of (0,0.05) [m]. Some of these values are systematized and compared in Table 5.6, as well as in the graphs from Figure 5.8 to Figure 5.10.

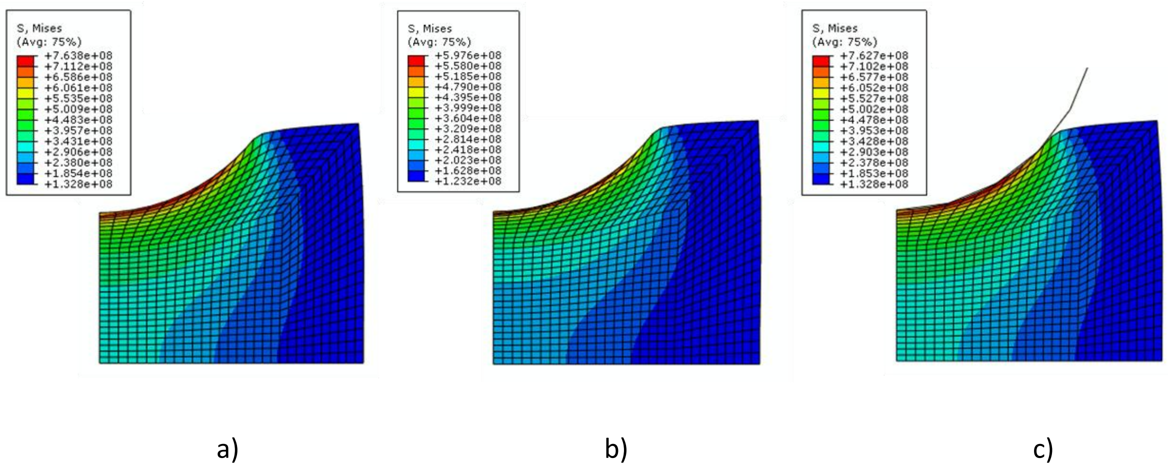


Figure 5.7 - Results for a) spherical tool with Bézier RSURFU, b) spherical tool with geometric RSURFU and c) spherical tool.

As it is possible to observe in Table 5.6, the two different approaches needed the same number of time steps to establish contact. Additionally, the first component of the first derivative is the same for both approaches. However, the second component for the spherical tool with geometric RSURFU approach is not exactly zero, although its value is very low (residual values). It is possible to observe that the first component of the normal derivative is not the same, however it is very close and remains constant in all simulation for both approaches. The same conclusions as the ones considered for the second component of the first tangent can also be applied to the $DNDS(1,2)$.

[1] k in literature is very often called as “magic number”.

Table 5.6 - Results for the spherical tool validation.

	Spherical tool with Bézier RSURFU	Spherical tool with geometric RSURFU	Spherical tool
Time steps number until the contact is establish	44	44	-
TGT (1,1) [m]	-1.0 ± 0.00	-1.0 ± 0.00	-
TGT (1,2) [m]	0.00 ± 0.00	3.39 x10^{-33} ± 1.24 x10^{-31}	-
$DNDS$ (1,1) [m^{-1}]	-24.46 ± 0.0	-25.00 ± 0.0	-
$DNDS$ (1,2) [m^{-1}]	0.00 ± 0.00	8.47 x10^{-32} ± 3.11 x10^{-30}	-
Mean of the final distance between nodes for the spherical tool with Bézier RSURFU and the… [m]	-	4.39x10^{-4}	2.19 x10^{-6}
Mean of the final distance between nodes for the spherical tool with geometric RSURFU and the… [m]	-	-	4.39 x10^{-4}

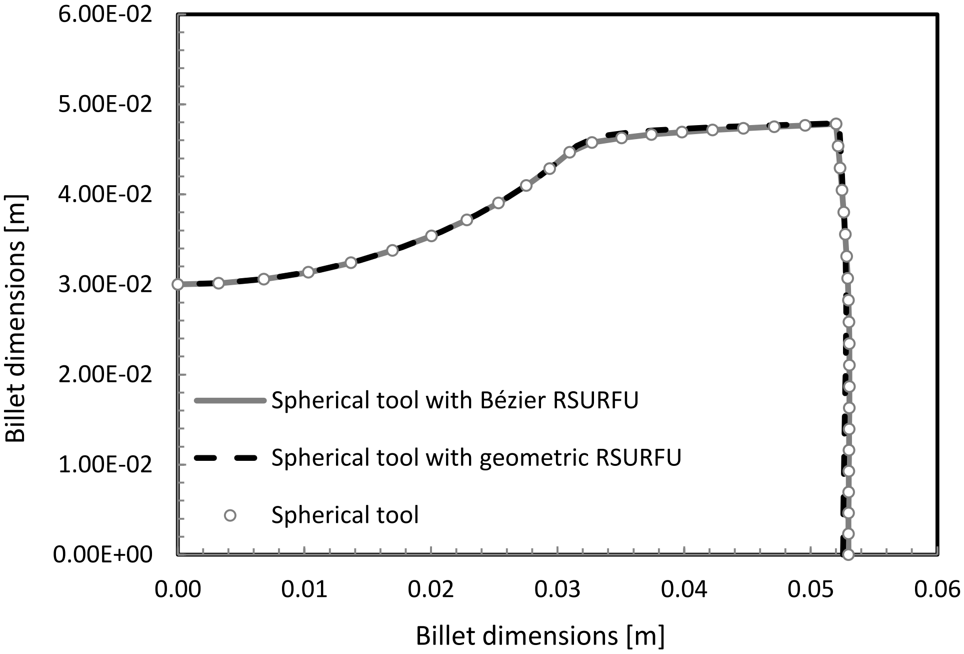


Figure 5.8 - Final shapes of the billet for the three approaches.

The distance between the nodes in the billet surface coming from the three approaches were also calculated (Table 5.6). From the three approaches presented, the two that obtained more similar results, when the difference in the final shape is considered, are the “spherical tool with Bézier

RSURFU" and the "spherical tool approach". Although these two approaches obtained more similar results the three approaches are really similar in a geometric point of view and this fact can be emphasized with the results presented in Figure 5.8.

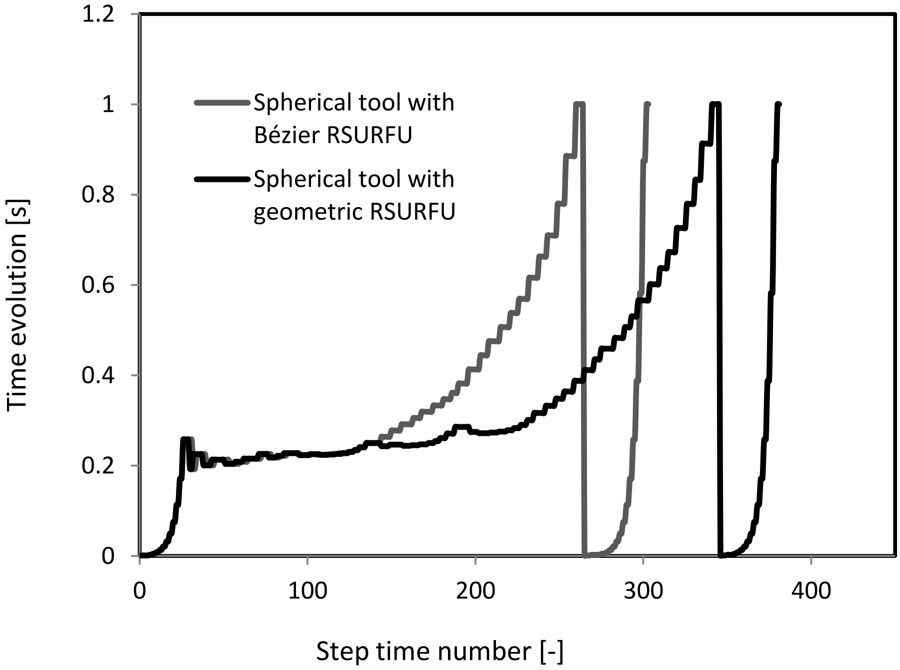


Figure 5.9 - Time evolution with the step time number.

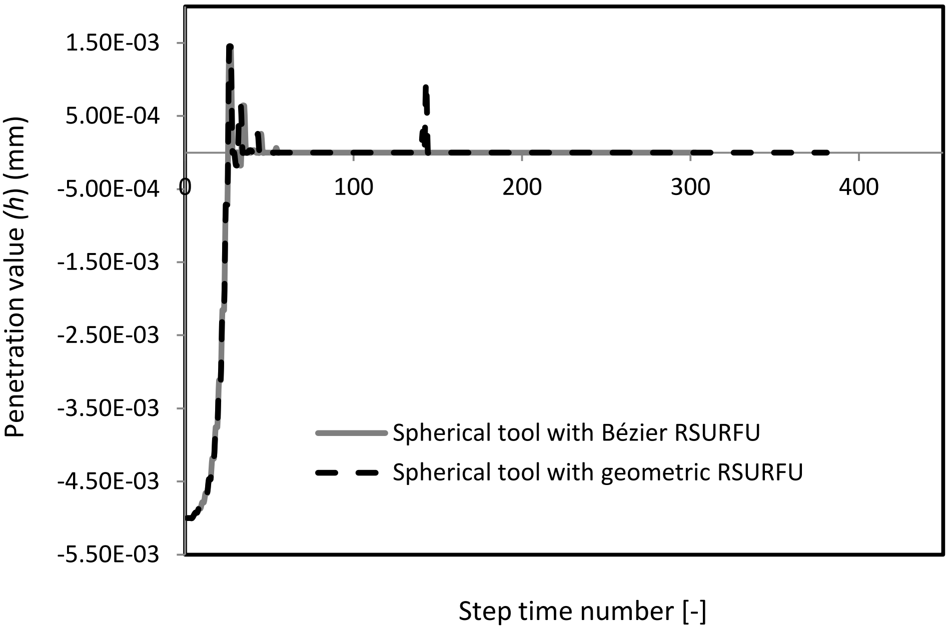


Figure 5.10 - Penetration value in function of the step time number.

Considering the results presented in Table 5.6, it is possible to conclude that the two RSURFU approaches needed the same number of steps to establish contact. This fact can be also observed in Figure 5.10, where the penetration value is zero in iteration 44. However, as it is possible to observe in Figure 5.9, the shape of the time evolution is only similar in the initial part (approximately until iteration 150) of the simulation for both approaches. For the case of the spherical tool formulated with a geometric RSURFU, the contact is lost at approximately iteration 150. This lost of contact is according to the fact that after 150 iterations the time evolution starts to diverge. For both approaches the shape of the time evolution curve is similar but it seems to be out of phase. As it is possible to observe in Figure 5.9, after the contact is established the time evolution with the step time number is the same for both approaches.

Based on the ideas stated before, it is possible to conclude that the approach considering the Bézier RSURFU had achieved better results in terms of convergence. Considering the similarities of both approaches the Bézier RSURFU implemented is considered validated.

5.3.4 Results for the NURBS subroutine

In this section a NURBS subroutine was implemented in the same way that the Bézier subroutine was implemented, varying only the equations of the cubic curve. The NURBS curve was validated considering that if a cubic NURBS with four control vertices have all the weights equal to one, the NURBS curve transforms itself into a Bézier curve with the same four control vertices. This fact was verified, leading these two RSURFU subroutines to the same results when this simplification is considered. Considering the stated before, the RSURFU user subroutine for the NURBS curve was considered validated.

References

[1] Oliveira M, Padmanabhan R, Baptista A, Alves J, Menezes L (2009) Sensitivity study on some parameters in blank design. Materials and Design 30:1223-1230.

[2] Oliveira M (2005) Algoritmos e estratégias de gestão do problema de contacto com atrito em grandes deformações. PhD Thesis. University of Coimbra (in Portuguese).

[3] ABAQUS® 6.7 (2007) User Manual, Simulia Inc, Dassault Systémes.

[4] Santos A (1993) Tool description and contact strategies used in the static explicit FEM code ITAS3D for simulation of 3-D sheet metal forming processes, PhD Thesis, University of Tokyo.

[5] Santos A, Makinouchi A (1995) Contact strategies to deal with different tool descriptions in static explicit FEM of 3-D sheet-metal forming simulation. Journal of Materials Processing Technology 50:277-291.

[6] de-Carvalho R, Andrade-Campos A, Valente R (2012) Defining analytical rigid curves and surfaces in tool optimization problems. Proceedings of the 3rd International Conference on Engineering Optimization, Rio de Janeiro, Brazil.

[7] Pressley A (2001) Elementary differential geometry, Springer-Verlag.

[8] O´Neill B (1966) Elementary differential geometry, Academic Press, Inc..

[9] Kleinermann J (2000) Identification parametrique et optimisation des procedes de mise a forme par problemes inverses, PhD Thesis, University of Liége (in French).

[10] Grešovnik I (2000) A general purpose computational shell for solving inverse and optimization problems – Applications to metal forming processes, PhD Thesis, University of Wales.

[11] Goldapp M (1991) Approximation of circular arcs by cubic polynomials, Computer Aided Geometric Design 8: 227-238.

[12] Riskus A (2006) Aproximation of a cubic Bezier curve by circular arcs and vice versa, Information Technology and Control 35:371-378.

Chapter 6

Optimization Algorithms

The optimization problem is described and presented. Several one-dimensional and multidimensional unconstrained methods are presented as well as multidimensional constrained problems.

6.1 Introduction

Optimization is one of the main topics of the present work. Concerning this, this chapter presents an introduction to the most used optimization methodologies.

Throughout the ages, optimization processes have made part of the man daily routine. In the past, even the choice for the better day to plant the crops could be considered as an optimization process. These processes were merely based in the common sense and in the experience obtained during the decades.

With the advance of the ages, empirical processes have been replaced by mathematical rules. The interest in optimization has taken a giant leap with the advent of the digital computer in the early fifties. Additionally, with the increasing of the computers efficiency, optimization techniques advanced rapidly in recent years, being now possible to solve complex optimization problems which were thought intractable only a few years ago [1].

Optimization problems can be multidisciplinary, and occur in disciplines such as engineering, physics, mathematics, economics, administration, commerce, social sciences and even politics. Engineering optimization can be applied in a variety of fields, such as electrical, mechanical, civil and chemical. Considering the present work, a special focus on the optimization related to the mechanical engineering will be carried out [1].

The large majority of real-life optimization problems can present several solutions, being sometimes possible an infinite number of solutions. In these cases, the optimization problem consists in finding the best solution of the problem considering a specific performance criterion. In order to solve an optimization problem, several general approaches are available such as analytical methods, graphical methods, experimental methods and numerical methods [1].

Being the most important general approach, optimization based on numerical methods is the main theme of this chapter. In this kind of process, numerical procedures are used to generate a series of progressively improved solutions to the optimization problem and start with an initial estimate solution. This process will be finished when some convergence criterion is achieved. These numerical methods can be used to solve highly complex optimization problems of the type that cannot be solved analytically [1].

6.2 Mathematical programming

Mathematical programming is the discipline that encompasses the theory and practice of numerical optimization methods. It is possible to enumerate a few branches of mathematical programming, such as linear, integer, discrete, quadratic, nonlinear and dynamic programming. Each one of these branches consists on the optimization techniques that are suited to a specific class of optimization problems. The differences between the branches are related to the structure of the objective function and the nature of the constraints [1].

Linear programming (LP) defines a wide range of optimization problems in which the objective function to be minimized is linear in the unknown variables and the constraints are a combination of linear equalities and inequalities. This kind of problems occurs, for example, in many real-life economic situations where profits are to be maximized (or costs minimized) with constraints (limits) on resources. Linear programming is also used for solving nonlinear programming problems (NLP). In these problems, successive linearization of a nonlinear problem leads to a sequence of LP problems which can be efficiently solved. This technique, in particular, is called sequential linear programming (SLP) [2].

In certain optimization problems, at least some of the variables are required to assume only integer values. Some examples are the number of cars that can be purchased, the number of operators that can be assigned to jobs, *etc.*. This class of optimization problems is known as integer programming problems. In cases where some variables have to be integers and other ones can take fractional values, the problem is then known as mixed integer programming problem [1,2].

Other class of mathematical programming is the discrete programming. In this sort of problems, the variables are to be chosen from a discrete set of variables. Several engineering problems fall into this category, such as the selection of shaft sizes, beam sections, engine capacities, *etc.* [2].

The quadratic programming (QP) is a family of optimization methods used to minimize quadratic objective functions subject to linear constraints. The convex quadratic programming is an important class of QP, in which the objective function is a convex quadratic function. A generalization of convex QP is convex programming (CP) in which the objective function is convex, however not necessarily quadratic ,and the feasible region is convex [1].

One of the most general classes of mathematical programming is the nonlinear programming. In this case, the objective function and usually the constraint functions are nonlinear. Linear programming and quadratic programming can be considered as special cases of nonlinear programming. It is possible to solve linear or quadratic programming by using nonlinear programming algorithms. However, there are specialized algorithms for linear or quadratic programming that are much more efficient [1].

The dynamic programming is used for optimization processes where decisions must be made in sequence and subsequent decisions are influenced by the earlier ones. In these cases, a number of optimization processes have to be performed in sequence and a general strategy may be required to achieve an overall optimum solution. Dynamic programming is usually based on the use of linear, integer, quadratic or nonlinear optimization algorithms [1].

6.3 The optimization problem

In general terms, optimization can be described by the minimization or maximization of a function subjected to constraints on its variables. For the optimization problem formulation a performance criterion F is needed, that depends on the n parameters x_1, x_2,..., x_n. This performance criterion can be stated as

$$F = f(x_1, x_2, \dots, x_n), \tag{6.1}$$

where F is a scalar quantity. Based in equation 6.1, the most basic optimization problem can be formulated as follows:

$$\text{minimize} \quad F = f(x_1, x_2, \dots, x_n). \tag{6.2}$$

In optimization problems, F is usually referred to as the objective or cost function. A more compact formulation can be adopted as follows:

$$\text{minimize} \quad F = f(\mathbf{x}) \quad \text{for } \mathbf{x} \in E^n, \tag{6.3}$$

where $\mathbf{x}^{\mathrm{T}} = [x_1\ x_2\ \dots\ x_n]$ and E^n represent the n-dimensional Euclidean space.

In some problems, the goal is to find the maximum of a function instead of the minimum calculation. However, this is an easy task since the maximization of a function is equal to the minimization of its negative function F, followed by the change of the sign of the minimum. This can be mathematically expressed as:

$$\max[f(\mathbf{x})] = -\min[-f(\mathbf{x})]. \tag{6.4}$$

For the resolution of the minimization problem numerous optimization algorithms are available. In a general way, the optimization algorithms follow a typical pattern. In Table 6.1, a general structure for optimization algorithms is presented [1].

Table 6.1 - General structure of an optimization algorithm [1].

```
Step 1
    Set $k = 0$ and initialize $\mathbf{x}_0$.
    Compute $F_0 = f(\mathbf{x}_0)$.
Step 2
    Set $k = k + 1$.
    Compute the changes in $x_k$ given by column vector $\Delta\mathbf{x}_k$, where
    $\Delta\mathbf{x}_k{}^T = [\Delta x_1 \, \Delta x_2 \dots \Delta x_n]$
    by using an appropriate procedure.
    Set $\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta\mathbf{x}_k$.
    Compute $F_k = f(\mathbf{x}_k)$ and $\Delta F_k = F_{k-1} - F_k$
Step 3
    Check if convergence has been achieved by using an appropriate criterion. If this is the case,
    continue to step 4, otherwise go to step 2.
Step 4
    Output $\mathbf{x}^* = \mathbf{x}_k$ and $F^* = f(\mathbf{x}^*)$.
    Stop
```

Considering the knowledge about the optimization problem, normally the optimization procedure starts initializing the vector $\mathbf{x}_0$, that is the initial guess. Steps 2 and 3 are repeated until the convergence is achieved. Each execution of step 2 and 3 constitutes an iteration. In this specific case, k is the iterations' number [1].

The step 4 is executed after the convergence, and in this case, $\mathbf{x}^*$ and F^* are outputs. The column vector $\mathbf{x}^* = [x_1^* \, x_2^* \dots \, x_n^*]$ is said to be the optimum (minimum) solution point or simply the minimizer. Also, F^* is said to be the optimum or minimum value of the objective function. Considering this, the pair $\mathbf{x}^*$ and F^* is the solution of the optimization problem [1].

The convergence achievement depends on the optimization problem and the optimization technique used. One example of a convergence criterion is when the reduction of the objective function value between any two iterations has become insignificant. This can be expressed by the following expression:

$$|\Delta F_k| = |F_{k-1} - F_k| < \varepsilon_F, \tag{6.5}$$

where ε_F is the optimization tolerance for the objective function. Another usual convergence criterion is to stop the optimization algorithm when the changes in all variables have become insignificant. This can be written by the following expression:

$$|\Delta x_i| < \varepsilon_x \qquad \text{for } i = 1, 2, \dots, n\,, \tag{6.6}$$

where ε_x is an optimization tolerance for variables $x_1, x_2, \dots, x_n$. A more complex criterion can be adopted, as the combination of the previous ones, and in this case the convergence is considered only when the last two criterions are satisfied simultaneously [1].

Some optimization problems are conditioned by physical relations and may entail a set of equality constraints and a set of inequality constraints. In this specific case, the problem is said to be a constrained optimization problem. Mathematically, a constrained problem can be formulated as:

$$\begin{array}{lll} \text{Minimize} & f(\mathbf{x}) & \text{for } \mathbf{x} \in E^n \\ \text{subject to:} & a_i(\mathbf{x}) = 0 & \text{for } i = 1, 2, \dots, p \\ & c_j(\mathbf{x}) \leq 0 & \text{for } j = 1, 2, \dots, q \end{array} \tag{6.7}$$

If a problem does not entail any equality or inequality constraints it is then said to be an unconstrained optimization problem. As expected, constrained optimization problems are more difficult to solve than unconstrained optimization problems. Constrained problems can be reformulated as unconstrained ones, making them easier to be solved. This process can be performed redefining the objective function such that the constraints are simultaneously satisfied when the objective function is minimized.

Each point $\mathbf{x}$ that respects all the equality constraints as well as the inequality constraints is known as a feasible point of the optimization problem. The set of feasible points is the feasible domain region of $f(\mathbf{x})$ [1].

6.4 Gradient information

Several optimization methods are gradient-based methods. In these methods, the evolution for a new optimum solution needs the gradient information pertaining to the objective function. Some optimization methods need both the first and the second derivatives of the objective function.

Considering $f(\mathbf{x}) \in C^1$, that means that $f(\mathbf{x})$ has continuous first-order partial derivatives, the gradient vector of $f(\mathbf{x})$ can be defined as:

$$\mathrm{g}(\mathbf{x}) = \nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n}\right]^{\mathrm{T}}. \tag{6.8}$$

Considering $f(\mathbf{x}) \in C^2$, that means that $f(\mathbf{x})$ has continuous second-order partial derivatives, the Hessian matrix of $f(\mathbf{x})$ can be written as:

$$\nabla g(\mathbf{x})^T = \mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \tag{6.9}$$

For the case when $f(\mathbf{x}) \in C^2$, it is possible to deduce that

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}, \tag{6.10}$$

and in this case $\mathbf{H}(\mathbf{x})$ is given by a $n \times n$ square symmetric matrix.

The gradient methods may only use the gradient information or both gradient and Hessian information. However, in certain applications it may be expensive, time consuming or impossible to compute these tensors. In these cases, optimization methods that do not require the gradient information are preferred [1].

6.5 Function extrema

The maxima and minima of a function are its extrema. The points at which a function has minima or maxima are known as minimizers or maximizers, respectively. It is possible to distinguish the weak local minimum, the strong local minimum and the global minimum [1,2].

A point x^* is a weak local minimum of $f(x)$ if there exists a distance $\varepsilon > 0$, such that:

$$f(x) \geq f(x^*), \tag{6.11}$$

for all $|x - x^*| < \varepsilon$.

A point x^* is a strong local minimum of $f(x)$ if there exists a distance $\varepsilon > 0$, such that

$$f(x) > f(x^*), \tag{6.12}$$

for all $|x - x^*| < \varepsilon$.

Finally, a point x^* is a global minimum of $f(x)$ if, for all $x \in E$,

$$f(x) > f(x^*). \tag{6.13}$$

In Figure 6.1 it is possible to graphically illustrate these definitions.

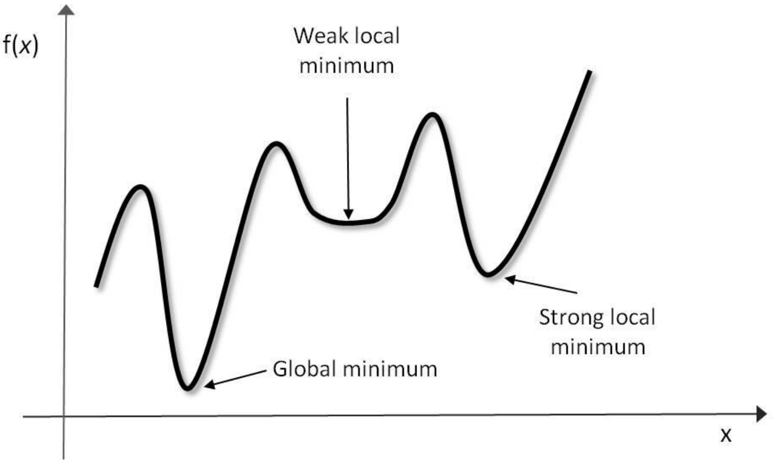


Figure 6.1 - Function extrema.

The main goal of an optimization problem is to find the global minimum (or maximum) of the cost function. Considering that a function can have several local minima, and that the optimization algorithms are iterative procedures, it is possible to understand that the final achieved solution by a gradient-based method is dependent of the considered initial point. In fact, the global minimum can be missed and a suboptimal solution can be found, and this solution can be or not acceptable. This problem can be solved performing the optimization method several times, with different initial estimates, and choosing in the end the best minimizer. Although this solution can be acceptable from a practical point of view, it is not possible to guarantee that the global minimum was achieved. Only for a specific class of problems (where the function satisfy certain convexity properties) any local minimum is also a global minimum and the optimal solution can be assured [1].

6.6 Optimization algorithms

In this chapter, a focus on nonlinear optimization problems is made due to the nonlinear character of the most of the engineering problems to be treated. The nonlinear optimization problems can be divided in three main classes, which are one-dimensional unconstrained problems, multidimensional unconstrained problems and multidimensional constrained problems.

The one-dimensional unconstrained problems are the easiest to solve and the multidimensional constrained are the most difficult ones. In fact, multidimensional constrained problems can be reduced to multidimensional unconstrained problems [1].

6.6.1 One-dimensional unconstrained optimization

For the one-dimensional optimization methods, two general classes can be considered: the search methods and the approximation methods. The search methods start from a bracket $[x_L, x_U]$ that contains the minimum x^*, and then is repeatedly reduced until the new bracket $\left[x_{L,k}, x_{U,k}\right]$ be sufficiently small. The minimizer is assumed to be at the centre of this new interval.

The approximation methods consist in the approximation of the function to be minimized by a low-order polynomial, which will be then minimized considering elementary calculus. The interval $[x_L, x_U]$ is reduced and this processes is repeated until a precise value of x^* is reached. This approach needs a continuous and differentiable optimization function, while the search methods do not pose this requirement [1].

There are several one-dimensional unconstrained optimization algorithms, such as, dichotomous search, Fibonacci search, golden-section search, quadratic interpolation method, cubic interpolation method, Davis, Swann and Campey method, secant method, Newton-Raphson method, bisection method [1,3].

In the present session the bisection method, the secant method and the Newton-Raphson method are described more in detail, since these methods are the most used in mathematical programming.

6.6.1.1 Bisection method

The bisection method can be applied to the optimization of a unimodal function in the search interval, *i.e.*, a function that has only one maximum or one minimum. In this case, the method can be efficiently applied if both the function value and the derivative of the function are available. In this method a first interval $[L, R\ ]$ (where $f'(L) < 0$ and $f'(R) > 0$) is considered. Its midpoint can be calculated as:

$$z = \frac{L + R}{2}. \tag{6.14}$$

The derivative at point z should be calculated. If $f'(z) > 0$, then the interval $[z, R]$ is eliminated from the search. On the other hand, if $f'(z) < 0$ then it is the interval $[L, z]$ that is eliminated. After the choice of the new interval the middle point is recalculated, and this iterative process is performed until the convergence is reached. The process is purely based on the sign of the derivative, not using its magnitude. On the other hand, the secant method is a method that uses both the sign and the magnitude of the derivative [4].

6.6.1.2 Secant method

The secant method can be applied also for a unimodal function. Considering the interval $[L, R\ ]$, such that their derivatives are opposite in sign, the secant algorithm approximates the derivative $f'(x)$ with a straight line between these two points. This line is known as a "secant line". After this approximation, the algorithm determines the next point where the secant line is zero, that implies that the $f'(x) = 0$. Therefore, the next approximation of the point x^* is given by:

$$z = R - \frac{f'(R)}{[f'(R) - f'(L)]/(R - L)}. \tag{6.15}$$

If $|f'(z)| \leq \varepsilon$, the algorithm had reached the optimum value x^*. Otherwise, the interval between z and one of the points L and R (selected in order that their derivatives are opposite in sign) is considered. The next value should be calculated again considering the equation 6.15 and this iterative process should be performed until the convergence is reached [4].

6.6.1.3 Newton-Raphson method

The Newton-Raphson method can only be applied to determine minimums of functions twice differentiable. The iterative process start with a point x_1, that is the initial estimate for the root of the equation $f'(x) = 0$. Then, a linear approximation at point x_1 is computed and the point where the linear approximation vanishes is taken as the next approximation. Mathematically, given the point x_k as the initial estimate, the linear approximation of the function $f'(x)$ at x_k is defined as [4]:

$$y = f'(x_k) + f''(x_k)(x - x_k). \tag{6.16}$$

The objective is to find the point where this approximation vanishes. Therefore, it is possible to calculate the next approximation point as [4]:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}. \tag{6.17}$$

6.6.2 Multidimensional unconstrained optimization

The methods for solving multidimensional unconstrained problems can be classified in two broad categories. These categories are based on the type of information that the algorithm needs to find the minimum. The categories are the (i) direct-search methods and the (ii) gradient-based algorithms.

The direct-search methods only need the objective function value to find the minimum. The gradient-based methods need also the objective function and the second-order methods needs both the first and the second derivatives of $f(x)$. Several algorithms belonging to these categories are presented, since single method (or class of methods) can solve all problems with equal efficiency. Usually, it will be the engineer to choose the best method for each specific optimization problem. This choice should be performed based in the available computer storage, the time needed for each evaluation, the accuracy needed in the final result, the possibility to obtain the derivatives of the objective function, among other factors [4].

In the followings sections it is considered that $f(x)$ has a single minimum in the domain considered. For the cases where $f(x)$ is a multimodal function the minimum found can be a local or the global minimum.

6.6.2.1 Direct-search methods

The direct-search methods can be applied to problems where the gradient does not exist or, alternatively, is very complex. The direct-search methods can be classified into metaheuristic techniques and theoretically based techniques. The metaheuristic techniques are random search methods normally based in natural processes that only guarantee empirical results. The theoretically based techniques have a mathematical foundation guaranteeing convergence under restricted conditions [4].

As examples of metaheuristic methods, it is possible to consider the genetic algorithm [5], simulated annealing [6], particle swarm optimization [7], ant colony optimization [8], artificial neural networks [3] and evolutionary [9] algorithms. The most important theoretically based method is the Powell's conjugate direction method, among others [4,10]. Considering the scope of the present work, more attention will be given to the metaheuristic methods. The metaheuristic methods generate new points in the search space by applying operators to current points and statistically moving toward more optimal places in the search space. This search is considered an intelligent search of large (but finite) solution space using statistical methods. These algorithms doesn´t need the objective function derivatives and therefore, can deal with discrete variables and noncontinuous cost functions, being based in natural processes that are remarkably successful in optimizing natural phenomena. Considering the main objective of the present work, the Evolutionary Algorithms are described in detail and a brief introduction to a nature-inspired optimization algorithm is performed. More information can be found, for instance, in reference [10].

The simulated annealing method, as the name suggests, simulate the annealing process in which a substance is heated above its melting temperature and then is gradually cooled to produce the crystalline lattice, which minimizes its energy probability distribution. This crystalline lattice is composed of millions of atoms that are perfectly aligned, being one example of the nature finding an optimal structure. However, if the rate of change of the temperature is not controlled, the quickly cooling or quenching the liquid retards the crystal formation and leads to an amorphous mass with a higher energy state than the optimum state. The simulated annealing method is analogue to this process and begins with a random guess for the cost function. Heating corresponds to randomly modify the variables values, as the heat increases the random fluctuations increases as well. After modifying the variables values, the cost function returns the output. If the output decreases, then the new set of variables replaces the old one and If the output increases, then the output is accepted provided that [10]:

$$r \leq e^{[f(\mathbf{p}_{\mathrm{old}})-f(\mathbf{p}_{\mathrm{new}})]/T}, \tag{6.18}$$

where r is an uniform random number and T is a variable analogous to temperature. Otherwise, the new set of variables is rejected. Therefore, even if a variable set leads to a worse cost, it can be accepted with a certain probability. The new set of variables is achieved by taking a random step from the old variable, such as [10]:

$$\mathbf{p}_{\mathrm{new}} = d\mathbf{p}_{\mathrm{old}}. \tag{6.19}$$

The variable d can be either uniformly or normally distributed about $\mathbf{p}_{\mathrm{old}}$. This is the control variable that sets the step size and forces the algorithm to make larges changes in the variables values at the beginning of the process. Sometimes it will force the algorithm to go away from the optimum, allowing to explore new regions of variable space. After some iterations the new set of variables doesn´t lead to lower values of the cost function. At this time, the values T and d decrease by a certain percentage and the algorithm repeats. The algorithm will stop when $T \cong 0$. This decrease in T is called the cooling schedule, being possible different cooling schedules. Considering that the initial temperatures is T_0 and the ending temperature is T_N, then the temperature at step n is given by [10]:

$$T_n = f(T_0 T_N, N, n), \tag{6.20}$$

in which f decreases with time. Some potential cooling schedules can be considered such as:

- Linear decreasing : $T_n = T_0 - n(T_0 - T_n)/N$.
- Geometrically decreasing: $T_n = 0.99\, T_{n-1}$.

Other variations are possible and the temperature is usually lowered slowly so that the algorithm is able to find the correct valley before trying to get to the lowest point in the valley.

The Evolutionary Algorithms have many different variants, although the common idea behind all these techniques is the same. The idea is that for a given population of individuals the environment pressure causes natural selection, surviving the fittest and this will cause a rise in the fitness of the population. Considering a quality function that should be maximized, a randomly set of candidates solutions can be considered in the calculation of the quality function and it is then assumed that the higher solution is the better solution. It will be based on this fitness that some of the better candidates are chosen to seed the next generation, by applying recombination and/or mutation to them. Recombination is an operator where two or more selected candidates (called parents) are combined resulting one or more new candidates (the children). Mutation is applied to one candidate resulting in a new candidate. The recombination and mutation lead to a set of new candidates (the "offspring") that will compete with the old ones for a place in the next generation. This process can be iterated until a good candidate is found or a previously computational limit is reached, and a schematic representation of it is presented in Figure 6.2 [11].

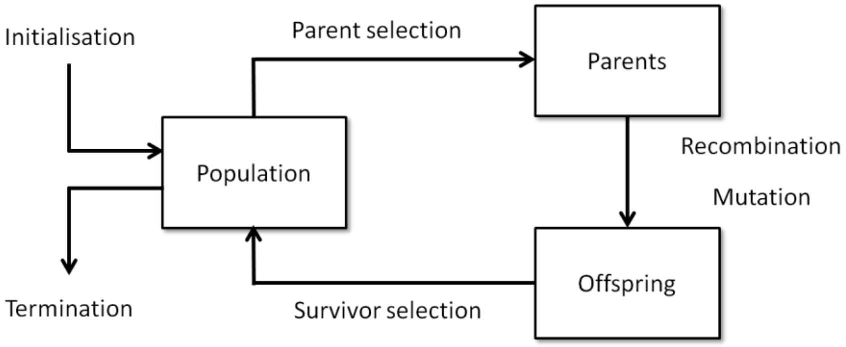


Figure 6.2 - The general scheme of an Evolutionary Algorithm [11].

The combination of the variation operators (recombination and mutation) generally leads to the improvement of the fitness values in consecutive populations.

As Figure 6.2 suggests, the most important components in an Evolutionary/Genetic Algorithm are the representation (definition of individuals), the evaluation function (or fitness function), the population, the parent selection mechanism, the variation operators (recombination and mutation) and the survivor selection mechanism [11].

The representation is the first step in Evolutionary/Genetic Algorithms. In this case it is intended to link the "real world" with the "EA/GA world". To perform this task, objects forming possible solutions within the original problem are known as phenotypes and their encoding, the individuals within EA/GA, are known genotypes. In the representation step, the phenotypes are mapped onto the genotypes that represent these phenotypes and this process is known as encoding. The inverse mapping is usually called decoding [11].

The evaluation function is commonly called the fitness function and is technically a function or procedure that assigns a quality measure to genotypes.

The population is a set of genotypes and its role is to hold possible solutions. The individuals are static objects that not change or adapt, however the population does. The parent selection is related to the population, distinguishing among the individuals, based on their quality, the better individuals to become parents of the next generation. An individual, if selected, become a parent. Together with the survivor selection the parent selection, is responsible to improve the quality [11].

The variation operators are the mutation and the recombination operators. Their role is to create new individuals from old ones. The mutation operator is applied to one genotype and delivers a modified mutant, known as the child or offspring. This operator is always stochastic. The recombination or crossover merges information from two parent genotypes into one or two offspring genotypes. This operator is also a stochastic operator. The principle behind recombination is that by mating two individual with different but desirable features it is possible to produce an offspring which combines both of those features [11].

The survivor selection can be called also as environmental selection and distinguish among individuals based on their quality, being similar to parent selection. However, it is performed in a different stage of the evolutionary cycle. It consists in deciding which individuals will be allowed in the next generation. This decision is made considering their fitness, however sometimes the concept of age is also used. In this case, this process is often deterministic [11].

The initialization is a simple step, being the first population a seed by randomly generated individuals. The termination condition occurs if the optimal fitness level is achieved. However, EA's are stochastic and there are no guarantees to reach the minimum. For these cases, computational constraints can be considered such as the allowed CPU time, the total number of generations and, for a given period of time, the fitness improvement remains under a threshold value [11].

6.6.2.2 Gradient-based methods

The direct methods referred in the previous section are important because very often, in practical engineering problems, the objective function values are the only reliable information. However, the direct methods require an excessive number of function evaluations to find the minimum. It is based on the idea that the direct methods are very time consuming that the gradient-based methods are considered as good bets. As the name suggests, the gradient methods are based on gradient information. The gradient methods range from simple to highly sophisticated methods [1]. Examples of gradient-based methods are the steepest descent method, the Newton´s methods, the conjugate direction methods and the quasi-Newton methods. In this section some of these methods are briefly introduced.

Considering $f(\mathbf{x})$ a real-valued differentiable function, it can be proved that the gradient acts in such a direction that for a given small displacement, the function $f(\mathbf{x})$ increases more in the direction of the gradient than in any other direction. Therefore, the direction in which the $\nabla f(\mathbf{x})$ points is the direction of maximum rate of increase of f at $\mathbf{x}$. As a consequence, the direction in which $-\nabla f(\mathbf{x})$ points is the direction of maximum rate of decrease of f at $\mathbf{x}$. Based on these facts, the direction of negative gradient is a good direction to search for the minimum [3].

The above idea can be expressed by a simple algorithm, which starts at a given point $\mathbf{x}^k$. The next point $\mathbf{x}^{k+1}$ is found moving $\mathbf{x}^k$ an amount $-\alpha_k \nabla f(\mathbf{x}^k)$, where α_k is a positive scalar called the step size. The relation to compute x^{k+1} is as follows:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k). \tag{6.21}$$

This relation is known as a gradient descent algorithm (or simply a gradient algorithm). During the search the gradient varies, tending to zero as it approaches the minimizer. Many different methods use the explained philosophy, however the most popular is the steepest descent method [3].

The steepest descent method is a gradient method in which the step size α_k is chosen to achieve the maximum amount of decrease of the objective function at each individual step. The step size should be always positive and can be achieved by:

$$\alpha_k = \arg\min_{\alpha} f\left(\mathbf{x}^k - \alpha\nabla f(\mathbf{x}^k)\right). \tag{6.22}$$

In a summarized way, the steepest descent algorithm starts at x^k and at each step a line search in the $-\nabla f(\mathbf{x}^k)$ direction is conducted. This line search tries to find the step size that respects equation 6.22, and after this search the new $\mathbf{x}^{k+1}$ is calculated. This process is repeated until the minimum is found. Due to its intrinsic nature, the steepest descent method moves in orthogonal steps [3].

The steepest descent method uses only the first derivatives in the search direction. This method can be improved if second order derivatives are used. This is the case of the Newton´s method, which uses both first and second derivatives and improves the obtained results by the steepest descendent method if the initial point is close to the minimizer. This method starts with an initial guess, where a quadratic approximation of the objective function is constructed. This quadratic function should have the same first and second derivatives as the original objective function. Then the approximation function will be minimized, instead the original objective function. After this minimization, the minimum found for the approximation function will be considered as the initial guest for the next iteration. It is easy to understand that if the objective function is a quadratic function, the algorithm only needs one iteration to find the minimum. The quadratic approximation in the point $\mathbf{x}_k$ can be defined considering the Taylor series expansion of $f(\mathbf{x})$ as follows:

$$q(\mathbf{x}) = f(\mathbf{x}^{\mathrm{k}}) + (\mathbf{x} - \mathbf{x}^{\mathrm{k}})^T \mathbf{g}^k + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{\mathrm{k}})^T \mathbf{H}(\mathbf{x}^{\mathrm{k}})(\mathbf{x} - \mathbf{x}^{\mathrm{k}}). \tag{6.23}$$

The function $f(x)$ have to be twice continuously differentiable and $\nabla^2 f(\mathbf{x})$ have to be positive definite. Considering that the minimum is achieved (when $\nabla q(\mathbf{x}) = 0$), this leads to:

$$\boldsymbol{\nabla} q(\mathbf{x}) = \mathbf{g}^k + \mathbf{H}(\mathbf{x}^{\mathrm{k}})(\mathbf{x} - \mathbf{x}^{\mathrm{k}}) = \mathbf{0}. \tag{6.24}$$

With this information it is possible to deduce that $q(\mathbf{x})$ reaches the minimum at:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{H}(\mathbf{x}^k)^{-1}\mathbf{g}^k. \tag{6.25}$$

This equation represents the Newton´s method and $\mathbf{d}^k = -\mathbf{H}(\mathbf{x}^k)^{-1}\mathbf{g}^k$ is the search direction [3].

The Levenberg-Marquardt is a modification of the Newton´s algorithm. This algorithm is based on the idea that if the Hessian matrix $\mathbf{H}(\mathbf{x})$ is not positive definite, then the search direction $\mathbf{d}^k$ may not point in a descent direction. To avoid this problem, a simple technique to ensure that the search direction is a descent direction is introduced. This technique is the so-called Levenberg-Marquardt modification of the Newton´s algorithm. This modification can be shown by the following equation [3]:

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} - (\mathbf{H}(\mathbf{x}^{k}) + \mu_k\mathbf{I})^{-1}\mathbf{g}^{k}, \tag{6.26}$$

in which $\mu_k \geq 0$.

The idea behind the Levenber-Marquardt modification is as follows. First, consider the symmetric matrix $\mathbf{H}$, that may not be positive definite, where $\lambda_1, \dots, \lambda_n$ are its eigenvalues and $\mathbf{v}_1, \dots, \mathbf{v}_n$ its eigenvectors. The eigenvalues are real but not necessary all positive. Next, the matrix $\mathbf{G} = \mathbf{H} + \mu\mathbf{I}$ (with $\mu \geq 0$) is considered. Considering this, it is possible to deduce that:

$$\begin{aligned} \mathbf{G}\mathbf{v}_i &= (\mathbf{H} + \mu\mathbf{I})\mathbf{v}_i \\ &= \mathbf{H}\mathbf{v}_i + \mu\mathbf{I}\mathbf{v}_i \\ &= \lambda_i\mathbf{v}_i + \mu\mathbf{v}_i \\ &= (\lambda_i + \mu)\mathbf{v}_i. \end{aligned} \tag{6.27}$$

The previous deduction proves that, for all $i = 1, \dots, n$, $\mathbf{v}_i$ is an eigenvector of $\mathbf{G}$ with eigenvalue $\lambda_i + \mu$. Thus, if μ is sufficiently large, then all the eigenvalues of $\mathbf{G}$ are positive and $\mathbf{G}$ is positive definite. Considering this, if the parameter μ_k in the Levenberg-Marquardt modification of Newton´s algorithm is large enough, then the search direction $\mathbf{d}_k = -(\mathbf{H}(\mathbf{x}^{k}) + \mu_k\mathbf{I})^{-1}\mathbf{g}^{k}$ always points in a descent direction. This fact results from a theorem that says that: let $\{\mathbf{x}^{k}\}$ be the sequence generated by Newton´s method for minimizing a given objective function $f(\mathbf{x})$ if $\mathbf{H}\ (\mathbf{x}_k) > 0$ and $\mathbf{g}^{k} \neq 0$ then the direction

$$\mathbf{d}^{k} = -(\mathbf{H}(\mathbf{x}^{k}) + \mu_k\mathbf{I})^{-1}\mathbf{g}^{k} = \mathbf{x}^{k+1} - \mathbf{x}^{k}, \tag{6.28}$$

from $\mathbf{x}^{k}$ to $\mathbf{x}^{k+1}$ is a descent direction for f, existing an $\bar{\alpha} > 0$ such that for all $\alpha \in (0, \bar{\alpha})$. In this case

$$f(\mathbf{x}^{k} + \alpha\mathbf{d}^{k}) < f(\mathbf{x}^{k}). \tag{6.29}$$

According to this, it is also possible to introduce a step size α_k in the Levenberg-Marquardt modification of Newton´s algorithm as follows:

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} - \alpha_k(\mathbf{H}(\mathbf{x}^{k}) + \mu_k\mathbf{I})^{-1}\mathbf{g}^{k}, \tag{6.30}$$

and the descent property still holds. The Levenberg-Marquardt modification reduces to the pure Newton´s method if $\mu_k \to 0$. If $\mu_k \to \infty$, the algorithm approaches a pure gradient method with small step size [3].

Other subgroup of the gradient-based algorithms are the conjugate direction methods. For the previously described multidimensional optimization methods, the direction of search in each iteration depends on the local properties of the objective function, such as the gradient and the

Hessian matrix [1]. As a consequence, methods in which exists a relation between successive search directions are needed. These methods are known as conjugate-direction methods. The conjugate direction methods can be viewed as an intermediate case between the method of steepest descent method and Newton´s method. The conjugate direction methods normally have better performances than the method of steepest descent method, however not as well as the Newton´s method [3]. As for the Newton´s method, the conjugate direction methods are developed for the quadratic optimization problem and then are extended to the general optimization problem [1].

The most important methods of this class are the conjugate-gradient method, the conjugate-direction method, the Fletcher-Reeves method, the Powell´s method, the Partan method, the Hestenes-Stiefel method and the Polak-Ribière method. In the present section, the conjugate direction method, the conjugate gradient method and the Fletcher-Reeves, the Hestenes-Stiefel and the Polak-Ribière modifications are presented.

When the conjugate direction method is applied to a quadratic function of n variables it reaches the solution after n steps. The quadratic function considered in the conjugate direction method is given by [3]:

$$f(\mathbf{x}) = \frac{1}{2}x^T\mathbf{Q}x - \mathbf{x}^T b, \mathbf{x} \in \mathbb{R}^n \; and \; \mathbf{Q} = \mathbf{Q}^T > 0. \tag{6.31}$$

The conjugate directions are computed based on the following definition. Let $\mathbf{Q}$ be a real symmetric $n \times n$ matrix, the directions $d^0, d^1, d^2, \dots, d^m$ are $\mathbf{Q}$-conjugate if for all $i \neq j$, $d^{iT}\mathbf{Q}d^j = 0$ [1]. The conjugate direction algorithm for the equation 6.31 says that, given a starting point x^0 and $\mathbf{Q}$-conjugate directions $d^0, d^1, d^2, \dots, d^{n-1}$ for $k \geq 0$, the next point can be computed as [1]:

$$g^k = \nabla f(\mathbf{x}^k) = \mathbf{Q}\mathbf{x}^k - b, \tag{6.32}$$

$$\alpha_k = -\frac{\mathbf{g}^{kT}\mathbf{d}^k}{\mathbf{d}^{kT}\mathbf{Q}\mathbf{d}^k}, \tag{6.33}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k. \tag{6.34}$$

Other conjugate direction method is the conjugate gradient method. This method does not use specified conjugate directions, but instead computes the directions as the algorithm progresses. In this method, at each algorithm iteration the direction is calculated as a linear combination of the previous direction and the current gradient, in such a way that all the directions are mutually $\mathbf{Q}$-conjugated. Also, in this method and for a quadratic function of n variables, the minimizer can be achieved performing n searches along mutually conjugate directions [1].

The conjugate gradient algorithm starts in the initial point $\mathbf{x}^0$ and finds the next point considering the steepest descent direction, *i.e.*:

$$\mathbf{d}^0 = -\mathbf{g}^0 \tag{6.35}$$

and

$$\mathbf{x}^1 = \mathbf{x}^0 + \alpha_0 \mathbf{d}^0, \tag{6.36}$$

in which

$$\alpha_0 = -\frac{\mathbf{g}^{0T}\mathbf{d}^0}{\mathbf{d}^{0T}\mathbf{Q}\mathbf{d}^0}. \tag{6.37}$$

In the following $k+1$ steps, the direction d^{k+1} is computed as a linear combination of $\mathbf{g}^{k+1}$ and $\mathbf{d}^k$. This is represented by:

$$\mathbf{d}^{k+1} = -\mathbf{g}^{k+1} + \beta_k \mathbf{d}^k \qquad \text{for} \quad k = 0,1,2,\dots. \tag{6.38}$$

The coefficients β_k are defined by:

$$\beta_k = \frac{\mathbf{g}^{k+1T}\mathbf{Q}\mathbf{d}^k}{\mathbf{d}^{kT}\mathbf{Q}\mathbf{d}^k}. \tag{6.39}$$

In the conjugate gradient method the computations are relatively simple, being only more complicated when compared with the steepest descent method. In this method, no line searches are required. Problems associated with the inversion of the Hessian matrix are absent in this method. Additionally, for non-quadratic problems the convergence may not be achieved in rare occasions [1].

As it was described, the conjugate gradient method is a conjugate direction method. This algorithm can be extended to nonlinear function if equation 6.31 is interpreted as a second-order Taylor series approximation of the objective function. These functions behave approximately as quadratics when are near the solution. In quadratics functions, such as equation 6.32, the $\mathbf{Q}$ matrix is the Hessian matrix and is constant. However, in the case of nonlinear functions the Hessian is a matrix that should be re-evaluated at each iteration of the algorithm and this process can be computationally expensive. Based on this idea the following methods appear, trying to avoiding the Hessian evaluation at each iteration. Considering the conjugate gradient method, it was observed that the Hessian $\mathbf{Q}$ only appeared in the α_k and β_k calculations and the α_k calculation can be easily replaced by a line search procedure. Therefore, the following methods only concerns with the modification of the β_k formula (equation 6.39) and are modifications of the conjugate gradient method. This modifications are algebraically manipulations of the β_k formula that allows to eliminate $\mathbf{Q}$. Three different modifications are presented. The first one is the Hestenes-Stiefel formula, given by [3]:

$$\beta_k = \frac{\mathbf{g}^{k+1T}[\mathbf{g}^{k+1} - \mathbf{g}^k]}{\mathbf{d}^{kT}[\mathbf{g}^{k+1} - \mathbf{g}^k]}. \tag{6.40}$$

The other modification corresponds to the Polak-Ribière formula and is given by [1]:

$$\beta_k = \frac{\mathbf{g}^{k+1T}[\mathbf{g}^{k+1} - \mathbf{g}^k]}{\mathbf{g}^{kT}\mathbf{g}^k}. \tag{6.41}$$

The last modification is the well known Fletcher-Reeves formula, and is as follows [1]:

$$\beta_k = \frac{\mathbf{g}^{k+1T}\mathbf{g}^{k+1}}{\mathbf{g}^{kT}\mathbf{g}^{k}}. \tag{6.42}$$

In the quadratic case, these three formulas are all equal. However, this doesn´t happen in the general nonlinear case.

The quasi-Newton methods are another class of methods that do not require explicit expressions for the second derivatives. As the name suggests, the foundation of these methods is the classical Newton method. In the quasi-Newton methods, the direction of search is based on a $n \times n$ direction matrix $\mathbf{S}$ which serves the same purpose as the inverse Hessian in the Newton method. This matrix is an approximation of the Hessian inverse matrix ($\mathbf{H}^{-1}$) and is computed from the available data. As the number of iterations increases, the $\mathbf{S}$ matrix becomes progressively more identical to the $\mathbf{H}^{-1}$ matrix. For the case of convex quadratic objective functions, the $\mathbf{S}$ matrix is equal to the $\mathbf{H}^{-1}$ matrix in $n+1$ iterations. The quasi-Newton methods ranks among the most efficient methods available and are extensively used in numerous applications [3].

In the last years, several distinct quasi-Newton methods have evolved, such as, the rank-one method, the Davidon-Fletcher-Powell method (known as DFP method), the Broyden-Fletcher-Goldfarb-Shanno method (known as BFGS method) and the Fletcher method. The DFP and BFGS methods are explained in the following [3].

The DFP method was originally developed by Davidon in 1959, and was subsequently modified by Fletcher and Powell in 1963, as the name suggests. The approximation matrix $\mathbf{S}$ for the DFP method is as follows [1]:

$$\mathbf{S}_{\mathrm{DFP}}^{k+1} = \mathbf{S}^{\mathbf{k}} + \frac{\Delta\mathbf{x}^k[\Delta\mathbf{x}^k]^{\mathrm{T}}}{[\Delta\mathbf{x}^k]^{\mathrm{T}}\Delta\mathbf{g}^k} - \frac{[\mathbf{S}^{\mathbf{k}}\Delta\mathbf{g}^k][\mathbf{S}^k\Delta\mathbf{g}^k]^{\mathrm{T}}}{[\Delta\mathbf{g}^k]^{\mathrm{T}}\mathbf{S}^k\Delta\mathbf{g}^k}\,, \tag{6.43}$$

with

$$\Delta\mathbf{x}^k = \alpha_k\mathbf{d}^k \tag{6.44}$$

and

$$\Delta\mathbf{g}^k = \mathbf{g}^{k+1} - \mathbf{g}^k. \tag{6.45}$$

The BFGS method was independently suggested in 1970 by Broyden, Fletcher, Goldfarb and Shanno. The approximation matrix $\mathbf{S}$ for the BFGS method is given as [1]:

$$\mathbf{S}_{BFGS}^{k+1} = \mathbf{S}^k + \left(1 + \frac{[\Delta\mathbf{g}^k]^{\mathrm{T}}\mathbf{S}^k\Delta\mathbf{g}^k}{[\Delta\mathbf{g}^k]^{\mathrm{T}}\Delta\mathbf{x}^k}\right)\frac{\Delta\mathbf{x}^k[\Delta\mathbf{x}^k]^{\mathrm{T}}}{[\Delta\mathbf{x}^k]^{\mathrm{T}}\Delta\mathbf{g}^k} - \frac{\mathbf{S}^k\Delta\mathbf{g}^k[\Delta\mathbf{x}^k]^{\mathrm{T}} + \left(\mathbf{S}^k\Delta\mathbf{g}^k[\Delta\mathbf{x}^k]^{\mathrm{T}}\right)^{\mathrm{T}}}{[\Delta\mathbf{g}^k]^{\mathrm{T}}\Delta\mathbf{x}^k}. \tag{6.46}$$

6.6.3 Multidimensional constrained problems

The optimization algorithms so far presented are only valid for the solution of unconstrained optimization problems.

A constrained problem presents a number of technical issues that are not encountered in unconstrained problems. As an example, it is possible to consider the unconstrained optimization problem where the search is performed in the direction of the negative gradient. However, this problem cannot be extrapolated to constrained problems because the points along such search direction may not satisfy the constraints and, in this case, the search will not conduct to the real solution of the problem. As a consequence, new methods to solve constrained optimization problems have to be considered [3].

Most of the constrained optimization methods are based on the unconstrained optimization methods. In the case that the constraints are simply given in terms of lower and/or upper limits on the parameters, the problem can be converted into an unconstrained problem. Other methods transform the constrained problems into a sequence of unconstrained minimization problems. In the present section the optimization problem from equation 6.2 is considered [3].

Different constrained optimization algorithms exists, such as, the gradient-based method Rosen´s gradient projection method for linear constraints, the Zoutendijk´s method of feasible directions, the generalized reduced gradient method, the sequential quadratic programming method, penalty methods, projection methods, among others [2]. In the present section, a projection method is presented as well as the penalty methods.

For a better understanding of the presented constrained methods, it is needed to introduce the feasible point concept. A feasible point is a point that satisfies all inequalities and equalities constraints [2]. The projection method is based on the considered methods, where

$$x^{k+1} = \mathbf{x}^k - \alpha_k \mathbf{d}^k, \tag{6.47}$$

and d^k is normally a function of $\nabla f(x^k)$. This algorithm cannot be directly applied to solve an constrained optimization problem. Consider the following optimization problem

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) \\ &\text{subjected to:} \quad \mathbf{x} \in \Omega \end{aligned} \tag{6.48}$$

The algorithm from equation 6.47 can lead to a point x^k that doesn´t satisfy the constraints, that is, is not a feasible point. Based on this fact, the projection method is formulated. In the projection algorithm, if $\mathbf{x}^k - \alpha_k \mathbf{d}^k$ is in Ω, then $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \mathbf{d}^k$. On the other hand, if $\mathbf{x}^k - \alpha_k \mathbf{d}^k$ is not in Ω, then a projection to Ω is performed before the $\mathbf{x}^{k+1}$ calculation [1].

The point $\Pi[\mathbf{x}]$ is known as the projection of $\mathbf{x}$ onto Ω. In fact, $\Pi[\mathbf{x}]$ is the closest point in Ω to $\mathbf{x}$. Considering the projection operator Π, the previous unconstrained algorithm can be modified to:

$$\mathbf{x}^{k+1} = \Pi[\mathbf{x}^k - \alpha_k \mathbf{d}^k]. \tag{6.49}$$

Now, all the $\mathbf{x}^{k+1}$ lies inside Ω. Generically the projection onto Ω can be defined as:

$$\Pi[\mathbf{x}] = \arg\min_{\mathbf{z}\in\Omega} \|\mathbf{z} - \mathbf{x}\|. \tag{6.50}$$

For the penalty methods, the optimization problem considered in equations 6.7 is taken into account. The feasible region Ω is defined as $\Omega = \{\mathbf{x}: \mathbf{a} = \mathbf{0}, \mathbf{c} \leq \mathbf{0}\}$. The constrained problem considered can be converted in the unconstrained optimization of a new composite function:

$$T(\mathbf{x}, r) = f(\mathbf{x}) + rP(\mathbf{x}), \tag{6.51}$$

where $r > 0$ is called the penalty parameter, and this value controls the degree of penalty for violating the constraints. $P(\mathbf{x})$ is the penalty function, and is defined as:

$$P(\mathbf{x}) = \sum_{i=1}^{p} [a_i(\mathbf{x})]^2 + \sum_{j=1}^{q} \{\max[0, c_j(\mathbf{x})]\}^2. \tag{6.52}$$

For the cases that $\mathbf{x}$ is inside of the feasible region, the penalty function assumes as null. Afterwards, the initial problem is now the unconstrained minimization of $T(\mathbf{x})$. If the constrains are violated, a big term will be added to $f(\mathbf{x})$ function, such that the solution is pushed back towards to the feasible region. Since the convergence is from the outside of the feasible region, these methods are called “exterior penalty methods” [2,12].

There are other methods that approach the optimum from the interior of the feasible region, being known as interior point or barrier methods. For these methods, only inequality constraints are allowed, being the feasible region defined as $\Omega = \{\mathbf{x}: \mathbf{c} \leq \mathbf{0}\}$. In these methods, the new unconstrained function is defined as:

$$T(\mathbf{x}, r) = f(\mathbf{x}) + \frac{1}{r} B(\mathbf{x}). \tag{6.53}$$

In this case, $B(\mathbf{x})$ is the interior penalty function or barrier function. This function has to be continuous, and also $B(\mathbf{x}) \geq 0$ and $B(\mathbf{x}) \to \infty$ as $\mathbf{x}$ approaches the boundary of Ω. The most used interior penalty functions are the inverse barrier function, defined as [2]:

$$B(\mathbf{x}) = -\sum_{j=1}^{q} \frac{1}{c_j(\mathbf{x})}, \tag{6.54}$$

and the log barrier function, defined as:

$$B(\mathbf{x}) = -\sum_{j=1}^{q} \log[-c_j(\mathbf{x})]. \tag{6.55}$$

6.7 Algorithms used in the present work

6.7.1 Evolutionary Algorithm (EA)

In the present work, an evolutionary algorithm was used in the optimization process, being schematically presented in Table 6.2. It is possible to see that all the processes considered in section 6.6.2.1 are presented. Further information about the implementation of this algorithm can be found in [12,14]

Table 6.2 - Evolutionary Algorithm.

```
Initialize the population with random candidate solutions
Evaluate the objective function at each candidate
    REPEAT
        Select parents
        Recombine pairs of parents
        Mutate the resulting offspring
        Evaluate new candidates
        Select individuals for the next generation
    UNTIL convergence is reached or the maximum number of iteration is achieved
End
```

6.7.2 Levenberg-Marquardt (LM) algorithm

The gradient-based optimization methods are known by its computational efficiency and assurance of the convergence to a local minimum [13,14]. In the present work, the gradient-based algorithm used combines two classical minimization techniques: i) the steepest descent method, used to enhance the starting set of parameters, and ii) the Levenberg-Marquardt method [15], used to accelerate the convergence towards the ending stages of the optimization process [14]. In Table 6.3 it is presented the gradient-based algorithm adopted in the present work.

Table 6.3 - Levenberg-Marquardt algorithm.

```
Chose the starting set (initial guess) and other initial process parameters
Evaluate the objective function at the initial guess
Chose the Levenberg-Marquardt parameter $\mu_k$
Convergence by Levenberg-Marquardt method:
    REPEAT
        Calculate the gradient $\mathbf{g}^{(\mathrm{k})}$
        Solve the equations system $\mathbf{d}^{(k)} = -(\mathbf{H}(\mathrm{x}^{(k)}) + \mu_k \mathbf{I})^{-1}\mathbf{g}^{(\mathrm{k})}$
        Update $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}^{(k)}$
        Check for convergence
        Update $\mu_k$
    UNTIL convergence is reached or the maximum number of iteration is achieved
End
```

References

[1] Antoniou A, Lu W (2007) Practical Optimization – Algorithms and Engineering Applications. Springer-Verlag.

[2] Belegundu A, Chandrupatla T (1999) Optimization Concepts and Applications in Engineering. Prentice-Hall.

[3]. Chong E, Zak S (2001) An Introduction to Optimization. John Wiley and Sons.

[4] Ravindran A, Ragsdell K, Reklaitis G (2006) Engineering optimization – Methods and applications. John Wiley and Sons.

[5] Holland J (1975) Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press.

[6] Kirkpatrick S, Gelatt C, Vecchi M (1983) Optimization by simulated annealing. Science 220:671-680.

[7] Parsopoulos K, Vrahatis M (2002) Recent approaches to global optimization problems through particle swarm optimization. Kluwer Academic, 1:235-306.

[8] Dorigo M, Maria G (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. 1:53-66.

[9] Schwefel H (1995) Evolution and optimum seeking. John Wiley and Sons.

[10] Haupt R, Haupt S (2004) Practical genetic algorithms. John Wiley and Sons.

[11] Eiben A, Smith J (2007) Introduction to evolutionary computing. Springer-Verlag.

[12] de-Carvalho R, Valente R, Andrade-Campos A (2011) Optimization strategies for non-linear material parameters identification in metal forming problems. Computer & Structures 89:246-255.

[13] Pilvin P (1990) Approches multichelles pour la prévision du comportement anélastique des métaux. PhD thesis. Université Paris (in French).

[14] Andrade-Campos A, Thuillier S, Pilvin P, Teixeira-Dias F (2007) On the determination of material parameters for internal variable thermoelastic-viscoplastic constitutive models. International Journal of Plasticity 23:1349-1379.

[15] Marquardt D (1962) An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics 11:431-441.

Chapter 7

The Program SDL Optimization Lab

A brief introduction underlining the importance of the SDL Optimization Lab is performed. The general structure of the program is explained as well as the problem definition for the correct implementation of optimization problems. Issues like integration methods and the gradient calculation are also presented.

7.1 Introduction

Nowadays optimization processes are increasing and becoming a fundamental stage in engineering processes. Process and product optimization, inverse problems, shape optimization and topology optimization are frequent problems in both industry and scientific communities [1-8]. In order to solve this kind of problems, general mathematical/technical computing software, such as MatLab [9] and Mathematica [10], programming languages (e.g. C++, Fortran, java, *etc*.) or the combination of these and Finite Element Analysis (FEA) commercial software (*e.g.* Abaqus® [11], LS-DYNA [12]) are generally used [8].

MatLab and Mathematica are used in wide fields of engineering and have their own optimization toolboxes. However, they are general-purpose software and are not adapted for solving complex mechanical problems. In this case, these can be combined with FEA software, being this approach more versatile. Other commonly used approach is the combination of programming languages and FEA software. In these cases, it is necessary to write and implement the whole optimization algorithm, including the objective function, the optimization method, the input/output data, *etc.*. Although computationally efficient, this approach can be time consuming, and requires a full knowledge of programming languages [8].

New commercial engineering optimization software packages start to be used by many researchers and technicians. ModeFrontier [13], Knitro [14], Heeds [15], among others, are multidisciplinary and multi-objective software, written to allow an easy coupling to any computer aided engineering (CAE) tool, being easily adapted to mechanical problems resolution. However, these packages have the disadvantage of being a kind of “closed *black-box*”, where the user cannot change any details in the optimization methodologies or, in some cases, cannot implement new optimization methods [8].

In the present chapter an engineering optimization framework that aims to overcome the previously mentioned disadvantages is presented. The SDL Optimization Lab [8] is a non-commercial framework designed for specific engineering inverse problems (such as parameter identification, initial shape optimization and initial tool shape optimization) [5-7,16,17] and shape optimization problems [18,19].

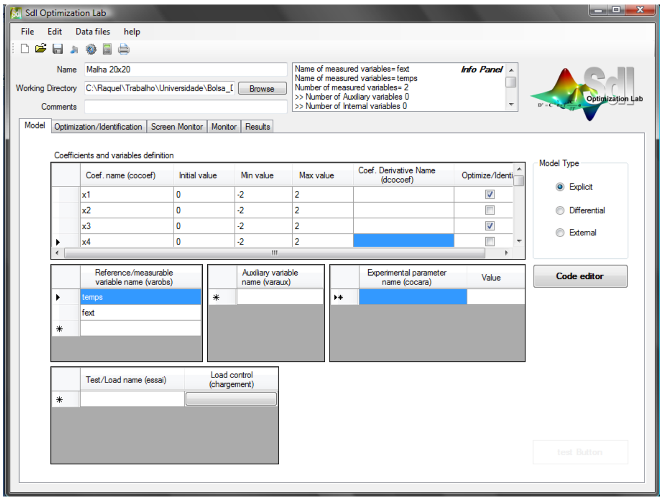
The shape optimization problem can be similar to the parameter identification problem if the shape to be optimized is defined by a finite number of parameters. Therefore, both problems can be solved by the same approach [8].

The SDL Optimization Lab inherits the large experience coming from the SIDOLO code [16,20] and adds the latest developments in direct search optimization algorithms. It can also be used by researchers that wishes to control every step of the engineering optimization procedure or by users with no experience in programming languages, that want to try an intuitive graphical interface. The user subroutines capabilities in SDL Optimization Lab allow the program to be customized for particular applications or, alternatively, to implement new optimization methods and strategies [8]. The SDL Optimization Lab framework is still being developed, particularly for parameters identification and inverse shape optimization problems in the GRIDS research group.

7.2 General structure

The SDL Optimization Lab is an interface where the user can define the optimization problem. It allows to define: i) initial and problem data information; ii) objective function definition; iii) gradient calculations; iv) core algorithm definition. It also allows to have a real time monitoring and graphically see the results. It is possible to observe in Figure 7.1 the problem data definition for an example problem. In Figure 7.2 the real time monitoring is presented and in Figure 7.3 the results are shown.

The first step in using the SDL Optimization Lab is to define the type of formulation used in the mathematical/numerical relations that the considered model use in the determination of the observable variables. The observable variables $\mathbf{Z}(t)$ are the variables that can be measured, such as stress, strain, forces *etc.*.



Sdl Optimization Lab
File Edit Data files help
Browse
Info Panel
Optimization Lab
Model Optimization/Identification Screen Monitor Monitor Results
Code editor
test Button

Name: Malha 20x20

Working Directory: C:\Raquel\Trabalho\Universidade\Bolsa_[

Comments:

Name of measured variables= fext
Name of measured variables= temps
Number of measured variables= 2
>> Number of Auxiliary variables 0
>> Number of Internal variables 0

Coefficients and variables definition

Coef. name (cocoef)	Initial value	Min value	Max value	Coef. Derivative Name (dcocoef)	Optimize/Identi
x1	0	-2	2		☑
x2	0	-2	2		□
x3	0	-2	2		☑
x4	0	-2	2		□

Model Type: ● Explicit ○ Differential ○ External

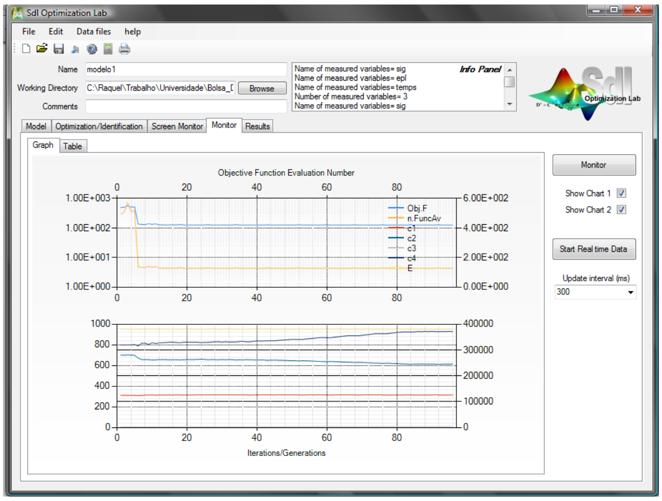
Reference/measurable variable name (varobs)
temps
fext

Auxiliary variable name (varaux)

Experimental parameter name (cocara)	Value

Test/Load name (essai)	Load control (chargement)

Figure 7.1 - SDL Optimization Lab GUI: problem data definition.



Name: modelo1

Working Directory: C:\Raquel\Trabalho\Universidade\Bolsa_[

Comments:

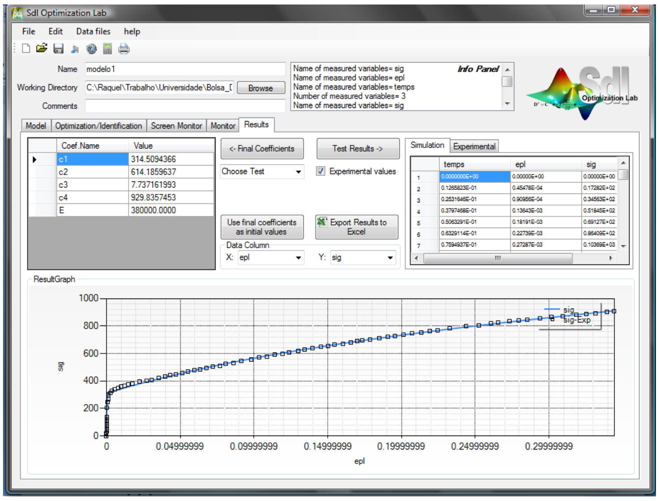
Name of measured variables= sig
Name of measured variables= epl
Name of measured variables= temps
Number of measured variables= 3
Name of measured variables= sig

Show Chart 1 ☑

Show Chart 2 ☑

Update interval (ms): 300

Figure 7.2 - SDL Optimization Lab GUI: real time monitoring.



Name: modelo1

Working Directory: C:\Raquel\Trabalho\Universidade\Bolsa_[

Comments:

Name of measured variables= sig
Name of measured variables= epl
Name of measured variables= temps
Number of measured variables= 3
Name of measured variables= sig

Coef.Name	Value
c1	314.5094366
c2	614.1859637
c3	7.737161993
c4	929.8357453
E	380000.0000

Choose Test

☑ Experimental values

Data Column X: epl Y: sig

	temps	epl	sig
1	0.0000000E+00	0.00000E+00	0.00000E+00
2	0.1265823E-01	0.45478E-04	0.17282E+02
3	0.2531646E-01	0.90956E-04	0.34563E+02
4	0.3797468E-01	0.13643E-03	0.51845E+02
5	0.5063291E-01	0.18191E-03	0.69127E+02
6	0.6329114E-01	0.22739E-03	0.86409E+02
7	0.7594937E-01	0.27287E-03	0.10369E+03

Figure 7.3 - SDL Optimization Lab GUI: results.

Three types of formulation in the observable variables determination can be considered: i) explicit; ii) differential and iii) external. It is possible to see this option in Figure 7.1 in the right side. The explicit option should be considered if the explicit relations are used in the observable variables determination, i.e.:

$$\mathbf{Z}(t) = \mathbf{H}(\mathbf{A}, t). \tag{7.1}$$

In this case, $\mathbf{Z}(t)$ is a function of $\mathbf{A}$, which are the model parameters, and t, which represents the time, the incremental evolution of an event, instants of observation or different points of observation. An example of this relation can be the elastic relation:

$$\sigma = E\varepsilon(t) \tag{7.2}$$

In cases where the determination of $\mathbf{Z}(t)$ requires the resolution of a system of first order differential equations, a differential model should be chosen. The evolution of a differential problem can be defined by a system of equations expressed by:

$$\frac{\mathrm{d}\mathbf{Y}}{\mathrm{d}t} = \mathbf{F}(\mathbf{Z}_e, \mathbf{Y}, \mathbf{A}, t) \quad \text{with} \quad Y(t_0) = Y_0, \tag{7.3}$$

$$\mathbf{Z}(t) = \mathbf{G}(\mathbf{Y}, \mathbf{A}, t). \tag{7.4}$$

In this case $\mathbf{Z}(t)$ is also a function of $\mathbf{Y} = \{y_1, y_2, \dots, y_n\}^{\mathrm{T}}$**,** which represents the set of n variables defined in differential (or rate) form. One example can be the evolution of the yield function, considering isotropic and kinematic hardening:

$$f(\boldsymbol{\sigma}, \bar{\varepsilon}^{\mathrm{p}}) = \sigma_e\big(\boldsymbol{\sigma} - \boldsymbol{X}(\bar{\varepsilon}^{\mathrm{p}})\big) - \sigma_{\mathrm{y}0} + r(\bar{\varepsilon}^{\mathrm{p}}), \tag{7.5}$$

where the backstress can be formulated by a differential equation, such as, the Prager kinematic hardening law:

$$\mathrm{d}\mathbf{X} = c\mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{p}}. \tag{7.6}$$

Exists one more option which is the external model. This option should be considered when the $\mathbf{Z}(t)$ calculation is performed by an external program. In this case, it is necessary to build up an interface between the simulation program and the SDL Optimization Lab. A general interface is available in SDL Optimization Lab, however this interface requires little modifications in order to take in account specific characteristics of the program. For simulation programs widely spread (such as Abaqus®, Msc.Marc, DD3Imp, among others) pre-developed interfaces are available [8].

In the present work, differential mode was used in the parameters optimization, and the external mode was used for the initial shape optimization and initial tool shape optimization.

7.3 Problem definition

After correctly choose the model type, it is fundamental to define the initial and problem data information. The optimization variables should be defined with the respective names and boundaries. This task can be performed in the SDL Optimization Lab interface, as shown in Figure 7.1. The names of the measured variables should also be defined. The objective function and the mathematical models (in the case of parameter optimization) should be implemented considering the code editor present in the SDL Optimization Lab interface. To finish the problem definition, experimental values should be defined. With all this information, the problem is correctly established and it is then possible to create an executable file. This file is self-sufficient and can be used in other platforms, such as single computers or clusters, without any adjustment. For the result analysis real time monitoring is available in SDL Optimization Lab [8].

7.4 Integration methods

One issue that is important when a differential problem is considered, refers to know which integration methods are used in the resolution of the first order partial derivatives (see equation 7.3). For this sort of problems, the SDL Optimization Lab can use the Runge-Kutta explicit integration methods of second (rk21), fourth (rk43) and fifth (rk54) orders. The explicit Runge-Kutta methods for the solution of the differential equations 7.3 are of the form:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{j=1}^{\nu} b_j \mathbf{k}_j \tag{7.7}$$

with

$$\begin{aligned} \mathbf{k}_1 &= f(t_n, \mathbf{y}_n), \\ \mathbf{k}_2 &= f(t_n + c_2 h, \mathbf{y}_n + a_{21} h \mathbf{k}_1), \ldots, \\ \mathbf{k}_\nu &= f\left(t_n + c_\nu h, \mathbf{y}_n + h \sum_{j=1}^{\nu-1} a_{\nu,j}\, \mathbf{k}_{1j} \right). \end{aligned}$$

These methods have proved to be efficient, and this efficiency is more evident when using an adaptive time step algorithm [16], which is the case of SDL Optimization Lab.

7.5 Gradient evaluation

In gradient-based algorithms, one of the main tasks is the efficient computation of the objective function gradients. This process can be also called “sensitivity analysis”. The simplest way to obtain the derivatives is using a numerical difference method, such as the Newton´s forward difference formula. In this case, considering Ψ as the objective function whose sensitivities are needed to be computed, it is possible to write:

$$\frac{\partial \Psi(\mathbf{A})}{\partial A_i} \approx \frac{\Psi(A_1, A_2, \ldots, A_i + \varepsilon, \ldots, A_r) - \Psi(\mathbf{A})}{\varepsilon}. \tag{7.8}$$

In equation 7.8, ε is the divided difference parameter and each variable is perturbed (one at a time), and Ψ is then re-evaluated. As a consequence, it is possible to understand that $\boldsymbol{r+1}$ function evaluations are needed to compute the gradient of Ψ at $\mathbf{A}$. This scheme is not recommended in problems where each function evaluation is computationally time consuming. For greater accuracy, at a higher CPU cost, the central difference formula can be used [8]:

$$\frac{\partial\Psi(\mathbf{A})}{\partial A_i} \approx \frac{\Psi(A_1, A_2, \dots, A_i + \varepsilon/2\,, \dots, A_r) - \Psi(A_1, A_2, \dots, A_i - \varepsilon/2\,, \dots, A_r)}{\varepsilon}. \qquad (7.9)$$

References

[1] Andrade-Campos A, Thuiller S, Pilvin P, Teixeira-Dias F (2007) On the determination of material parameters for internal variable thermoelastic-viscoplastic constitutive models. International Journal of Plasticity 23:1349-1379.

[2] Ponthot J, Kleinermann J (2006) A cascade optimization methodology for automatic parameter identification and shape/process optimization in metal forming. Computational Methods in Applied Mechanics and Engineering 195:5472-5508.

[3] Belegundu A, Chandrupatla R. (1999) Optimization Concepts and Applications in Engineering. Prentice-Hall.

[4] Ceretti E, Braga D, Giardini C (2010) Optimization of process parameters and part geometry for high diameter tube hydroforming. Proceeding of the 13th ESAFORM conference 2010, Brescia University, Italy.

[5] de-Carvalho R, Valente R, Andrade-Campos A, (2011) Optimization strategies for non-linear material parameters identification in metal forming problems. Computer and Structures 89:246-255.

[6] de-Carvalho R, Silva S, Valente R, Andrade-Campos A (2011) The geometry definition influence in inverse analysis – Application to carter forming process. Proceeding of the 14th ESAFORM Conference 2011, Belfast, Ireland.

[7] de-Carvalho R, Andrade-Campos A, Valente R (2012) Defining analytical rigid curves/surfaces in tool optimization problems. Proceeding of the 1st Young Investigators Conference 2012, Aveiro, Portugal.

[8] Andrade-Campos A (2001) Development of an optimization framework for parameter identification and shape optimization problems in engineering. International Journal of Manufacturing, Materials, and Mechanical Engineering 1:57-79.

[9] Matlab (2007) The Language of technical Computing, User Manual, v. R2007. The Mathworks Inc..

[10] Mathematica (2009) Wolfram Mathematica 7, Wolfram Research, Inc..

[11] ABAQUS 6.7 (2007) User Manual, Simulia Inc, Dassault Systémes.

[12] LS-DYNA Theory Manual Livermore Software Technology Corporation.

[13] ModeFrontier (2008) ESTECO ModeFrontier - multi-objective optimization and design environment software, v4.0, ESTECO.

[14] KNITRO Optimization software (2007), Ziena Optimization Inc..

[15] Heeds (2009) Professional Design Optimization Software, Red Cedar Technology Inc..

[16] Cailletaud G, Pilvin P (1994) Identification and inverse problems related to material behaviour, Proceedings of the International Seminar on Inverse Problems, France, 79-86.

[17] Liu G, Han X (2003) Computational Inverse Techniques in Nondestructible Evaluation, CRC Press, USA.

[18] Maniatty A, Zabaras N (1996) Investigation of regularization parameters and error estimating in inverse elasticity problems. International Journal for Numerical Methods in Engineering, 37:1039-1052.

[19] Fourment L, Balan T, Chenot J (1996) Optimal design for non-steady-state metal forming processes. II. Application of shape optimization in forging. International Journal for Numerical Methods in Engineering, 39:51-65.

[20] Andrade-Campos A, Thuiller S, Pilvin, P, Teixeira-Dias F (2007) On the determination of material parameters for internal variable thermoelastic-viscoplastic constitutive models. International Journal of Plasticity, 23:1349-1379.

III Parameter Identification

Chapter 8

Experimental Material Characterization for Parameter Identification

The parameter identification problem consists in the comparison between experimental results (coming from mechanical tests) and a constitutive model that characterizes the studied material. Concerning this, in the present chapter, the mechanical characterization of the aluminium alloy AA6082 is presented. This is a generic procedure in the characterization of sheet metal forming materials. Tensile, shear and bulge tests are conducted. The mechanical characterization is performed using the ARAMIS system.

8.1 Introduction

Aluminium alloys are one of the most used materials in sheet metal forming industry. This is due to their advantageous strength to weight ratio. The metallic sheet is obtained by means of rolling process. However, only a few of this aluminium alloys are important for the metal forming industry. The aluminium alloys are categorized considering the metallic element that is more abundant. The aluminium alloys from 6XXX series correspond to approximately 80% of the annual worldwide production of extrusion components. In this alloy, the Magnesium and Silicon are the principal metallic elements, however these alloys have small quantities of copper, chromium and manganese [1].

In order to correctly simulate a forming process, it is necessary, a priori, to know the parameters that define the constitutive model that traduces the material behaviour. This identification process is known parameter identification problem and is one of the main topics of the present work. The number of mechanical tests to perform, in order to characterize mechanically the material, depends on the constitutive model chosen.

The sheet metal forming process allows the achievement of complex shapes starting from a metallic sheet. This metallic sheet has small thickness, and this fact influence the mechanical tests that can be performed in order to characterize the material. Considering this, the mechanical tests performed need to be in the sheet plane, however this characterization is simple considering the conventional tests: tensile, shear, bulge, *etc.*. Compression tests in the sheet plane are impossible to execute leading to difficulties in finding some constitutive parameters. The more relevant information that can be obtained from these tests is the young modulus, the yield stress and the anisotropy coefficients [2].

In order to determine the constitutive parameters, the information more relevant is the stress evolution with the strain, the well-known stress-strain curves. In this chapter the experimental tests more used in the characterization of metallic sheets are shortly presented. Tensile, shear and bulge tests were performed in order to characterize mechanically the AA6082, and the tests were performed using the ARAMIS device [3]. The present chapter intends to demonstrate how a material should be mechanically characterized for a correct parameter identification. However, it should be mentioned that the results obtained here are not used in the other chapters of this work, since, it was intended to compare the results with the ones of the literature, where a different material is considered.

8.2 Material used

The material characterized in the present work is the aluminium alloy AA6082. The Aluminium alloy 6082 was considered due to its main characteristics, being a medium strength alloy with excellent corrosion resistance. Concerning the 6000 series, this alloy has the highest strength and is known as a structural alloy. It is the alloy most commonly used in machining and, as a relatively new alloy, has replaced the 6061 in many applications due to its higher strength. In this alloy, the addition of a large amount of manganese controls the grain structure and this leads to a stronger alloy. It is complicated to extrude being the extrude surface finish not as smooth as other similar strength alloys in 6000 series. The AA6082 is typically used in highly stressed applications, trussed, bridges, cranes, transport applications, ore skips, beer barrels, milk churns, *etc.*. Different heat treatments can be considered for this alloy, however the most common are the T6 (solution heat treated and artificially aged), T4 (solution heat treated and naturally aged to substantially stable condition) and T651 (solution heat treated, stress relieved by stretching then artificially aged). The alloy 6082 is usually supplied as a channel, angle, tee, square bar, square box section, rectangular box section, flat bar, tube and sheet (that is the case of the present work) [4]. The chemical composition of the aluminium alloy 6082 is presented in Table 8.1.

Table 8.1 - Chemical composition for the aluminium alloy 6082 [5].

%SI	%Fe	%Cu	%Mn	%Mg	%Zn	%Ti	%Cr	%others	%Al
0.70-1.30	0.50	0.10	0.40-1.00	0.60-1.20	0.20	0.10	0.25	0.10	The remaing

The material was received as a sheet with 2 mm thickness, however nothing was said about the heat treatment considered. Comparing the results with the ones presented in [5], it is possible to conclude that the material considered have a heat treatment T6/T651. The main mechanical properties are presented in Table 8.2, where σ_m is the maximum value of the nominal stress, $\sigma_{0.2\%}$ is the nominal stress for a 0.2% of deformation and E the Young modulus.

Table 8.2 - Main mechanical properties of the aluminium alloy 6082 with a heat treatment T6/T651 [5].

σ_m [MPa]	$\sigma_{0.2}$ [MPa]	E [MPa]
340	310	70

8.3 The ARAMIS system

ARAMIS[1] is a non-contact optical 3D deformation measuring device that allows to analyse, calculate and document deformations. The ARAMIS device is composed by two digital cameras that recognize the surface structure of the object to be measured and the software allocates coordinates to the image pixels. In the deformation measuring processes, one initial photo is taken and this image will be considered as the undeformed object state. During the mechanical deformation process, further images are recorded and compared between them in order to track the pixels. This process allows to calculate directly the displacement and deformation of the object. In the case that the measuring object is homogeneous, the surface must be prepared in order to have contrast between the pixels considered by the cameras. One of the most suitable methods to perform this preparation is by applying a stochastic colour spray pattern, as shown in Figure 8.1. The ARAMIS device is used in different fields such as material testing, component dimensioning, examination of non-linear behaviour, strain computation, *etc.* [3].

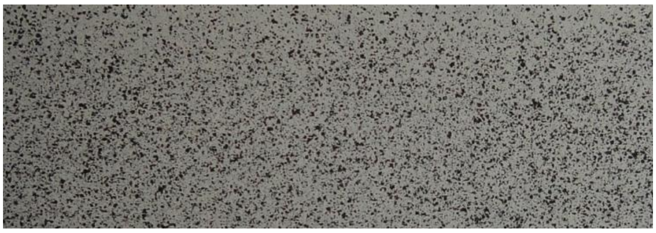


Figure 8.1 - Stochastic spray pattern applied in a metal specimen.

8.4 Mechanical tests

The mechanical behaviour of the AA6082 is investigated under three different stress and strain states, *i.e.* uniaxial tension, simple shear (both of these tests are performed at several orientations to the rolling direction or RD) and bulge test. The experimental procedure is described in the following paragraphs.

[1] ARAMIS is a registered mark from gom - Optical Measuring Techniques.

To perform the mechanical tests and use the ARAMIS device it is necessary to prepare the specimens as it was mentioned before with the stochastic colour spray pattern. In the figures are presented the tensile, shear and bulge specimens prepared to be analysed with the ARAMIS.

The main results are presented in the following. These results are obtained considering the ARAMIS system. The first set of stress-strain curves are obtained considering the average of the deformation in the considered useful area of the specimen. Later on, some stress-strain curves are presented, where the strain measurement is performed for a specific point. This is one of the main potentialities of the ARAMIS system.

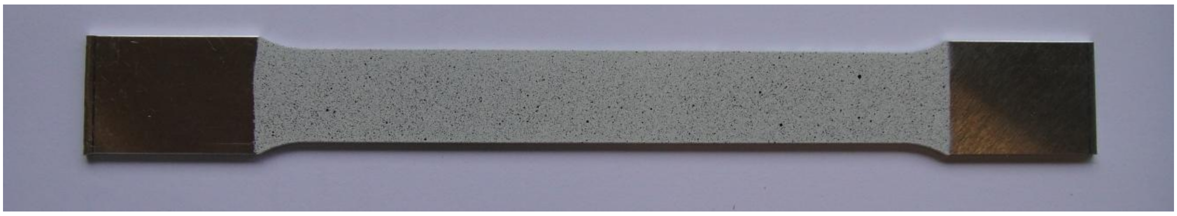


Figure 8.2 - Tensile specimen prepared to be analysed with the ARAMIS device.

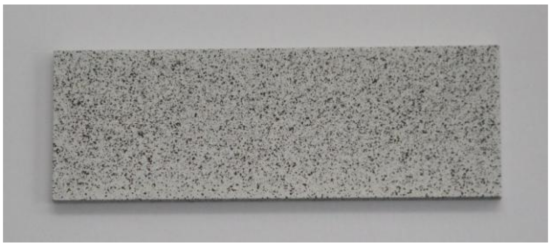


Figure 8.3 - Shear specimen prepared to be analysed with the ARAMIS device.

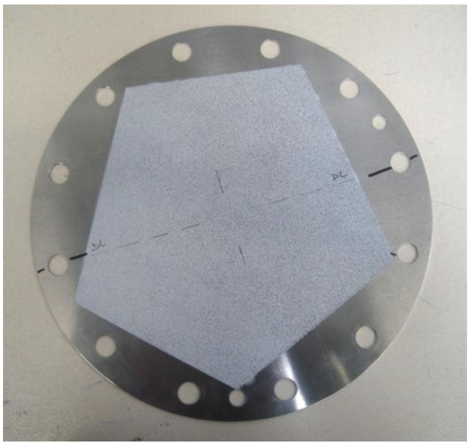


Figure 8.4 - Bulge specimen prepared to be analysed with the ARAMIS device.

8.4.1 Uniaxial tensile tests

The uniaxial tensile tests were carried out in specimens with the dimensions presented in Figure 8.5. The free edges of the specimens were machined in order to eliminate the hardened areas induced by the cutting and consequently to increase the range of homogeneous deformation.

Components of the strain tensor in the sheet plane are calculated by image correlation, considering the ARAMIS system.

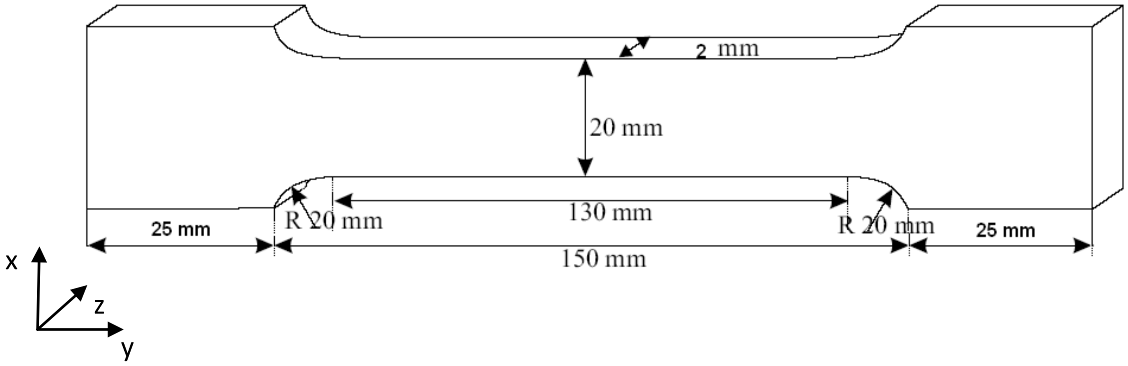


Figure 8.5 - Dimensions of the tensile specimen.

A set of monotonic tensile tests were performed along five different rolling directions (RD) in order to study the material anisotropy. The five directions are oriented by 0°, 22.5°, 45°, 67.5° and 90° respective of the RD. The universal tensile machine from the LIMATB, University of Bretagne-Sud, presented in Figure 8.6 was used to perform all the tests. In this figure the universal machine is adapted to the tensile tests. It is also possible to see in the image the two cameras used for the ARAMIS analyses.

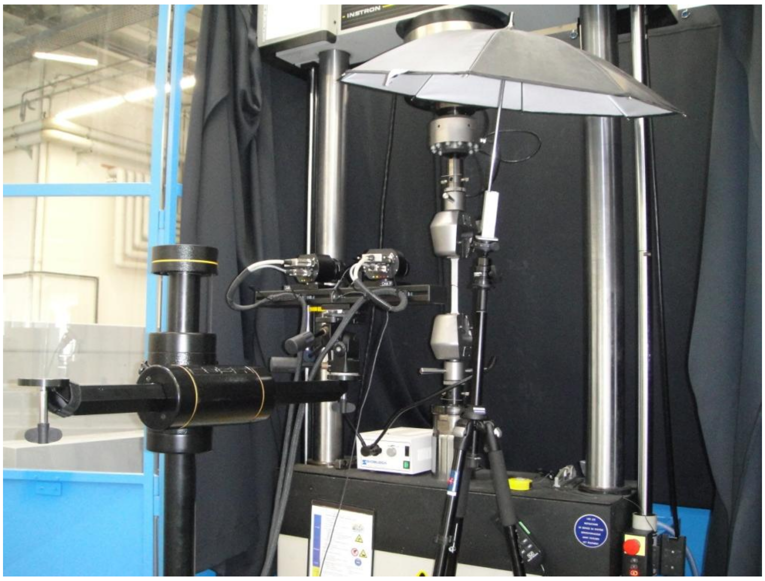


Figure 8.6 - Universal tensile machine from the LIMATB, University of South Brittany.

The test is controlled by the evolution of F_{yy} with time and by constraining $\sigma_{xx} = \sigma_{zz} = 0$. The Cauchy stress is calculated as $\sigma_{yy} =$ load/(actual section) and the nominal stress as σ_n =load/(initial section).

In Figure 8.7 it is presented the evolution of the Cauchy and the nominal stresses with the logarithmic strain for the five different rolling directions.

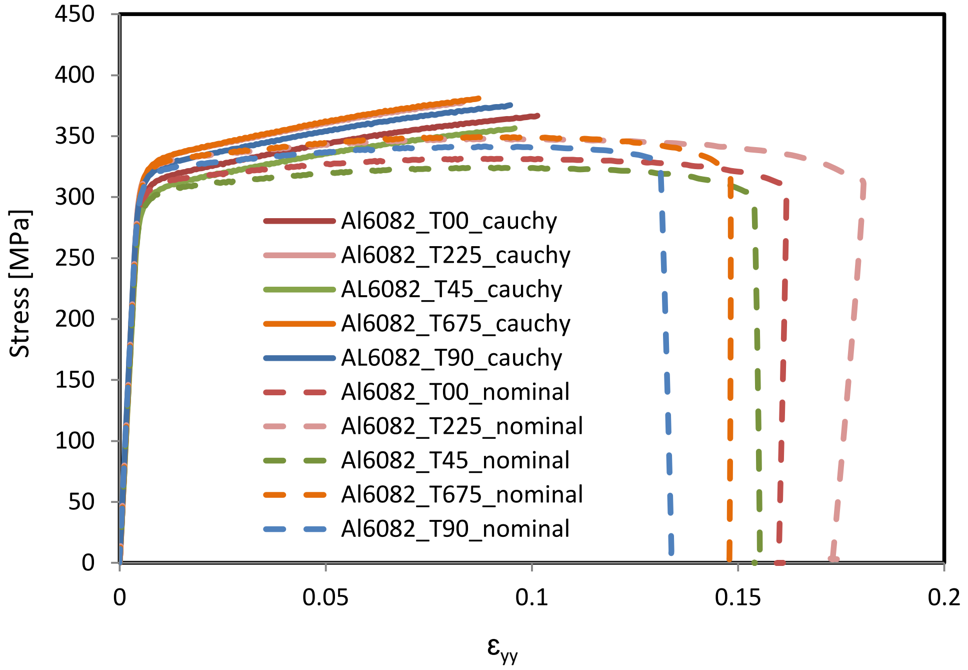


Figure 8.7 - Evolution of Cauchy and nominal stress with the strain for the five rolling directions.

In Figure 8.7 it is possible to observe that the rupture of the specimens occurred at different values of deformation. However was not verified a tendency of the deformation values with the angle. Also different yield stresses where obtained for the different tests.

In Table 8.3, it is shown the main mechanical properties obtained for the five rolling directions, where σ_m is the maximum value of the nominal stress, r_α are the anisotropy coefficients and are given by $r_\alpha = d\varepsilon_{xx}^p / d\varepsilon_{zz}^p$. The $\sigma_{0.2\%}$ is the nominal stress for a 0.2% of deformation. In the table are presented the means of these values considering the tests performed.

Table 8.3 - Summary of the main mechanical properties for AA6082.

Directions [°]	0	22,5	45	67,5	90
$\bar{\sigma}_m$ [MPa]	334,3±2,8	347,9±0,0	324,5±0,4	349,8±0,5	338,5±3,0
E [GPa]	60,07	70,5	62,02	67,7	69,16
$\bar{r}_\alpha$	0,535±0,005	0,605±0,005	0,695 ±0,015	0,645±0,015	0,51±0,0
$\dot{\varepsilon}$	0,00128	0,00133	0,00135	0.00130	0,00135
$\bar{\sigma}_{0.2\%}$	307,7±0,1	316,8±0,0	294,8±0,0	315,8±0,0	312,4±0,2

Considering the results obtained for the five rolling directions and the mechanical properties given in Table 8.2 it is possible to conclude that this material have similar properties to the properties presented for the aluminium alloy 6082 with T6/T651.

In Figure 8.8 it is presented a comparative study for the influence of the velocity of deformation in the evolution of the Cauchy stress. As can be seen, when different velocities are considered the hardening is not equal to the hardening for the tensile test made at constant velocity and equal to 10 mm/min. It is then possible to conclude that the Cauchy stress depends on the velocity of deformation.

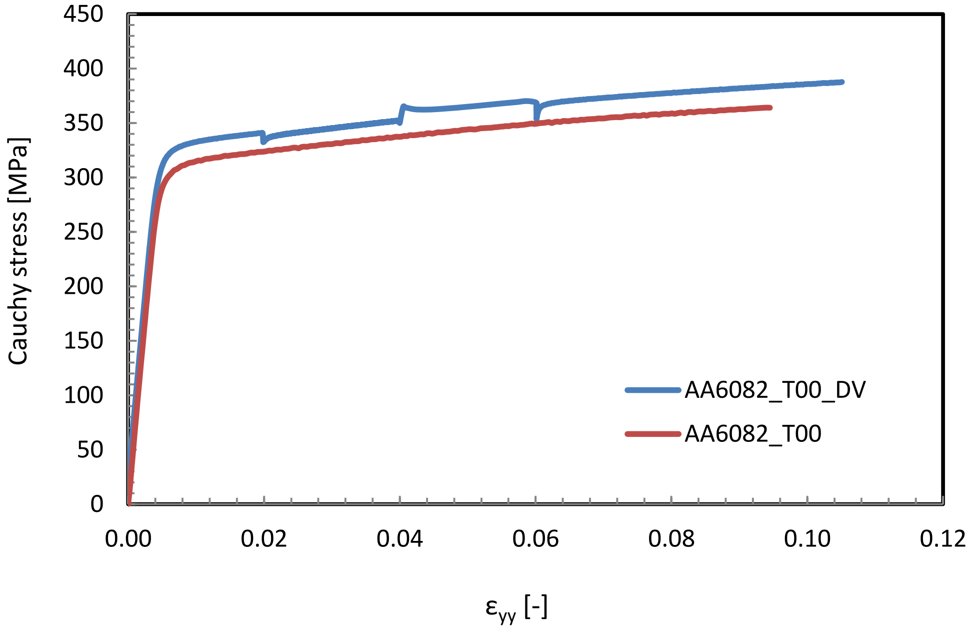


Figure 8.8 - Evolution of Cauchy stress with the strain for tests at 0°. T00 with v=10 mm/min and T00_DV with three different velocities, $v1$=1mm/min, $v2$=10 mm/min, $v3$=100 mm/min and $v4$=10mm/min.

8.4.2 Simple shear tests

The shear tests were performed in rectangular specimens with the dimensions represented in Figure 8.9.

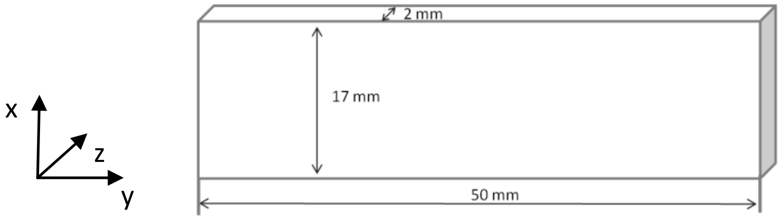


Figure 8.9 - Dimensions of the shear specimen.

A set of shear tests were performed in three different rolling directions (RD). The three directions are 0°, 45° and 90° of the RD. In Figure 8.10 it is possible to see the universal machine adapted to the shear tests.

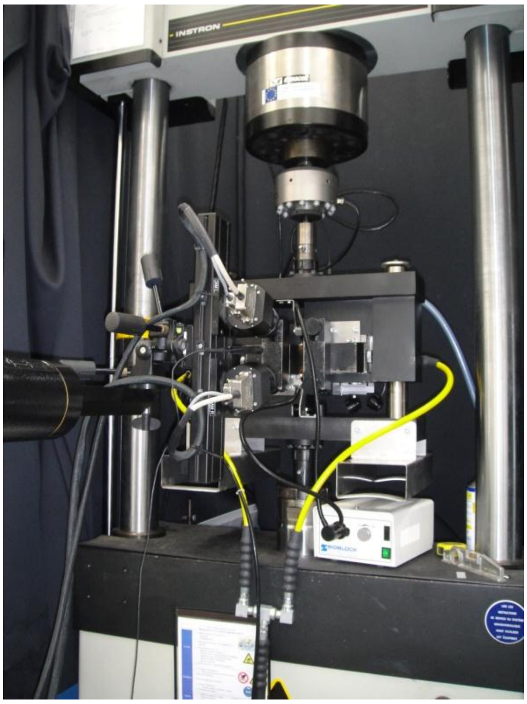


Figure 8.10 - Universal tensile machine adapted to shear tests from the LIMATB, University of South Brittany.

The shear device is presented in detail in [6]. The gauge area of h=4 mm is presented in Figure 8.11 and the shear direction is along the shear specimen.

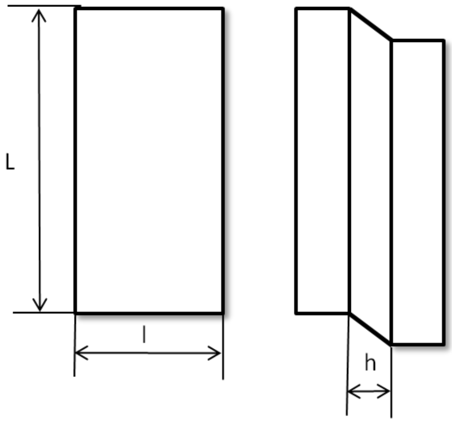


Figure 8.11 - Shear specimens before and after deformation, respectively.

In Figure 8.12 it is shown the evolution of the nominal stresses with the logarithmic strain for the three different rolling directions. It is important to remember that $\gamma_{12} = 2\varepsilon_{12}$.

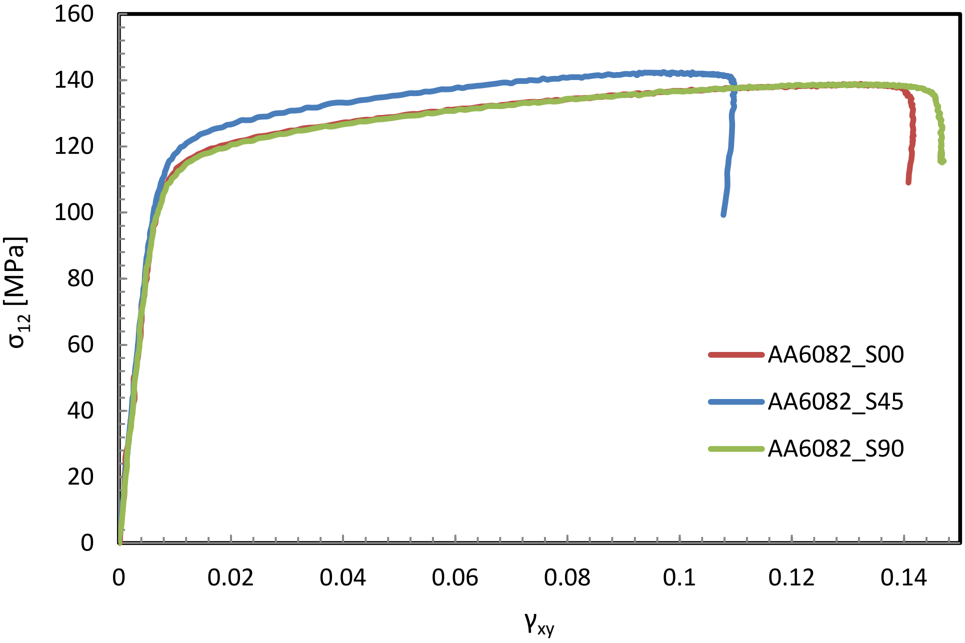


Figure 8.12 - Evolution of nominal stress with the strain for the three rolling directions.

Considering the results presented in Figure 8.12 for the 0° and 90° to the RD the results in terms of stress-strain evolution are approximate. However, the specimen at 45° to the RD has a light higher yield stress and the rupture happened at lower deformations when compared to the other two specimens.

In order to characterize the kinematic hardening of the AA6082 samples, two Baushinger tests at 0.05 and 0.1 of deformation for 0° of the RD were performed. In Figure 8.13 the results and the comparison with the shear test at 0° of the RD are presented.

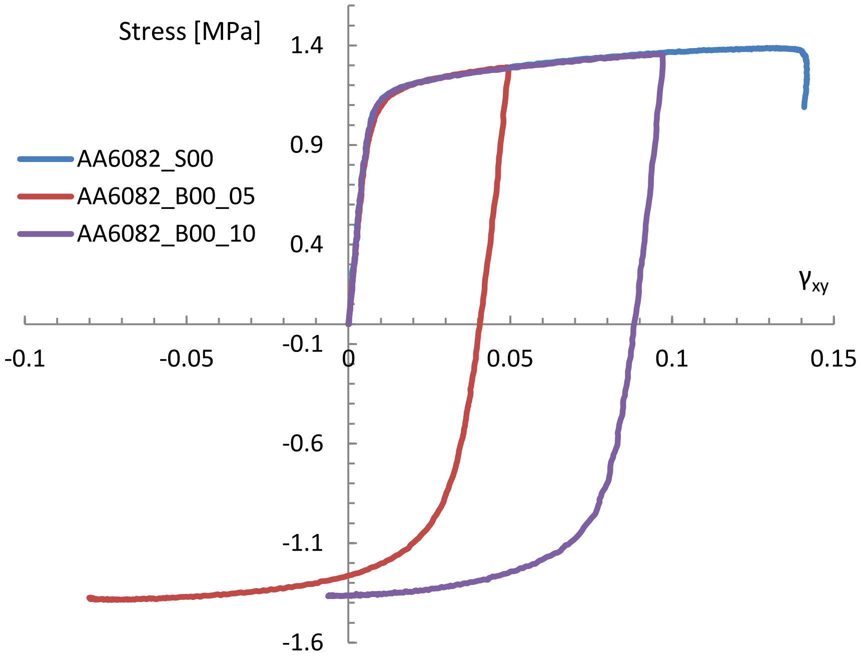


Figure 8.13 - Baushinger tests for 0.05% and 0.1% of deformation for the 0° of the RD.

The tests were performed only for 0.05 and 0.1 values of deformation because as it is possible to observe in Figure 8.13 the rupture for the shear test occurs approximately at 0.14 of deformation. For the initial stresses (without inversion of loading) the three curves are very similar. After the inversion of the loading the two curves, at 0.05 and 0.1, have shape very similar also.

8.4.3 Bulge tests

The bulge tests were performed in specimens with the dimensions represented in Figure 8.14.

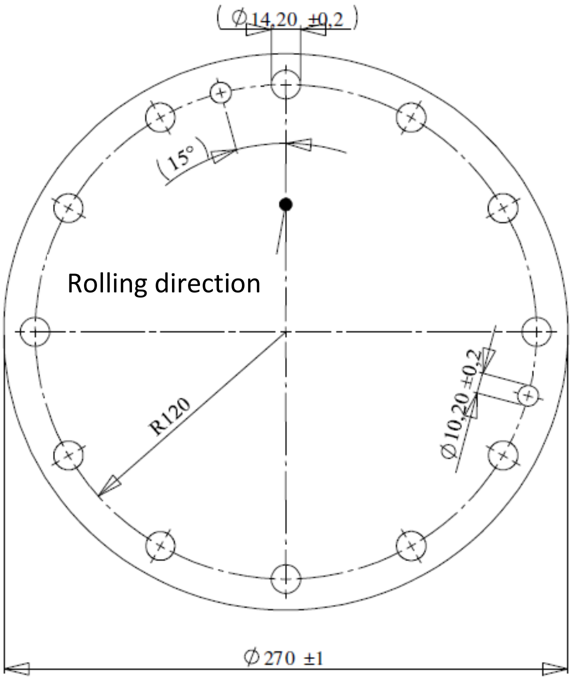


Figure 8.14 - Dimensions of the bulge specimen.

In Figure 8.15 it is possible to see the universal machine adapted to the bulge tests.

In the bulge test, the circular blanks are clamped by screws between a blank holder and a die and a fixed volume of water is pressed under the blank by the displacement of an actuator. The fluid pressure is given by a pressure sensor and the strain field is measured in the area around the center point by a digital correlation. During the test the strains $\varepsilon_{\mathrm{xx}}$ and $\varepsilon_{\mathrm{yy}}$ are recorded and as $\vec{e}_{\mathrm{x}}$ is parallel to the RD and $\vec{e}_{\mathrm{y}}$ is perpendicular in the sheet plane it can be shown that the two components are very close to each other, though only one is presented. In Figure 8.16 it is presented the evolution of the pressure with the logarithmic strain [7].

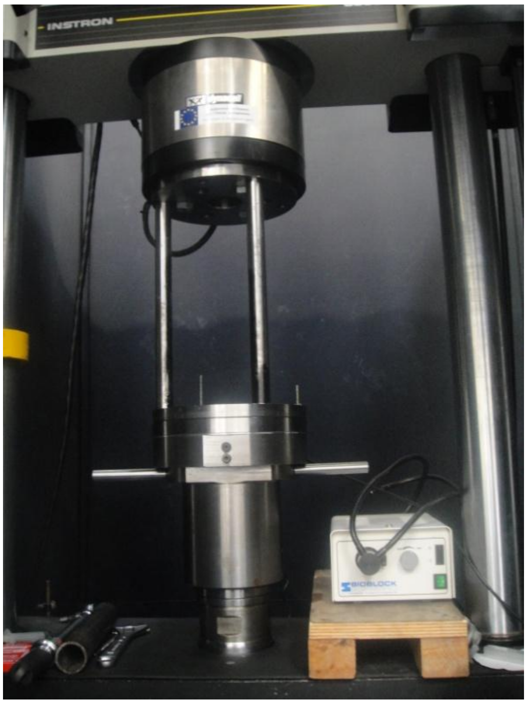


Figure 8.15 - Universal tensile machine adapted to bulge tests from the LIMATB, University of South Brittany.

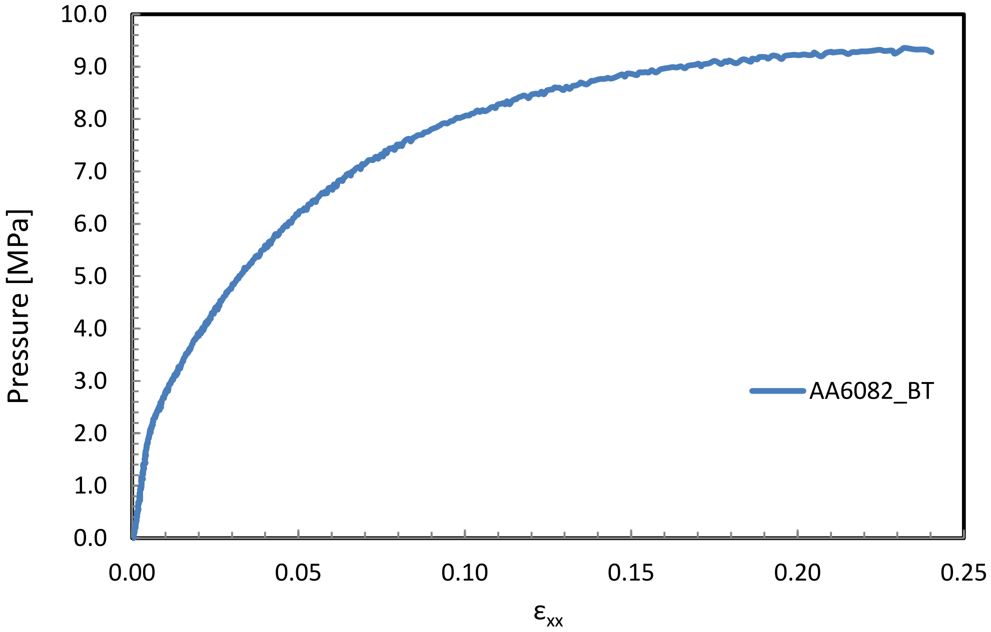


Figure 8.16 - Evolution of the pressure with the logarithmic strain.

8.4.4 Tensile results with ARAMIS system

In the present section it is studied the influence of considering the mean of deformation for the stress-strain curves (as it was considered in the previous sections) or consider the deformation for a specific point in the specimen. This study is performed for the tensile test. In Figure 8.17 it is possible to observe the deformation field obtained by the ARAMIS system during the tensile test.

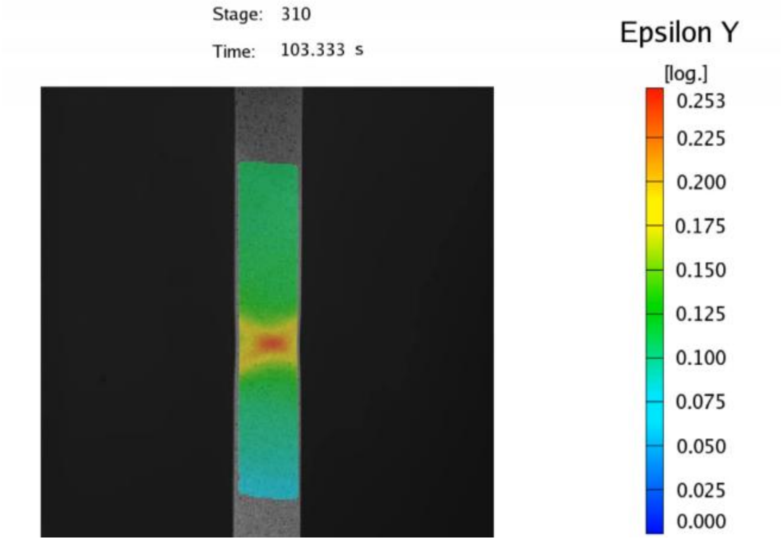


Figure 8.17 - Tensile results obtained with the ARAMIS software.

As it is possible to observe for the same time, different deformation values are registered for different zones of the specimen. In the previous results the deformation used was a mean where it is assumed that the entire specimen behaves homogeneously. As it is possible to observe in Figure 8.17 this simplification is not completely correct and it is important to understand what really is going on the specimen during the tensile test.

The localization of the four different points is shown in Figure 8.18 where US means upper side, BS means bottom side, CP1 and CP2 are respectively the central point 1 and 2.

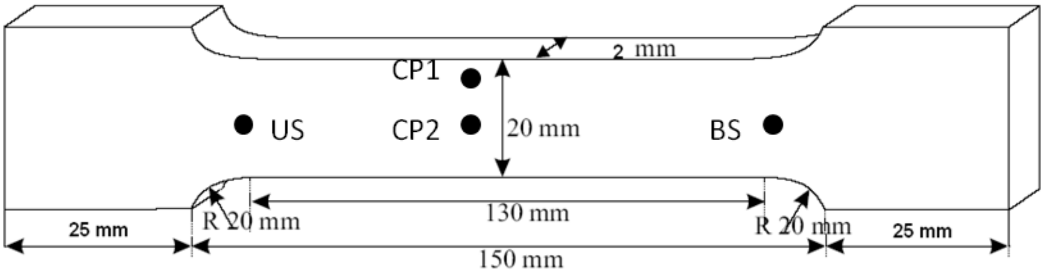


Figure 8.18 - Localization of the four different points for the ARAMIS results.

In Figure 8.19 it is represented the evolution of the Cauchy stress for four different points situated in the tensile specimen and possible to see that it is for the two central points that higher deformation values are achieved. Considering these results it is possible to conclude that in terms of stresses values obtained no difference was achieved being the stress-strain curves very similar. For parameter identification problems, in this case, it is indifferent which curve will be considered.

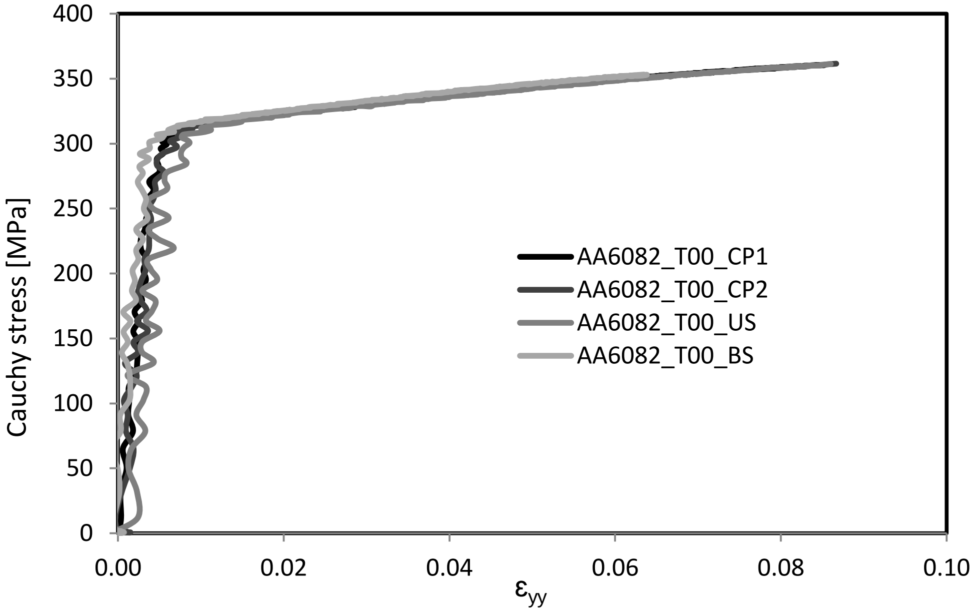


Figure 8.19 - Cauchy stress for the ARAMIS tensile results.

References

[1] Andrade-Campos A (2005) Numerical analysis and modelling of the mechanical and thermal behaviour of aluminium alloys. PhD thesis, University of Aveiro (in Portuguese) .

[2] Chaparro B (2006) Comportamento plástico de materiais metálicos: identificação e optimização de parâmetros. PhD thesis, University of Coimbra (in Portuguese) .

[3] ARAMIS User Manual Software v6.1 2009, gom.

[4] AALCO - The UK´s largest independent multi-metals stockholder (http://www.aalco.co.uk), 2012.

[5] Lanema - leader in plastics and aluminum of high technology, Lanema product catalog (http://www.lanema.pt), 2012.

[6] Thuillier S, Manach P (2009) Comparison of the work-hardening of metallic sheets using tensile and shear strain paths. International Journal of Plasticity 25:733–751.

[7] Zang S, Thuillier S, Port A, Manach P (2011) Prediction of anisotropy and hardening for metallic sheets in tension, simple shear and biaxial tension. International Journal of Mechanical Sciences 53:338–347.

Chapter 9

Optimization Strategies for Non-linear Material Parameters Identification

Parameter identification inverse problems are studied in order to achieve the best material parameters for specific constitutive models. A non-linear elastic-plastic hardening model, a hyperelastic model, and an elasto-viscoplastic model with isotropic and kinematic work-hardening were considered. Two different optimization algorithms were used: (i) the gradient-based Levenberg-Marquardt algorithm, and (ii) a real search-space evolutionary algorithm (EA). Strategies such as cascade, parallel and hybrid approaches are analysed in detail.

9.1 Introduction

As introduced in chapter 1, one category of inverse problems is called parameter identification. The main goal of these problems is the material parameters estimation for constitutive models. The parameters determination is performed confronting mathematical and experimental results, being the difference between them evaluated through a specific function, called the “objective function” [1-3].

Usually, there is no unique algorithm robust enough to deal with every possible mechanical problem [1]. In 1996, Patnaik [4] performed a study that analysed the performance of eight different optimization algorithms in the resolution of 41 different structural problems. The main conclusion of his work was that none of the eight optimization algorithms could successfully solve all the problems [4]. In order to alleviate the convergence difficulties, in 1997 Patnaik proposed the so-called “cascade optimization” strategy [5]. The cascade strategy uses several optimization methods, one followed by another, in a specific sequence, being the main aim of this strategy to take advantage of the strength of each optimization algorithm [1]. This cascade strategy is also used by Papadrakakis *et al.* [6-9], where consider that this approach is a very promising one, and proved to be robust and efficient. Patnaik *et al.* have other publication in this field, such as the references [10,11]. Ponthot *et al.* [1] studied eight different cascade approaches that combine: the conjugate gradient method, the BFGS, the Gauss-Newton, the Levenberg-Marquardt and the globally convergent method of moving asymptotes.

In previous works it was possible to find the combination of different deterministic algorithms, the combination of different evolutionary algorithms or even the combination of these two sort of algorithms.

In the present work, and in order to determine the best parameters set, optimization methods are applied. In this chapter, it is intended to develop and study new optimization strategies that efficiently can lead to accurate material parameters for different constitutive models, considering a-priori that these constitutive models are the relevant ones for the studied materials.

The constitutive models accounted for this study are a non-linear elastic-plastic hardening model, a hyperelastic model and an elasto-viscoplastic model with isotropic and kinematic work-hardening. The aim of these strategies is to take advantage of the strength of each selected algorithm and improve the overall robustness and efficiency of classical optimization methodologies based on single stages. Deterministic algorithms, evolutionary-inspired algorithms or even the combination of these two algorithms are used in the proposed strategies. Two different optimization algorithms are used: (i) the Levenberg-Marquardt algorithm, which is gradient-based, and (ii) a real search-space evolutionary algorithm (EA). Strategies such as cascade, parallel and hybrid approaches are analysed in detail. In hybrid strategies, cascade and parallel approaches are integrated.

9.2 Definition of the objective function

An accurate constitutive parameters determination requires the definition of the error function between experimental and numerical data. This error function is the objective function which will be subjected to optimization methods. As mentioned before, different optimization methods, such as gradient-based and evolutionary-based algorithms, have been applied in the study of this kind of objective functions. Therefore, a correct definition of this function is essential to all the optimization process and the determination of the constitutive parameters.

An ideal objective function must respect the following criteria [3,12]: (i) prior to the optimization process the experimental data should be filtered; (ii) all experimental data should be considered in the optimization as having equal opportunities to be optimized; (iii) in the multi-curves optimization, all experimental curves have equal opportunity to be optimized even if the number of experimental points is different for each curve; (iv) an objective function should be able to deal with multi-sub-objective problems. The units of the sub-objective problems can be different, but all of them should have an equal opportunity to be optimized. Different units and/or number of curves in each sub-objective should not affect the overall performance of the fitting and (v) the above criteria should automatically be respected without choosing weighting factors manually.

The experimental data used in this type of optimization process consists of several discrete values representing measured points, leading to an experimental curve. For the particular case of constitutive material models, being given by a set of (experimentally-based) stress-strain points, a stress-strain curve is obtained, which will be afterwards compared to the numerical constitutive curve (see Figure 9.1).

One of the most used error functions consists on the sum of the squares of the stress difference at different strain levels [2]. This gap between an experimental point and the corresponding point on the numerical curve can be seen in Figure 9.1.

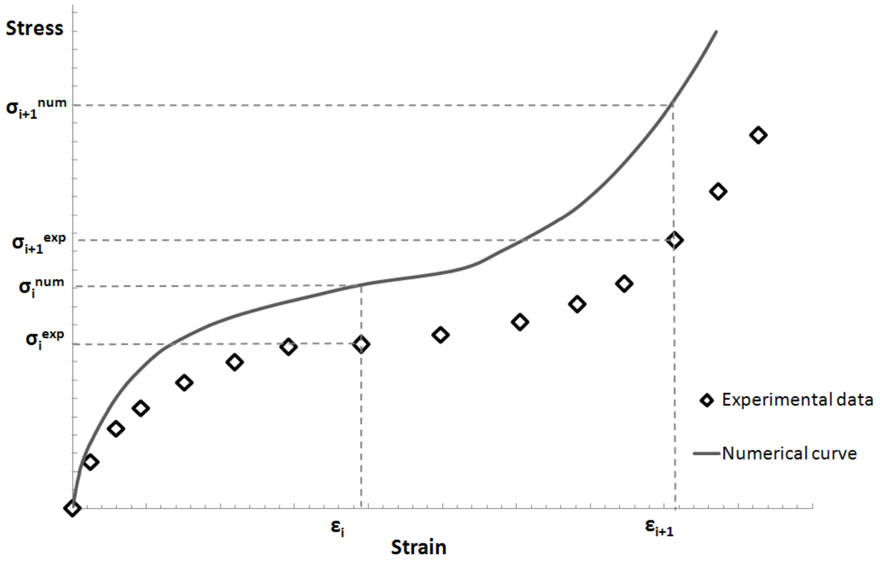


Figure 9.1 - Stress-strain numerical curve and experimental data.

Considering the parameters identification optimization problems, the objective function can be defined as

$$S_{\text{obj}}(\mathbf{x}) = \sum_{k=1}^{n_{\text{tests}}} \sum_{i=1}^{n_{\text{points}}} \left(\frac{\sigma_i^{\text{exp}} - \sigma_i^{\text{num}}(\mathbf{x})}{W_{\text{abs}} + W_{\text{rel}} \sigma_i^{\text{exp}}} \right)^2 , \tag{9.1}$$

where the numerator is the difference between the experimental and the numerical stress for the i-th value of strain, while W_{abs} and W_{rel} are weighting factors that must be adapted to the optimization problem in study. In this work this objective function was considered due to its simplicity, others objective function can also be used [13]. Analysing the objective function, it can be observed that if the correspondence between the physical experiment and the numerical model is perfect, then $S_{\text{obj}}(\mathbf{x})$ would be equal to zero. The reality is that this fact never happens and it is only expected that $S_{\text{obj}}(\mathbf{x})$ takes a low non-zero value [3]. The existence of local minimums also leads to practical difficulties in the interpretation and selection of the obtained results. Therefore, during the analysis of the results it is necessary to remember that it is possible to obtain several distinct sets of parameters for which the objective function assumes reasonable values. In these cases, it is the duty of the user to evaluate the obtained results considering the physical definition and the meaning of each parameter.

9.3 Constraints

The identification process of constitutive model parameters has to consider specific constraints to ensure a real physic behaviour of the materials. In the minimization problem, the constraints were considered using exterior penalties. Penalty method transforms constrained problems to unconstrained ones [14]. In this work, it is intended to minimize the objective function ($S_{\text{obj}}(\mathbf{x})$) subjected to constraints $g_i(\mathbf{x})$. Applying the penalty function $P(\mathbf{x})$ to the objective function, it will lead to a new unconstrained function. This penalty function will be activated only when the objective function needs to be constrained (active constraint).

An often-used class of penalty functions is:

$$P(\mathbf{x}) = \sum_{i=1}^{m} [\max|0, g_i(\mathbf{x})|]^q \text{ , where } q \geq 1. \tag{9.2}$$

It is noted that if $q = 1$, $P(\mathbf{x})$ is called the linear penalty function. This function may not be differentiable at points where $g_{\mathrm{i}}(\mathbf{x}) = 0$ for some i. Setting $q = 2$ is the most common in practice, being called the quadratic penalty function [15]. Therefore, in this work it will be used $q = 2$.

Afterwards, the initial problem is now the unconstrained minimization of $T(\mathbf{x})$. If the constrains are violated, meaning $g_i(\mathbf{x}) > 0$, a big term will be added to the $T(\mathbf{x})$ function such that the solution is pushed back towards the feasible region. Since the convergence is from the outside of the feasible region, these methods are called exterior penalty methods [14,16].

9.4 Constitutive models

In this section, three constitutive models, whose parameters will be determined with the developed optimization strategies, are briefly presented. These models were selected in order to represent distinct material types. It was considered a priori that these constitutive models are the relevant ones for the studied materials.

9.4.1 Non-linear elasto-plastic hardening model

The first application of the proposed optimization strategies is the parameters identification of a constitutive elastic-plastic model with non-linear hardening for a stainless steel AISI 304. Experimental tests were carried out in proportional loading for uniaxial tension, and the hardening law can be described by the following equation [1]:

$$\bar{\sigma}(\bar{\varepsilon}^{pl}) = \sigma^0 + (\sigma^\infty - \sigma^0)\left[1 - \exp(-\delta\bar{\varepsilon}^{\mathrm{pl}})\right] + \zeta\bar{\varepsilon}^{\mathrm{pl}} \,, \tag{9.3}$$

where $\bar{\boldsymbol{\varepsilon}}^{pl} = \sqrt{2/3\boldsymbol{\varepsilon}^{\mathrm{pl}}\boldsymbol{\varepsilon}^{\mathrm{pl}}}$ is the equivalent plastic strain and $\boldsymbol{\varepsilon}^{\mathrm{pl}}$ is the plastic strain tensor. Considering a one-dimensional analysis, this elastic-plastic model with non-linear hardening contains 5 parameters: the elastic parameter *E* and the plastic parameters: σ^0, σ^∞, δ and ζ.

This constitutive model leads to feasible stress-strain results only when $\sigma^0 < \sigma^\infty$. Therefore, during the formulation of the minimization problem, this constraint must be taken into account. For this constitutive model, it was chosen a weighting factor equal to 10 during the definition of the objective function. This assumption is based on the objective function values.

9.4.2 Hyperelastic model – Ogden model

The selected material to perform in this part of the present work was silicone-rubber. The experimental results used for the determination of the hyperelastic model parameters were obtained by Martins *et al.* [17] for this material, when it was subjected to uniaxial tension tests. Silicone-rubber belongs to the hyperelastic material category and exhibits strong nonlinear elastic behaviour for a high degree of deformation. In this study, it was assumed that silicone-rubber has a mechanical behaviour similar to incompressible hyperelastic materials [17].

The most applied models in the simulation of the hyperelastic materials behaviour are the Humphrey [18], Martins [19], Mooney-Rivlin [20,21], Neo-Hookean, Ogden [22], Veronda-Westman [23] and Yeoh [17] models [3]. Martins *et al.* [17] and Andrade-Campos *et al.* [3] have pointed out that the Ogden, Yeoh and Martins models are the ones that best fit the silicone-rubber experimental data. Considering these three hyperelastic models, the one which the parameter determination is more difficult is the Ogden model [22] due to its relatively high number of parameters [3]. In this work, it is intended to evaluate the performance of different optimization strategies. If these strategies efficiently lead to accurate parameters when applied to complex constitutive models, consequently they will work when applied to simple models. Taking into account these reasons, the Ogden model was considered.

The unidimensional Ogden stress function for incompressible isotropic elastic materials [22] is given by:

$$\sigma = \sum_{i=1}^{M} \mathrm{c}_{2i-1}\left(\lambda^{c_{2i}} - \lambda^{c_{2i}/2}\right), \tag{9.4}$$

which can easily be simplified in the equivalent equation (for $M = 4$)

$$\sigma = c_1\left(\lambda^{c_2} - \lambda^{-c_2/2}\right) + c_3\left(\lambda^{c_4} - \lambda^{-c_4/2}\right) + c_5\left(\lambda^{c_6} - \lambda^{-c_6/2}\right) + c_7\left(\lambda^{c_8} - \lambda^{-c_8/2}\right) \tag{9.5}$$

In equation 9.4 and 9.5, σ represents the uniaxial stress, $\lambda = 1 + \varepsilon$ is the stretch, while c_i are the Ogden´s material parameters.

In Ogden *et al.* [22], it is explained that for $M = 3$ the material parameters differ considerably for the different experimental tests. However, for $M = 4$, the difference between the tests is very small and this fact justifies the introduction of the fourth term, although there is no obvious physical basis for it [21]. This reason justified the use of the Ogden´s stress-deformation function for incompressible isotropic elastic materials with $M = 4$.

To be sure about the previous assumption, $M = 3$ and $M = 4$ models were studied. Although both models are very close to the experimental silicone-rubber data, it is possible to note that for *M*=4 the numerical model can fit substantially better the experimental data. This fact was expected due to the higher number of parameters. Superior values of *M* were not tested because they lead to an increase in the function complexity, without any gain for the material characterization.

Analytical formulations show that a constraint should be considered in order to get a realistic response. If this constrain is not taken into account, a good fit to the experimental values can be obtained but stress values can be negative for very low deformations levels, which is physically impossible.

To ensure a real response in a monotonic uniaxial tensile test, the stress σ should always be a non-negative value for all levels of stretch. Considering that the stress is zero when λ=1, then

$$\left.\frac{\mathrm{d}\sigma}{\mathrm{d}\lambda}\right|_{\lambda=1} \geq 0 \Rightarrow \sigma \geq 0 \text{ for } \forall\, \lambda \geq 1. \tag{9.6}$$

Applying these conditions to equation 9.5 the following constrain is obtained:

$$\mathrm{c}_1\mathrm{c}_2 + \mathrm{c}_3\mathrm{c}_4 + \mathrm{c}_5\mathrm{c}_6 + \mathrm{c}_7\mathrm{c}_8 \geq 0. \tag{9.7}$$

The constitutive models constrains were taken into account using a penalty function and for this constitutive model, a weighting factor equal to 0.01 was considered during the definition of the objective function.

9.4.3 Elasto-viscoplastic model with isotropic and kinematic work-hardening

For the elasto-viscoplastic constitutive model, the material studied was a hardening mild steel E220BH. The experimental data used was obtained by Thuillier *et al.* [24]. This data comprises experimental values of monotonous tensile and shear tests, both carried out at 0° to the RD (rolling direction), and of three tests performed in order to highlight the Bauschinger effect and to determine kinematic work-hardening parameters. The constitutive model was analysed by Thuillier *et al.* [24], and it is based on the elasto-plastic criterion of Hill [25] taking into account the kinematic work-hardening and the viscous character of the material.

The yield criterion takes into account the orthotropic symmetry of the sheet in its reference frame, with the axis $\vec{1}$ aligned with the RD, the axis $\vec{2}$ with the transverse direction (TD) and the axis $\vec{3}$ with the normal direction.

The yield criterion considered is given by:

$$f(\boldsymbol{\sigma}, \mathbf{X}, R) = \bar{\sigma} - R = \sqrt{\frac{3}{2}(\boldsymbol{\sigma}^{\mathrm{d}} - \mathbf{X})\!:\mathbf{M}\!:(\boldsymbol{\sigma}^{\mathrm{d}} - \mathbf{X})} - R, \tag{9.8}$$

where $\boldsymbol{\sigma}^{\mathrm{d}}$ represents the deviatoric part of $\boldsymbol{\sigma}$, $\bar{\sigma}$ is the equivalent stress and $\mathbf{M}$ is the matrix representation of the 4th order Hill's constitutive tensor. The 6 non-zero coefficients of $\mathbf{M}$ depend on the 6 coefficients F, G, H, L, M, N of the quadratic Hill's criterion. The condition on the initial elastic limit along the RD imposes the relation $G + H = 2$. The vectorial representation of the second-order tensor $\boldsymbol{X}$ represents the backstress and R the term of isotropic work-hardening.

The viscoplastic component of the strain follows a flow rule derived from a viscoplastic potential Ω, which is a power function of the yield function (Lemaitre and Chaboche [15]):

$$\Omega(f) = \frac{K^{\nu}}{n^{\nu}+1}\left(\frac{f^{+}}{K^{\nu}}\right)^{n^{\nu}+1}, \tag{9.9}$$

where n^{v} is the strain rate sensitivity coefficient, K^{v} a weighting coefficient of the viscous part of the stress and f^{+} the positive part of f. The behavior is thus elastic if $f < 0$ and if $f > 0$ the viscoplastic strain rate is written as:

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}} = \frac{\partial \Omega}{\partial \boldsymbol{\sigma}} = \Omega'(f)\frac{\partial f}{\partial \boldsymbol{\sigma}}, \tag{9.10}$$

in order to obtain $f = 0$ for plastic deformation.

The equivalent viscoplastic strain rate $\dot{\varepsilon}_{\mathrm{eq}}^{\mathrm{vp}}$ is defined from the plastic work conservation principle:

$$\dot{\varepsilon}_{\mathrm{eq}}^{\mathrm{vp}} = \frac{\left(\boldsymbol{\sigma}^{\mathrm{d}} - \boldsymbol{X}\right) : \dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}}}{\bar{\sigma}}\,. \tag{9.11}$$

The work-hardening combines isotropic and kinematic contributions, and the evolution of the isotropic work-hardening is related to the accumulated plastic strain following the swift law, *i.e.*

$$R = K(\varepsilon_{\mathrm{eq}}^{\mathrm{vp}} + \varepsilon_0)^{n} \quad \text{with} \quad \varepsilon_0 = \left(\frac{\sigma_0}{K}\right)^{1/n}, \tag{9.12}$$

where K is a material parameter, n the hardening coefficient and σ_0 is the initial yield stress in tension along the RD.

The non-linear evolution law of the kinematic work-hardening is based in the Armstrong and Frederick (1966) [26] law with a linear component of Prager type:

$$\boldsymbol{X} = \frac{2}{3}Q_x\boldsymbol{\alpha} + \frac{2}{3}H_{\mathrm{p}}\boldsymbol{\varepsilon}^{\mathrm{vp}} \quad \text{with} \quad \dot{\boldsymbol{\alpha}} = \dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}} - b_x\dot{\varepsilon}_{\mathrm{eq}}^{\mathrm{vp}}\boldsymbol{\alpha}, \tag{9.13}$$

where $\boldsymbol{\alpha}$ is the internal variable associated to $\mathbf{X}$, Q_x/b_x determines the intensity of the non-linear kinematic work-hardening and H_p is the slope of the linear kinematic work-hardening [24].

These models were implemented considering the integration methods explained in section 7.4.

9.5 Optimization strategies

There is no unique algorithm robust enough to deal with every optimization problems and guarantee without any doubts that the global minimum is reached. Therefore, if different optimization methods can be used and combined, it is possible to take advantage of the strength of each algorithm and increase the probability to reach the global minimum.

The gradient-based methods are objective and have a low computational cost. Nevertheless, the performance of these methods is highly dependent on the starting point and, facing non-convex functions, they have high probability to converge to local minimum. On the other hand, evolutionary algorithms have proved to be independent of the initial point and have a large probability to find the global minimum. However, these advantages are reached with the increasing of the CPU time and the number of objective function evaluations. Analysing the features of these two algorithms it seems that these have opposing strengths and weaknesses. Consequently, these have a high potential to be used in combined optimization strategies [3].

The two algorithms used in this work were the Levenberg-Marquardt algorithm (LM) and an Evolutionary Algorithm (EA), mentioned previously. The strategies studied here combine algorithms in cascade, parallel and hybrid arrangements.

The criteria chosen to switch from one optimizer to another were: (i) if from one iteration to another the relative decrease of the objective function is less than $1x10^{-15}$, or (ii) the maximum admissible iteration number (which is a predetermined value) is reached.

9.5.1 Cascade strategies

The cascade strategy is a multi-stage procedure where various optimizers are activated one after another in a pre-specified sequence. Four cascade strategies were studied, such as the LM+EA, the EA+LM, the LM+EA+LM and the EA+LM+EA strategy. The use of single-stages optimizers was performed and compared to the cascade strategies. These strategies were applied to the constitutive models presented previously.

9.5.2 Parallel strategies

Through the use of parallel strategies it is possible to transform the single-stages strategies in new strategies where different starting points can be considered. In Figure 9.2 the single-stage and the parallel strategies are represented. The starting point, **x**, represents the parameters set and *n* is the number of different starting points. The parameters set are generated randomly.

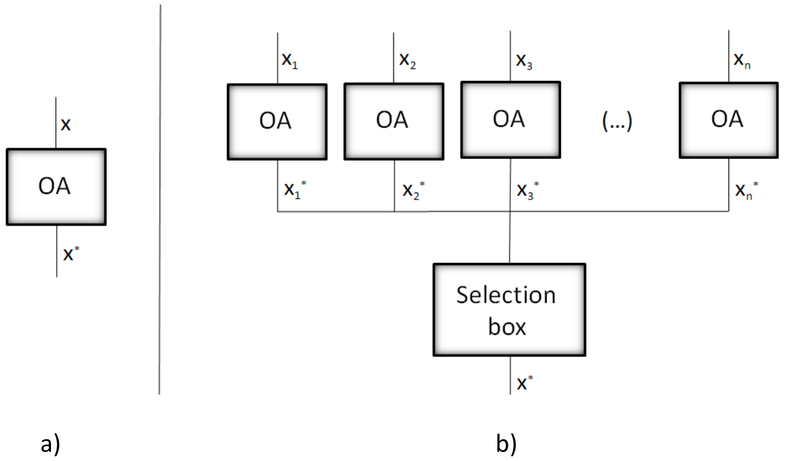


Figure 9.2 - Optimization strategies: a) single-stage and b) parallel strategy.

For each one of these starting points an optimization algorithm (OA) is applied, and **x*** is the optimized parameters set. In the parallel strategy, each optimized parameters set enters in the selection box and the best solution is chosen based on the achieved value of the objective function.

The comparison between single and parallel strategies was performed considering just the Levenberg-Marquardt algorithm. The gradient-based algorithms have a unique starting point, leading to the starting point dependency. With the parallel strategy it is possible to transform the single stages strategies for the gradient-based algorithms in algorithms less dependent on the starting point, acquiring “probabilistic features”. This is called “multistart strategy”.

It was considered that the improvements of a parallel strategy, for probabilistic optimization algorithms such as evolutionary algorithms, are not significant. The evolutionary algorithm has intrinsic parallel and probabilistic features that make this algorithm to be independent of the starting point.

In the parallel strategies studied here, eight starting points were considered. During their generation, the model constrains are taken in account.

9.5.3 Hybrid strategies

The hybrid strategies combine both potentialities of the cascade and the parallel strategies. In Figure 9.3 it is possible to observe the strategies studied. For the strategies *a*, *b*, *c* and *g* the starting point is the same as the one considered for the cascade strategies. Therefore the results and the analysis performed are related to this starting point. For the others strategies the starting set was obtained randomly. In Figure 9.3, it is possible to observe the two critical stages where the objective function is analyzed, also listed in Table 9.4. All the decisions relatively to the optimization parameters only take into account the objective function value.

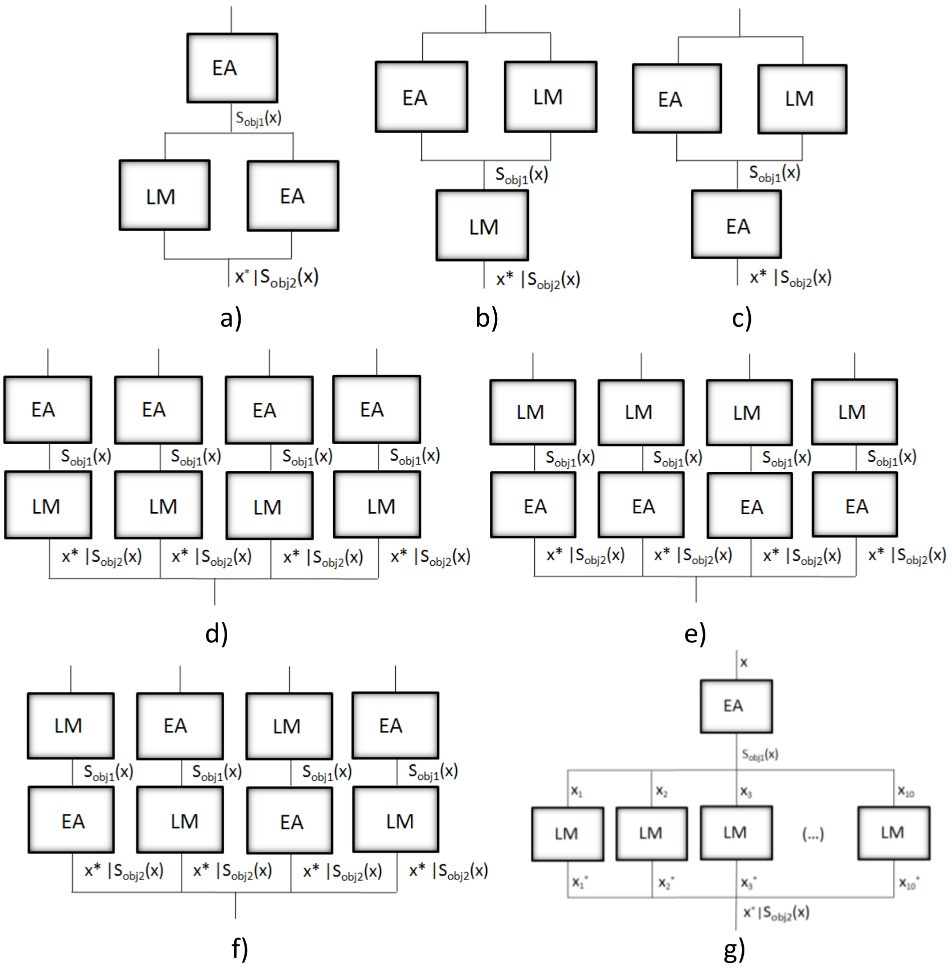

EA
Sobj1(x)
LM
EA
x* |Sobj2(x)
a)
EA
LM
Sobj1(x)
LM
x* |Sobj2(x)
b)
EA
LM
Sobj1(x)
EA
x* |Sobj2(x)
c)
EA EA EA EA
Sobj1(x) Sobj1(x) Sobj1(x) Sobj1(x)
LM LM LM LM
x* |Sobj2(x) x* |Sobj2(x) x* |Sobj2(x) x* |Sobj2(x)
d)
LM LM LM LM
Sobj1(x) Sobj1(x) Sobj1(x) Sobj1(x)
EA EA EA EA
x* |Sobj2(x) x* |Sobj2(x) x* |Sobj2(x) x* |Sobj2(x)
e)
LM EA LM EA
Sobj1(x) Sobj1(x) Sobj1(x) Sobj1(x)
EA LM EA LM
x* |Sobj2(x) x* |Sobj2(x) x* |Sobj2(x) x* |Sobj2(x)
f)
x
EA
Sobj1(x)
x1 x2 x3 x10
LM LM LM (...) LM
x1* x2* x3* x10*
x* |Sobj2(x)
g)

Figure 9.3 - Hybrid strategies: a) EA followed by a parallel strategy between LM and EA, b) parallel strategy between EA and LM followed by LM algorithm, c) parallel strategy between EA and LM followed by EA algorithm, d) parallel strategy between four cascade strategies (EA+LM), e) parallel strategy between four cascade strategies (LM+EA), f) parallel strategy between two cascade strategies (LM+EA) and two cascade strategies (EA+LM) and g) EA followed by a parallel strategy between ten LM algorithms.

9.6 Results and discussion

9.6.1. Cascade strategies

The results for the non-linear elasto-plastic hardening model and for the Ogden model, obtained with the different cascade strategies, are presented in Table 9.1. A maximum number of 3000 generations, with a population size of 10, was chosen for the EA method when used as the first optimization algorithm and, when used in the middle or in the end of a strategy, 1000 generations, with a population size of 10, were performed. The evaluations are also presented in Table 9.1. As an example, for the case of the EA+LM+EA strategy, for the hardening model, are registered 3000+52+1000. This means that the first EA algorithm made 3000 evaluations, followed by the LM algorithm that had made 52 evaluations and finished with an EA algorithm with 1000 generations.

Table 9.1 - Initial and optimal values for the hardening and the hyperelastic model parameters identification.

Constitutive model		Hardening model	Hyperelastic model
Parameters		$\{ \sigma^0, \sigma^\infty , \delta , \zeta \}^\mathrm{T}$	$\{ c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8 \}^\mathrm{T}$
Initial values (x_i)		{ 400, 700 , 20 , 400, 300000 }$^\mathrm{T}$	{ 0.8196, 0.0717 , 1.7192 , 0.5399, 2.3369, 2.6695, 1.2160, -0.8551 }$^\mathrm{T}$
Boundaries		$[\sigma^0_{\mathrm{min}}, \sigma^0_{\mathrm{max}}]$ [250,350] $[\sigma^\infty_{\mathrm{min}}, \sigma^\infty_{\mathrm{max}}]$ [600,1000] $[\delta_{\mathrm{min}}, \delta_{\mathrm{max}}]$ [2,20] $[\zeta_{\mathrm{min}}, \zeta_{\mathrm{max}}]$ [300,1000] $[E_{\mathrm{min}}, E_{\mathrm{max}}]$ [250000,450000]	$[c_i^{\mathrm{min}}, c_i^{\mathrm{max}}]$ [-5,5]
LM	$S_{\mathrm{obj}}(\mathbf{x})$ Evaluations $\mathbf{x}^*$	1.2761 60 { 316.9, 731.0, 5.8, 682.9, 381740 }$^\mathrm{T}$	16.3224 175 {0.8481, 0.0282, 3.5771, 0.3479, 0.1603, 4.1244, 0.7438, -1.2811 }$^\mathrm{T}$
EA	$S_{\mathrm{obj}}(\mathbf{x})$ Evaluations $\mathbf{x}^*$	2.2199 3000 { 318.12, 844.8, 4.9, 470.3, 300000 }$^\mathrm{T}$	13. 5207 3000 { 0.6149, 3.1860, -0.7421, 2.0073, 2.9125, 0.2508, 0.3045, -0.9093 }$^\mathrm{T}$
LM + EA	$S_{\mathrm{obj}}(\mathbf{x})$ Evaluations $\mathbf{x}^*$	1.2759 60+1000 { 316.9, 730.3, 5.8, 683.4, 381736 }$^\mathrm{T}$	7.0841 175+1000 { 0.6149, 0.0042, 3.8672, 0.3675, 0.1529, 4.2059, 0.9647, -1.2397 }$^\mathrm{T}$
EA + LM	$S_{\mathrm{obj}}(\mathbf{x})$ Evaluations $\mathbf{x}^*$	1.3009 3000+52 { 318.0, 881.5, 4.6, 410.5, 380420 }$^\mathrm{T}$	13.3277 3000+10 { 0.6149, 3.1860, -0.7421, 2.0073, 2.9125, 0.2508, 0.3060, -0.9138 }$^\mathrm{T}$
LM + EA + LM	$S_{\mathrm{obj}}(\mathbf{x})$ Evaluations $\mathbf{x}^*$	1.2759 46+1000+7 { 316.9, 730.3, 5.8, 683.4, 381736 }$^\mathrm{T}$	6.6565 175+1000+10 { 0.5523, 0.0042, 3.8863, 0.3675, 0.1529, 4.2059, 0.9647, -1.2397 }$^\mathrm{T}$
EA + LM + EA	$S_{\mathrm{obj}}(\mathbf{x})$ Evaluations $\mathbf{x}^*$	1.3008 3000+52+1000 { 318.1, 881.5, 4.6, 411.1, 380420 }$^\mathrm{T}$	10.4211 3000+10+1000 { 0.6679, 3.1033, -0.8250, 1.8086, 2.6794, 0.0921, 0.0007, -0.9732 }$^\mathrm{T}$

From the results presented, it is possible to conclude that for the hardening model all the strategies lead to satisfactory results and similar final parameters. As can be seen from these results, the LM method, when used separately, is more efficient than the EA. This efficiency is evaluated through the computational quality/cost relation (the computational cost is proportional to the number of iterations and generations). In fact, the LM method needs only 60 iterations (equivalent to the 60 evaluations of the objective function) versus the 3000 EA generations, with a population size of 10, and reach to a lower value of the objective function. This significant difference can be attributed to the size of the EA search universe. In the LM+EA strategy the results were better than for the single LM strategy but the numerical effort does not reward such improvement. The LM+EA+LM strategy does not bring any improvement comparatively with the LM+EA strategy.

For the strategies where EA method is used in first place, it is possible to conclude that the introduction of the LM method in cascade leads to better values in terms of objective function values. The strategy EA+LM+EA is not considered to be advantageous because it leads to the increasing of the evaluations without a significant improvement of the objective function.

It is possible to conclude that for the parameters determination of the considered non-linear elastic-plastic hardening model the LM method is more efficient than EA and a cascade strategy turns out not to be necessary to identify the parameters. In Figure 9.4 it is possible to observe the evolution of the objective function for the LM method. It is possible to observe that it is in the first 20 iterations that the objective function value has the bigger decrease, being this value in the next iterations near the optimum value reached.

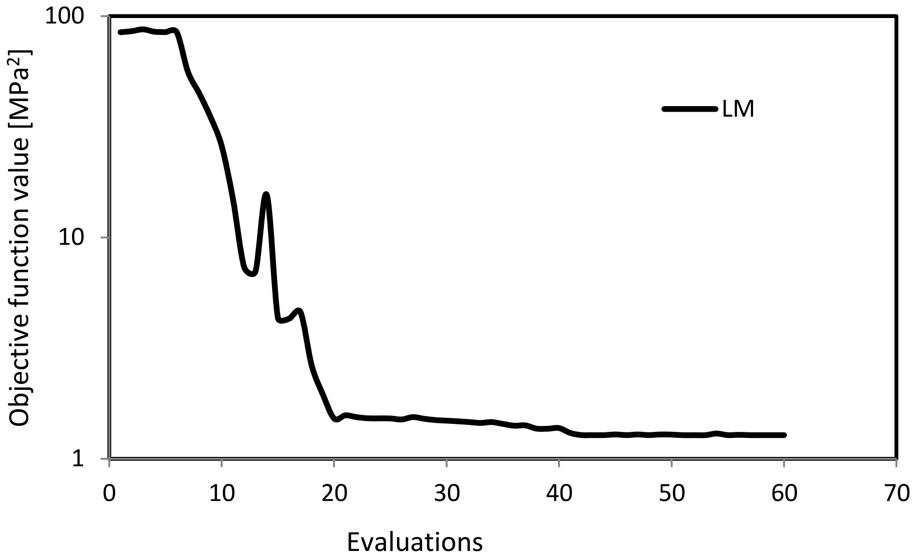


Figure 9.4 - Objective function evolution with the evaluations number of the better result of the cascade strategy for the hardening model.

For the hyperelastic model, the strategies also lead to satisfactory results for the same initial parameters set, but reaching completely different parameters. Considering this fact, it is possible to conclude that this constitutive model has a numerous number of local minimums, thus the next conclusions refer specifically to the starting set used. As can be seen from these results, when used separately, the EA method leads to better values of the objective function. However, this improvement required a higher numerical effort. This was expected because this function has a large number of local minimums. Thus, a probabilistic method will lead, with greater probability, to a lower local minimum or to the global minimum. In the LM+EA strategy a considerable improvement was observed, relatively to the single LM method, what derives from the reason stated before. The LM+EA+LM strategy improves the results from the LM+EA, because probably the LM+EA reach to a point very close to a local/global minimum and a gradient method is perfect to achieve it. For the strategies where EA method is firstly used it is possible to conclude that the introduction of the LM method in cascade does not lead to any improvement. For the EA+LM+EA strategy an improvement is registered in spite of being added a probabilistic method, which is ideal for this function.

It is possible to conclude that, for the parameters determination of the considered hyperelastic model, cascade strategies, such as the LM+EA and the LM+EA+LM approaches, achieve better material parameters[1] than the single-stages methods. In Figure 9.5 it is presented the evolution of the objective function value with the evaluations number for the LM+EA+LM strategy. Analysing the graph it is possible to conclude that is the first LM method that have the higher decrease in terms of objective function value. The following EA method also leads to a decrease of the objective function and the decrease obtained with the last LM method is very small.

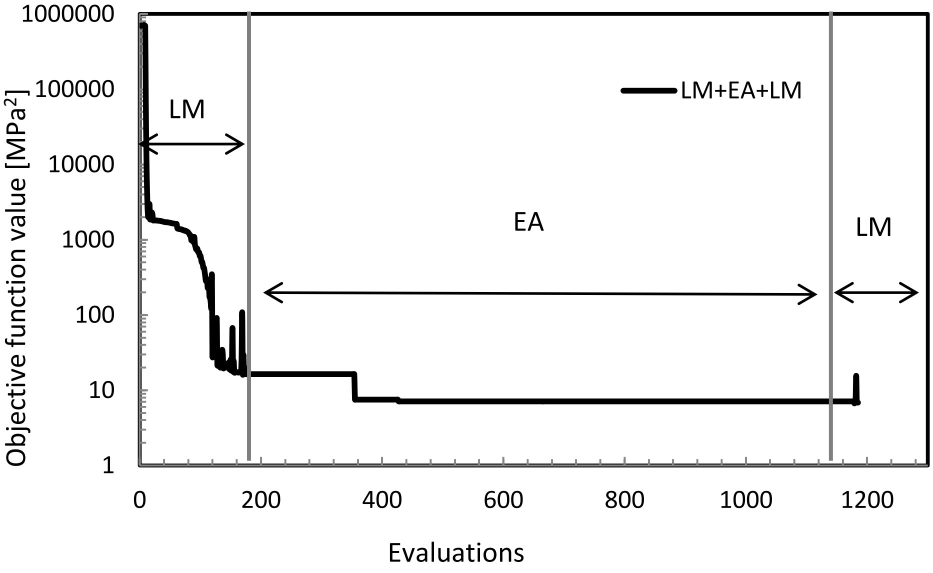


Figure 9.5 - Objective function evolution with the evaluations number of the better result of the cascade strategies for the hyperelastic model.

9.6.2. Parallel strategies

For both models, it was considered the same boundaries mentioned in Table 9.1. For each strategy (1-single-stage and 2 -parallel strategy), the starting points, the respective optimized parameters and the objective function values are represented in Table 9.2, for the hardening model, and in Table 9.3 for the hyperelastic model. From the results presented in Table 9.2, it is possible to conclude that all strategies lead to satisfactory results and similar final parameters for the hardening model. Different values were achieved for the objective function. From these results it is rather difficult to predict if the global minimum is inserted in the set of optimum solutions obtained.

It is within this context that the parallel strategies assume an important role. In the single stage strategies only one starting point exists, that leads to only one set of optimized parameters. Instead of this, the parallel algorithm uses a number of starting points defined by the user and

[1] Better material parameters in the sense of achieving a minor objective function.

with the increasing of n the possibility to reach the global minimum, or even to obtain a local minimum lower, increases.

Table 9.2 - Initial and optimal values for the hardening model.

Strat.	$\mathbf{x}$	σ^0	σ^∞	δ	ζ	E	$S_{\mathrm{obj}}(\mathbf{x})$
1	$\mathbf{x}_1$	299.4	962.4	3.8	321.5	283960	---
	$\mathbf{x}_1^*$	318.9	966.9	4.1	283.0	380680	1.3132
2	$\mathbf{x}_1$	279.1	725.5	2.9	775.6	384360	---
	$\mathbf{x}_1^*$	316.7	707.7	6.0	744.3	382480	1.2755
	$\mathbf{x}_2$	319.7	686.1	4.8	494.6	349320	---
	$\mathbf{x}_2^*$	316.9	761.1	5.5	622.8	380640	1.2815
	$\mathbf{x}_3$	342.3	836.5	12.7	961.4	305600	---
	$\mathbf{x}_3^*$	314.1	585.3	8.5	999.9	382110	1.2642
	$\mathbf{x}_4$	312.5	644.4	2.6	407.8	446950	---
	$\mathbf{x}_4^*$	317.4	702.3	6.1	746.9	382900	1.2750
	$\mathbf{x}_5$	275.9	921.3	4.8	547.5	404260	---
	$\mathbf{x}_5^*$	317.5	820.2	4.9	517.3	381970	1.2907
	$\mathbf{x}_6$	271.6	997.0	19.4	986.4	370110	---
	$\mathbf{x}_6^*$	314.4	615.4	7.6	927.9	382070	1.2626
	$\mathbf{x}_7$	340.9	707.6	2.6	718.3	338860	---
	$\mathbf{x}_7^*$	317.3	741.2	5.6	668.1	382080	1.2794
	$\mathbf{x}_8$	400.0	700.0	20.0	400.0	300000	---
	$\mathbf{x}_8^*$	316.6	731.3	5.7	682.5	382020	1.2761

In Figure 9.6 it is presented the evolution of the objective function value for the better LM result within the 8 parallel LM methods considered. The higher decrease in terms of objective function is during the 38 first iterations. The other iterations are really closer to the optimum value reached.

Table 9.3 - Initial and optimal values for the hyperelastic model.

Strat.	$\mathbf{x}$	c_1	c2	c_3	c_4	c_5	c_6	c_7	c_8	$S_{\text{obj}}(\mathbf{x})$
		[MPa^2]	[ ]	[MPa^2]	[ ]	[MPa^2]	[ ]	[MPa^2]	[ ]	[MPa^2]
1	$\mathbf{x}_1$	-0.989	-1.417	-1.907	2.089	1.971	2.920	-2.083	1.029	---
	$\mathbf{x}_1^*$	-0.884	-1.373	-1.949	1.909	2.186	2.469	-1.947	1.031	13.30
2	$\mathbf{x}_1$	0.921	-1.523	0.437	-0.275	0.884	0.838	0.638	1.327	---
	$\mathbf{x}_1^*$	1.347	0.541	-0.174	0.501	-1.490	0.721	0.367	3.442	9.864
	$\mathbf{x}_2$	2.370	1.002	-0.041	1.027	-0.059	0.367	1.608	-0.692	---
	$\mathbf{x}_2^*$	0.251	3.708	-0.123	-0.166	0.126	-0.439	-0.045	-2.801	22.93
	$\mathbf{x}_3$	-2.957	-1.100	2.307	-0.754	0.734	-1.678	-1.993	-2.003	---
	$\mathbf{x}_3^*$	-4.695	0.496	0.124	-0.885	1.399	-3.034	-1.879	-4.819	22.51
	$\mathbf{x}_4$	-2.270	1.097	-0.116	2.464	2.736	2.533	1.253	1.969	---
	$\mathbf{x}_4^*$	-3.568	1.617	-0.189	2.481	2.069	1.519	1.435	2.661	16.71
	$\mathbf{x}_5$	-0.503	-2.851	-0.223	-2.592	2.761	1.369	-2.715	-0.249	---
	$\mathbf{x}_5^*$	-1.427	-0.336	-0.096	-1.803	0.177	4.066	2.102	-0.118	17.91
	$\mathbf{x}_6$	-1.499	0.961	-1.794	-2.963	-0.956	1.808	2.095	-0.826	---
	$\mathbf{x}_6^*$	-0.001	-0.648	-1.305	-4.981	-1.656	-1.256	3.969	-1.668	14.01
	$\mathbf{x}_7$	-0.351	0.028	0.441	2.211	1.207	0.138	-0.494	2.195	---
	$\mathbf{x}_7^*$	-0.002	0.036	0.257	3.789	1.508	0.107	-0.121	1.738	6.356
	$\mathbf{x}_8$	1.359	-1.116	-2.423	-2.616	-1.812	2.063	0.565	-1.860	---
	$\mathbf{x}_8^*$	1.456	-2.784	-1.881	-4.855	-3.543	-0.099	1.102	-3.393	12.63

As it can be observed in Table 9.3, for each starting point a different optimum solution is achieved for the hyperelastic model. This fact is related with the high number of local minimums. Considering these results and the benefits of the parallel strategies, it is possible to conclude that for the optimization of functions with a high number of local minimums this strategy conducts to good results. According to the results, the parallel strategy leads to better results than the single stage strategy reaching a lower objective function value. Therefore, the parameters reached by the parallel strategy fits better the experimental hyperelastic model for the silicone-rubber than the ones reached by the single stage strategy. In Figure 9.7 it is presented the evolution of the objective function value with the number of iterations/generations for the better LM result within the 8 parallel LM methods considered. It is possible to conclude that the higher decrease in terms of objective function occurs during the first 90 iterations.

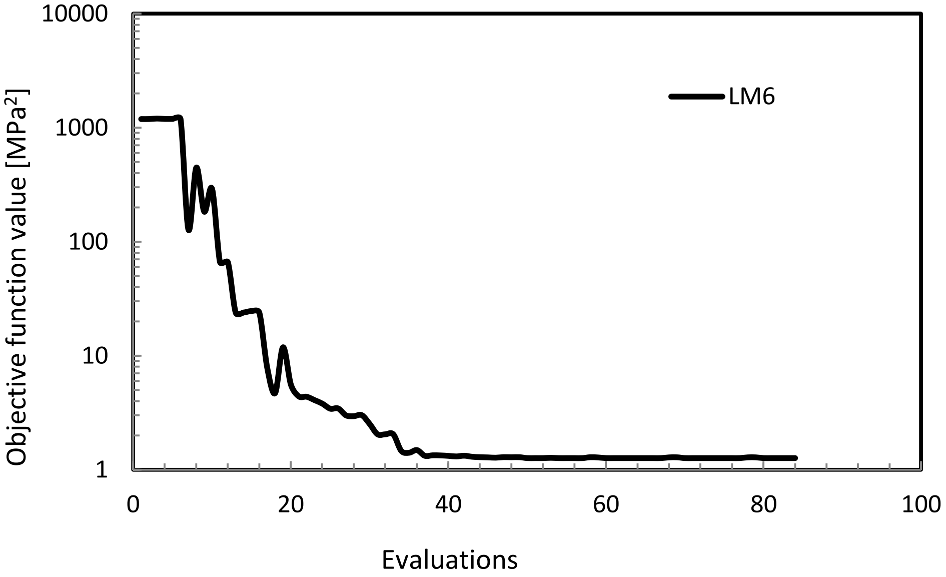


Figure 9.6 - Objective function evolution with the evaluations number of the better result of the parallel strategies for the hardening model.

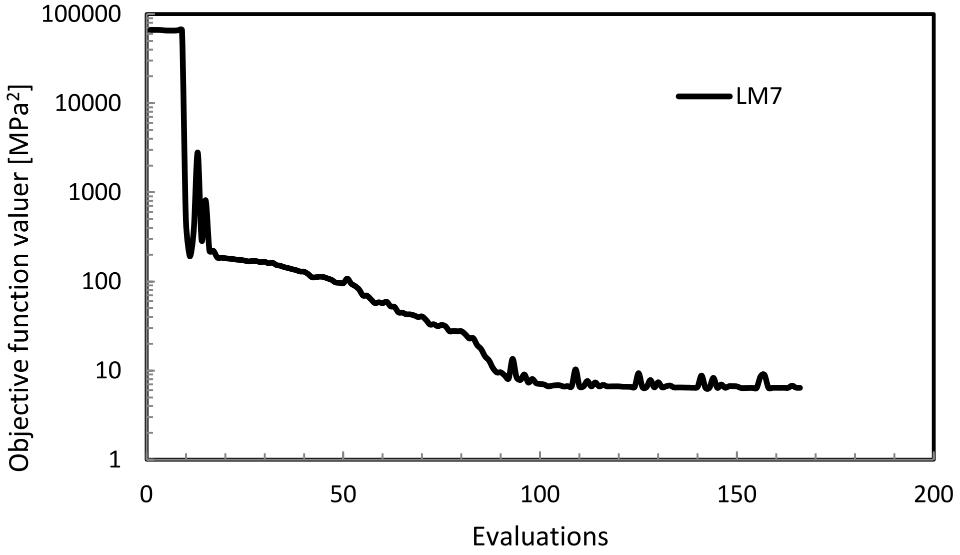


Figure 9.7 - Objective function evolution with the evaluations number of the better result of the parallel strategies for the hyperelastic model.

9.6.3. Hybrid strategies

Table 9.4 presents the results of the hybrid strategies for the three constitutive models studied in this work. Considering the results of the strategies applied to the hardening and the hyperelastic models, it was decided to analyse the elasto-viscoplastic model only considering hybrid strategies. As mentioned, this kind of strategies combine both potentialities of cascade and parallel strategies. In Table 9.4 the evaluations number is presented. For these results when appears 1000 means 1000 evaluation with the genetic algorithm, with a population size of 10. For example, the strategy *a* for the hardening model made 1000+26 evaluations. This means that 1000 evaluations were made with the genetic algorithm and then was the LM algorithm that gave better results and had made 26 evaluations.

Table 9.4 - Optimal values for the hybrid strategies.

Strategie		Hardening model	Hyperelastic mode	Viscoplastic model
		{ σ^0, σ$^\infty$, δ , ζ ,E}^{T}	{ c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8 }^{T}	{ $\boldsymbol{\sigma_0}$, K, n, Qx, Hp, bx }^{T}
a	$S_{obj1} \mid S_{obj2}$	1.6962 | 1.2783	11.1200 | 8. 5953	281.4934 |243.58
	Evaluations	1000 + 26	1000 + 10	1000 + 134
	x^*	{ 316.9, 738.8, 5.7, 665.7, 380915 }^{T}	{ -1.4894, -0.1869, 1.4050, 0.4754, 1.3050, 2.9300, -1.7300, 2.2100 }^{T}	{ 229.91, 774.25, 0.324, 69948.76, 193.16, 533,74 }^{T}
b	$S_{obj1} \mid S_{obj2}$	1. 2761 | 1. 2761	5.4895 | 5. 4253	243.5708 | 243.5655
	Evaluations	60 + 7	159 + 10	104 + 24
	x^*	{ 316.9, 731.0, 5.8, 682.9, 381740 }^{T}	{ -0.5183, 1.2441, 0.0465, 0.0283, -0.6522,-1.0743, 0.2510, -3.8530 }^{T}	{ 230, 773.75, 0.323, 69999.97, 193.66, 534,47 }^{T}
c	$S_{obj1} \mid S_{obj2}$	1. 2761 | 1.2759	5.4895 | 5. 4895	243.5708 | 243.5601
	Evaluations	60 + 1000	159 + 1000	104 + 1000
	x^*	{ 316.7, 730.0, 5.8, 683.5, 381734 }^{T}	{ -0.5183, 1.2441, 0.0463, 0.0282, -0.6493, -1.0691, 0.2515, 3.8528 }^{T}	{ 229.1, 753.75, 0.323, 69989.14, 189.77, 532,71 }^{T}
d	$S_{obj1} \mid S_{obj2}$	1. 9027 | 1.2624	13.5843 | 13.5843	343.1154 | 243.5691
	Evaluations	1000 + 91	1000 + 10	1000 + 200
	x^*	{ 314.7, 613.7, 7.7, 930.4, 382573 }^{T}	{ 2.7331, -1.0766, -1.0819, -3.0661, -1.5193, 1.2764, 1.0013, 2.6089 }^{T}	{ 229.98, 774.78, 0.324, 69987.47, 193.38, 534,10 }^{T}
e	$S_{obj1} \mid S_{obj2}$	1. 2638 | 1.2635	10.2299 | 7.3346	236.2208 | 236.0032
	Evaluations	59 + 1000	105 + 1000	175 + 1000
	x^*	{ 314.2, 590.3, 8.3, 986.9, 381911 }^{T}	{ -3.5489,0. 3062, -1.1464, -2.1641,1.2204, -1.2076, 0.3313, 3.5045 }^{T}	{ 218.76, 740.38, 0.4, 69921.13,172.45, 446.60 }^{T}
f	$S_{obj1} \mid S_{obj2}$	1.6962 | 1.2783	5.9576 | 5.9576	275.1006| 243.5655
	Evaluations	1000 + 26	161 + 1000	1000 + 96
	x^*	{ 316.9, 738.8, 5.7, 665.7, 380915 }^{T}	{ -0.1837, 1.0452, -0.0592, -3.0092, 0.2042, 3.9779, 0.8789, 0.2457}^{T}	{230, 775.46, 0.33, 69999,193.31, 533.94 }^{T}
g	$S_{obj1} \mid S_{obj2}$	1.6962 | 1.2654	11.1213 | 5.4583	281.4934 | 236.0471
	Evaluations	1000 + 56	1000 +165	1000 + 103
	x^*	{ 315.7, 656.4, 6.8, 837.8, 382810 }^{T}	{ -1.3325, 0.8525, 2.5264, 0.4968, 0.2079, 3.9889, -1.5818, -0.0218 }^{T}	{ 217.44, 741.13, 0.4, 69919.61, 174.70, 447.23 }^{T}

For the elasto-viscoplastic model, it was considered the following boundaries of each starting point: $\sigma^0 \in [50{,}230], n \in [0.1{,}0.4], K \in [600{,}800], Q_x \in [25000{,}70000], H_p \in [100{,}400]$ and $b_x \in [350{,}800]$. These boundaries consider the physical limitations and ensure the convergence of the optimization methods. For this model, the value of the objective function for the Levenberg-Marquardt single-stage optimizer was $\mathrm{S_{obj}}(\mathbf{x}^*) = 243.5708$ MPa^2 with 106 iterations. For the Evolutionary algorithm it was seen that $\mathrm{S_{obj}}(\mathbf{x}^*) = 281.4934$ MPa^2 for 1000 iterations.

Considering the results for the hybrid strategies applied to the hardening model it is possible to conclude that all the strategies lead to satisfactory and similar results. Comparing with the single-stage approaches, the strategies *a*, *d*, *f* and *g* have registered good improvements. For the other strategies, in terms of the objective function value at point 1 and 2, it is not possible to register improvements. In Figure 9.8 it is presented the better hybrid strategy results in terms of the objective function evolution for the hardening model, that is, the strategy d. It is possible to observe that the first 1000 generations from the EA method lead to a good decrease in the objective function value. However the LM method considered after also lead to a good decrease in terms of objective function value in only 91 iterations.

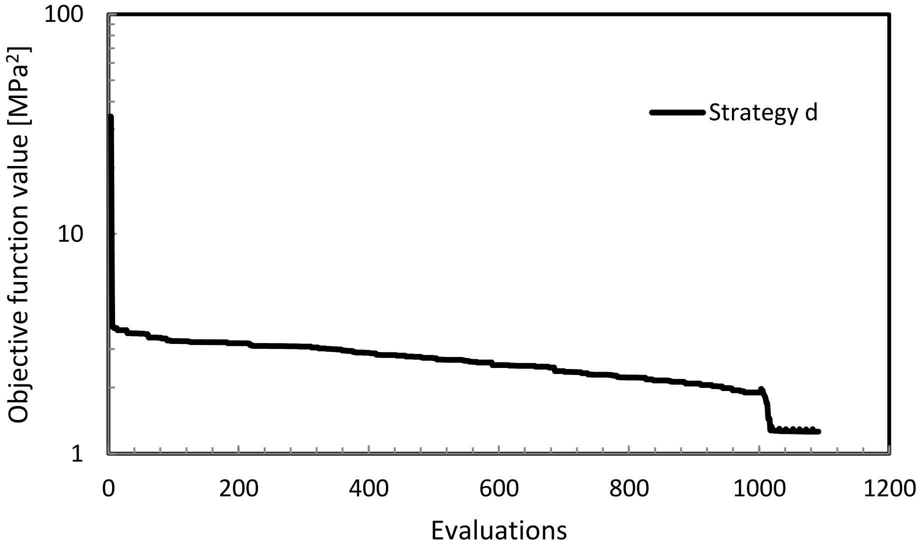


Figure 9.8 - Objective function evolution with the evaluations number of the better result of the hybrid strategies for the hardening model.

For the hyperelastic model all the hybrid strategies lead to satisfactory results even with the difference in the obtained parameters. This fact is related to the high number of local minimums. Comparing with the single-stage strategies, the strategies *a*, *e* and *g* have registered good improvements in the objective function value. For the other strategies, in terms of the objective function value at point 1 and 2, it is not possible to register considerable improvements.

In Figure 9.9 it is presented the evolution of the objective function value for the better hybrid strategy for the hyperelastic model, that is the strategy *b*. It is possible to observe that the second LM method considered, after the 159 iterations from the first LM, doesn´t lead to a significant decrease of the objective function value.

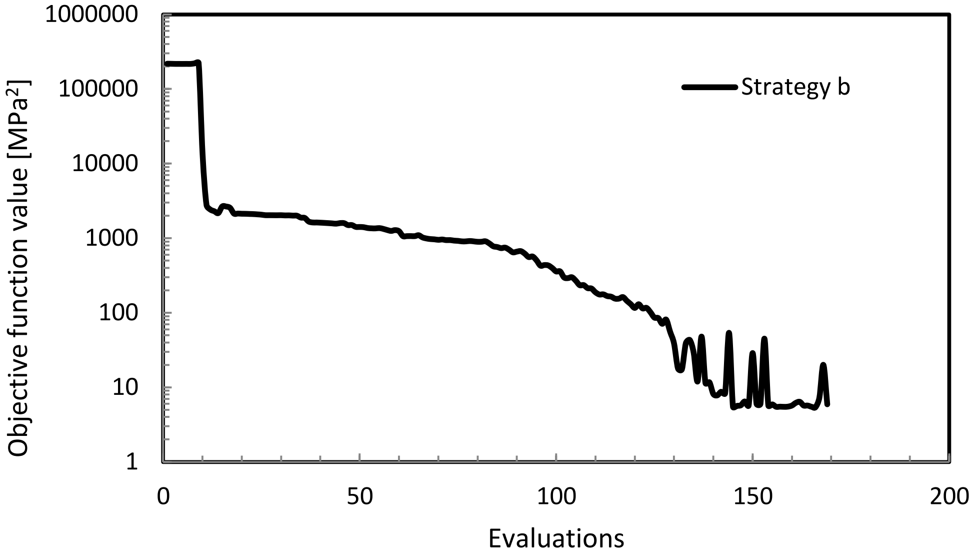


Figure 9.9 - Objective function evolution with the evaluations number of the better result of the hybrid strategies for the hyperelastic model.

Also for the elasto-viscoplastic model the hybrid strategies lead to satisfactory results and similar results. Comparing with the single-stage EA, it is possible to conclude that all the strategies lead to good improvements considering the objective function values. Comparing with the LM single-stage optimizer, only the strategies *e* and *f* leads to good improvements in terms of the objective function value. For the other strategies, in terms of the objective function value at point 1 and 2, it is not possible to register considerable improvements. In Figure 9.10 it is possible to observe the evolution of the objective function with the evaluations number for the strategy *e*. As it is possible to observe the higher decrease in terms of objective function value is registered for the first LM method applied. In this case the EA method doesn´t bring significant improvements.

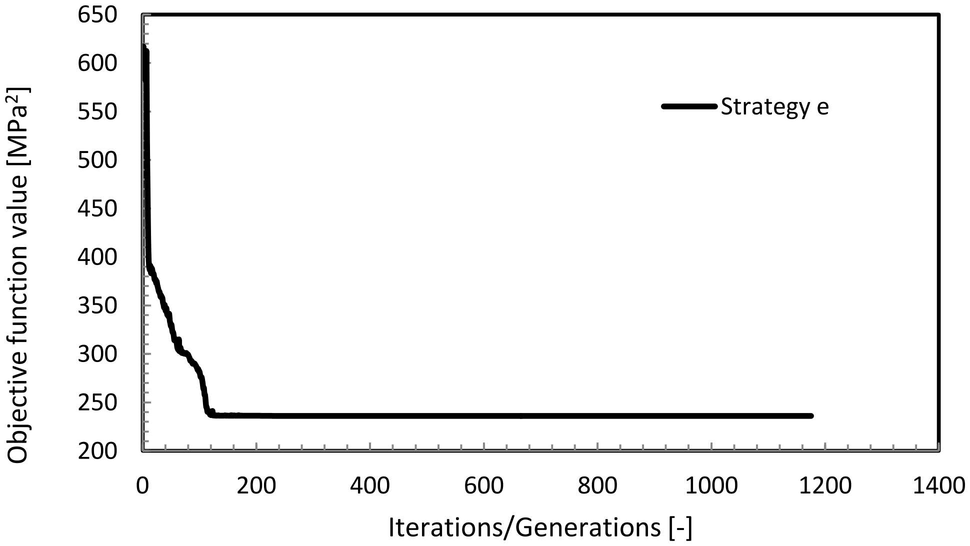
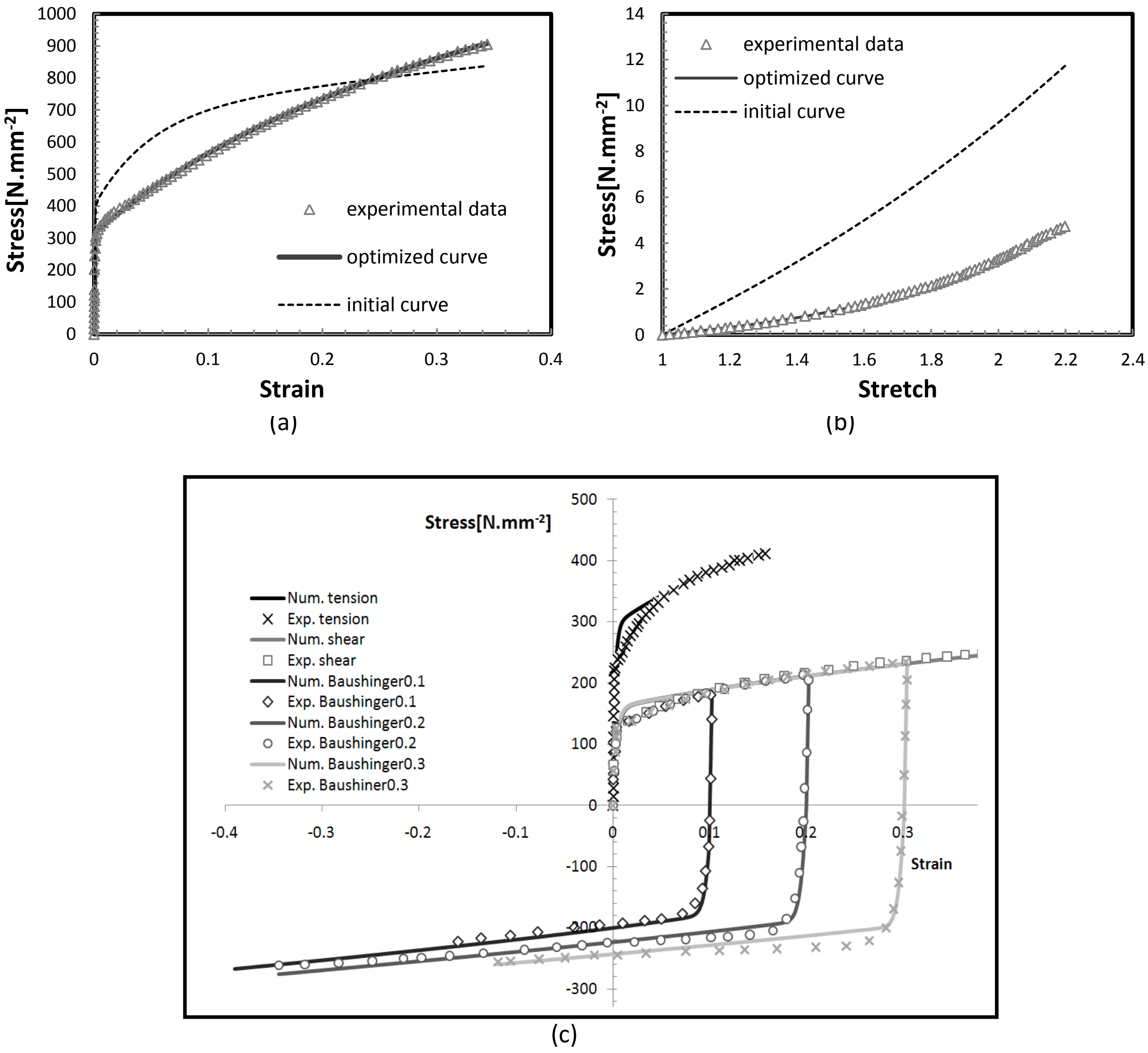


Figure 9.10 - Objective function evolution with the iterations/generations number of the better result of the hybrid strategies for the elasto-viscoplastic model.

However, it is worth mentioning that the fact of having parallel algorithms, cascade algorithms or even both leads to a major probability to reach the global minimum. These conclusions can be easily extrapolated to other problems with a previous analysis.

The initial and the optimized curves are compared for the three models presented. For the generation of the optimized curves it was used the best set of parameters found with the strategies previously proposed, being represented in Figure 9.11. It is shown that for the three constitutive models the optimized curve fits very well the experimental data.



(a)

(b)

(c)

Figure 9.11 - Experimental data and optimized curves for: a) hardening model, b) hyperelastic model and c) elasto-viscoplastic model.

9.7 Conclusions

In this chapter, the parameter's determination of three constitutive models was performed. The constitutive models studied are an elastic-plastic model with non-linear hardening, a hyperelastic model, the Ogden model, and an elasto-viscoplastic model with isotropic and kinematic work-hardening. It was considered a priori that these constitutive models are the relevant ones for the studied materials. Cascade, parallel and hybrid strategies were developed and studied. These strategies are based in the combination of a gradient-based algorithm and an evolutionary algorithm. The aim of these strategies was to take advantage of the strength of each selected algorithm and improve the overall robustness and efficiency of classical optimization methodologies based on single stages.

The obtained results for the constitutive models leads to numerical curves that fit quite well the experimental data. The developed strategies lead to better values of the objective function than the single stage optimization procedures. During the strategy selection for a specific constitutive model all the strategies should be analysed in detail.

From the results here presented, it was considered that for the non-linear model the hybrid strategies *b*, *c* and *e* are the only ones that do not give any improvement at all relatively to the classical single-stage methods. For the hyperelastic model, the strategies *a* and *d* are not recommended in the parameters determination. The others strategies here proposed reach good parameters set.

References

[1] J Ponthot, J Kleinermann (2006) A cascade optimization methodology for automatic parameter identification and shape/process optimization in metal forming simulation. Computer Methods in Applied Mechanics and Engineering 195:5472-5508.

[2] Andrade-Campos A, Pilvin P, Teixeira-Dias F, Simões J (2007) Determinação dos parâmetros constitutivos através de metodologias de optimização em cascata. Proceedings of CMNE/CILAMCE, Portugal.

[3] Andrade-Campos A, Pilvin P, Simões J, Teixeira-Dias F (2008) Software development for inverse determination of constitutive model parameters. Software Engineering: New Research, Nova Science Publishers, Inc..

[4] Patnaik S (1996) Comparative evaluation of different optimization algorithms for structural design applications. International Journal for Numerical Methods in Engineering 39:1761-1774.

[5] Patnaik S (1997) A cascade optimization strategy for solution of difficult design problems. International Journal for Numerical Methods in Engineering 40:2257-2266.

[6] Papadrakakis M, Tsompanakis Y, Lagaros N (1999) Structural shape optimization using evolution strategies. Engineering Optimization 31:515-540.

[7] Lagaros N, Papadrakakis M, Kokossalakis G (2002) Structural optimization using evolutionary algorithms. Computers and Structures 80:571-589.

[8] Lagaros N, Plevris V, Papadrakakis M (2005) Multi-objective design optimization using cascade evolutionary computations. Computational Methods in Applied Mechanics and Engineering. 194:3496-3515.

[9] Charmpis D, Lagaros N, Papadrakakis M (2005) Multi-database exploration of large design spaces in the framework of cascade evolutionary structural sizing optimization. Comput. Methods Appl. Mech. Engrg. 194:3315-3330.

[10] Patnaik S, Hopkins D (2000) General-purpose optimization method for multidisciplinary design applications. Advances in Engineering Software 31:57:63.

[11] Patnaik S, Guptill J, Hopkins D (2005) Subproblem optimization with regression and neural network approximators. Computational Methods in Applied Mechanics and Engineering. 194:3359-3373.

[12] Cao J, Lin J (2008) A study on formulation of objective functions for determining material models. International Journal of Mechanical Sciences 50:193-204.

[13] Andrade-Campos A, de-Carvalho R, Valente R (2012) Novel criteria for determination of material model parameters. Journal of Mechanical Sciences 54:294-305.

[14] Belegundo A, Chandrupatla T (1999) Optimization Concepts and Applications in Engineering. Prentice-Hall, Inc, New Jersey.

[15] Lemaitre J, Chaboche J (1996) Mécanique des Matériaux Solides. Editions Dunod, Paris-France.

[16] Yeniay O (2005) Penalty function methods for constrained optimization with genetic algorithm. Mathematical and Computational Applications 10:45-56.

[17] Martins P, Natal R, Ferreira A (2006) A comparative study of several material Models for Prediction of Hyperelastic Properties: Application to Silicone-Rubber and Soft Tissues. Blackwell Publishing Ltd 135-147.

[18] Humphrey J (2003) Continuum biomechanics of soft biological tissues. Proceedings of Mathematical Physical and Engineering Sciences 459:3-46.

[19] Martins J, Pires E, Salvado R, Dinis P (1998) A numerical model of passive and active behavior of skeletal muscles. Computer Methods in Applied Mechanics and Engineering 151:419-433.

[20] Mooney M (1940) A theory of large elastic deformation. Journal of Applied Physics 11:582-592.

[21] Rivlin R (1948) Large elastic deformations of isotropic materials. iv: Further developments of the general theory. Philosophical Transactions Royal Society London 241-379.

[22] Twizell E, Ogden R (1983) Non-linear optimization of the material constants in Ogden´s stress-deformation function for incompressible isotropic elastic materials. Journal of Australian Mathematical Society 424-434.

[23] Veronda D, Westmann R (1970) Mechanical characterization of skin-finite deformations. Journal of Biomechanics 3:111-124.

[24] Thuillier S, Manach P (2009) Comparison of the work-hardening of metallic sheets using tensile and shear strains paths. International Journal of Plasticity 733-751.

[25] Hill R (1948) A theory of yielding and plastic flow of anisotropic metals, Philosophical Transactions Royal Society London 281–297.

[26] Chaboche J (2008) A review of some plasticity and viscoplasticity constitutive theories. International Journal of Plasticity 24:1642-1693.

Chapter 10

On the Objective Function Evaluation – Single-Point or FE Analysis

In the present chapter two different approaches are presented and analysed: the single-point and FE analysis. The use of these different methodologies for the evaluation of objective function in the parameter identification process is still an open question and the interest in this field has been increasing among the metal forming community. To discuss this issue, two different constitutive models suitable for metals were used, *i.e.* an non-linear elasto-plastic hardening model and an elasto-viscoplastic model with isotropic and kinematic hardening. The determined material parameters for the two models, the respective objective function values and the CPU time required to perform the simulations are presented and discussed.

10.1 Introduction

The present chapter deals with parameter identification processes and, as it was mentioned before, the aim of these problems is to estimate material parameters for constitutive models. The determination of the constitutive models parameters can be accomplished solving an inverse problem, which consists in searching for a set of parameter values for which the experimental and the numerical simulation results are similar. The comparison between the results of the mathematical model and the experimental data is performed by an objective function that is subjected to optimization methods.

The use of Finite Element (FE) or single-point analysis for the evaluation of objective functions in the identification process is still an open question. By definition, a constitutive model is a mathematical representation of the phenomena that take place in an infinitesimal amount of material (according to the continuum mechanics theory). On one hand, the single infinitesimal point evaluation seems to characterize an infinitesimal amount of material subjected to all kind of deformation history. Although it is computationally very inexpensive, it cannot be used to account phenomena such as localized necking or springback. On the other hand, in all FE analysis codes, the constitutive model is implemented and accounted for each element integration point. Numerical approximations of the FE method include iteratively accumulated errors, and can impair the whole identification process. Additionally, the Finite Element Method is still a numerical approximation method for the continuum. Nonetheless, FE analyses allow to model the specimen used in the experimental procedure and predict geometric phenomena such as necking and springback. The main difference between the two approaches is the time and space integration methods used in each one. This fact influences the obtained results in the evaluation of the objective function and afterwards the optimization process. For the single-point analysis, the numerical method used to integrate in time is a second order Runge-Kutta method with adaptive time step and error control [1]. The Finite Element analysis was performed using the implicit FE code Abaqus®/STANDARD that generally uses Newton’s method as a numerical technique for solving the nonlinear equilibrium equations. For each time increment, the implicit methods need to converge leading to a large computation time when compared with the explicit methods. However the explicit methods can iteratively accumulate errors [2].

In this work, the use of single-point or metal forming FE analysis for the evaluation of the objective function in the identification process is analysed. The objective function considered is described in equation 9.1, from the previous chapter. This study is conducted considering two different constitutive models and two different steels commonly used in metal forming processes. The models are a non-linear elasto-plastic hardening model and an elasto-viscoplastic model, in which the work-hardening combines isotropic and kinematic contributions. In order to determine the best parameters set, a gradient-based Levenberg-Marquardt [3] optimization algorithm was applied. It was considered as the stopping criterion a maximum number of 200 iterations or, as an alternative, a stagnation value of 1x10^{-30}, in terms of objective function value for each iteration. The objective function sensibilities are calculated by means of forward finite differences with a perturbation value of 5x10^{-3}.

10.2 Constitutive models

10.2.1 Non-linear elasto-plastic hardening model

The first constitutive model here studied is the elasto-plastic model with non-linear hardening for stainless steel AISI 304 presented in the previous chapter in section 9.4.1. However in this model the elastic part is considered constant and defined as E=380 GPa and ν=0.29.

10.2.2 Elasto-viscoplastic model with isotropic and kinematic hardening

For this constitutive model the material studied was a hardening mild steel E220BH with $E = 210$ GPa. The experimental data used was obtained by Thuillier *et al.* [4] and is the same used in the previous chapter in section 9.4.3. The model considers the following yield criterion:

$$f(\boldsymbol{\sigma}, \mathbf{X}, R) = \bar{\sigma} - R = \sqrt{\frac{3}{2}(\boldsymbol{\sigma}^{\mathrm{d}} - \mathbf{X}) : \mathbf{M} : (\boldsymbol{\sigma}^{\mathrm{d}} - \mathbf{X})} - R, \tag{9.8}$$

where $\boldsymbol{\sigma}^{\mathrm{d}}$ represents the deviatoric part of $\boldsymbol{\sigma}$, $\bar{\sigma}$ is the equivalent stress and $\mathbf{M}$ is the matrix representation of the 4th order Hill's constitutive tensor.

The viscoplastic component of the strain follows the flow rule (Lemaitre and Chaboche [5]):

$$\Omega(f) = \frac{K^{\nu}}{n^{\nu} + 1}\left(\frac{f^{+}}{K^{\nu}}\right)^{n^{\nu}+1}, \tag{9.9}$$

where n^{v} is the strain rate sensitivity coefficient, K^{v} a weighting coefficient of the viscous part of the stress and f^{+} the positive part of f. The behavior is thus elastic if $f < 0$ and if $f > 0$ the viscoplastic strain rate is written as:

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}} = \frac{\partial \Omega}{\partial \boldsymbol{\sigma}} = \Omega'(f)\frac{\partial f}{\partial \boldsymbol{\sigma}}, \tag{9.10}$$

in order to obtain $f = 0$ for plastic deformation. The equivalent viscoplastic strain rate $\dot{\varepsilon}_{\mathrm{eq}}^{\mathrm{vp}}$ is defined from the plastic work conservation principle:

$$\dot{\varepsilon}_{\mathrm{eq}}^{\mathrm{vp}} = \frac{\left(\boldsymbol{\sigma}^{\mathrm{d}} - \boldsymbol{X}\right) : \dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}}}{\bar{\sigma}}. \tag{9.11}$$

The work-hardening combines isotropic and kinematic contributions, and the evolution of the isotropic work-hardening is related to the accumulated plastic strain following the swift law, *i.e.*

$$R = K(\varepsilon_{\mathrm{eq}}^{\mathrm{vp}} + \varepsilon_0)^n \quad \text{with} \quad \varepsilon_0 = \left(\frac{\sigma_0}{K}\right)^{1/n}, \tag{9.12}$$

where K is a material parameter, n the hardening coefficient and σ_0 is the initial yield stress in tension along the RD.

The non-linear evolution law of the kinematic work-hardening is based in the additive combination of a purely kinematic term (linear Ziegler hardening law) and a relaxation term (the recall term), which introduces the nonlinearity. This law is expressed as

$$\boldsymbol{X} = C\frac{1}{\sigma_0}(\boldsymbol{\sigma} - \mathbf{X})\dot{\varepsilon}_{\mathrm{eq}}^{\mathrm{pl}} - \gamma \mathbf{X}\dot{\varepsilon}_{\mathrm{eq}}^{\mathrm{pl}}, \tag{9.13}$$

where C and γ are material parameters that must be determined, with C being the initial kinematic hardening module, and γ defining the rate at which the kinematic hardening module decreases with the increasing of the plastic deformation [2,6].

10.3 Numerical results and discussion

The parameter identification process based on the single-point analysis was achieved with the SDL optimization software [7,8]. For the FE analysis the Abaqus® code was integrated with the SDL optimization program. In Abaqus®, two different specimens were considered in order to simulate the tensile and the shear tests for both constitutive models here studied. For both specimens, the nonlinear effects of large deformations were considered. Considering the symmetry inherent of the tensile tests only a quarter of the specimen was modelled for this test. Therefore, the dimensions considered were $7 \times 30 \times 1$ mm^3 and an equal spaced mesh of 20×50 elements was applied. The specimen subject to the shear test has $4.5 \times 50 \times 1$ mm^3 and an equal spaced mesh of 20×200 was applied. For both specimens a 4-node bilinear plane stress quadrilateral element with reduced integration and hourglass control was applied. In Figure 10.1 it is possible to observe the final meshes for one of the tensile and the shear tests performed.

During the parameter identification process an absolute weight factor equal to 1 was defined for the objective function evaluation. In Table 10.1, the results obtained with the single-point and the FE analysis, for the non-linear elasto-plastic model, are presented. For this model all the optimization parameters boundaries where considered equal to $[0{,}1000]$ MPa.

Table 10.1 - Single-point and FE analysis results for the non-linear elasto-plastic hardening model.

Parameters	Starting set	Single-point	FE analysis
σ^0 [MPa]	310	314.51	320.09
σ^∞ [MPa]	700	614.19	680.72
δ	7	7.74	6.72
ζ [MPa]	800	929.84	768.49
$S_{\text{obj}}(\mathbf{x})$ [MPa^2]	-	126.34	141.99
Iterations	-	86	57
CPU[1] [s]	-	2	780

The results presented in Table 10.1 allow to conclude that for the elasto-plastic hardening model the single-point analysis leads to more satisfactory results in terms of objective function value. For both approaches the boundaries of the optimization variables were not achieved and the optimum values are quite similar.

Figure 10.2 highlight the fact that the single-point approach reaches lower objective function values. For the FE analysis there are some peaks that correspond to values where the constraint was active (this constraint, implemented as penalty, is referred in section 9.4.1).

[1] CPU time expended in an Intel® Core$^{\text{TM}}$ 2 Quad CPU Q 9400 at 2.66GHz with 3.25GB of RAM.

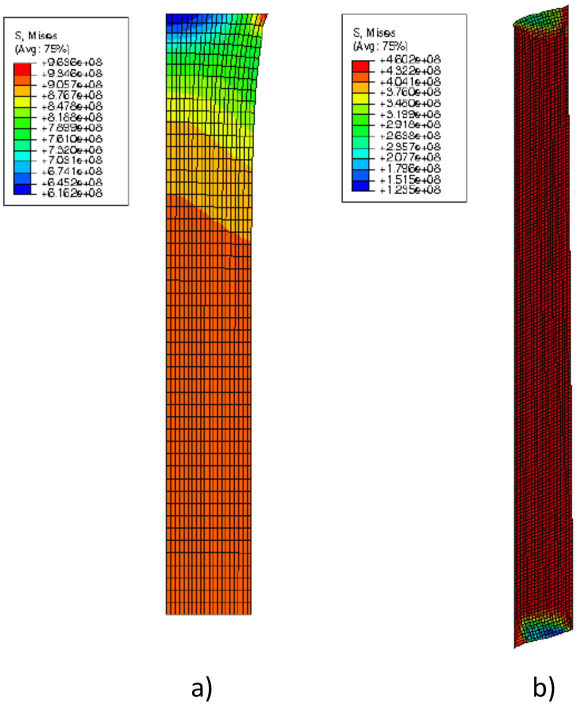


Figure 10.1 - The finite element mesh after the a) tensile and b) shear test.

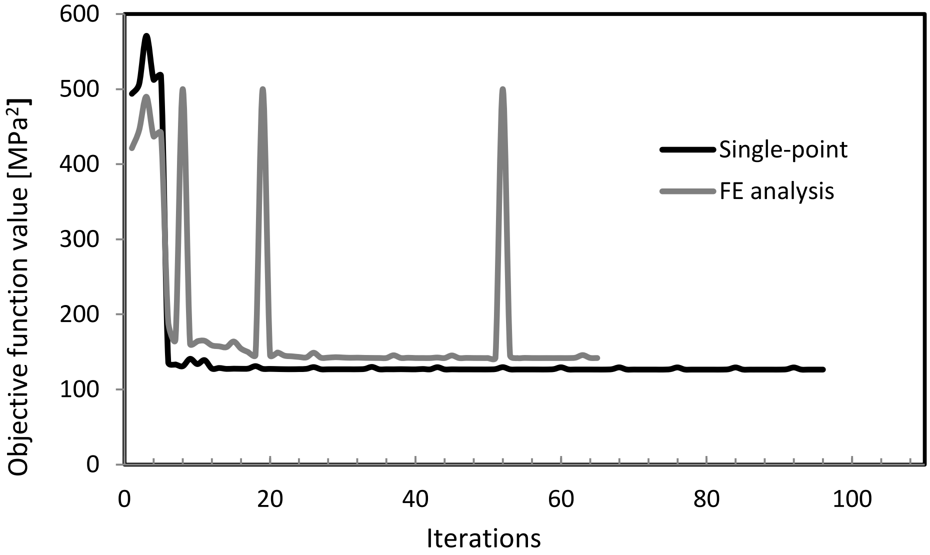


Figure 10.2 - Objective function evolution with the iteration number for the non-linear elasto-plastic hardening model.

Considering the CPU time, the single-point method also leads to better results. For the elasto-plastic hardening model it is considered that the single-point analysis is more efficient in terms of objective function value/computational cost relation, however both approaches leads to good values of the stress-strain curve as it is possible to observe in Figure 10.3.

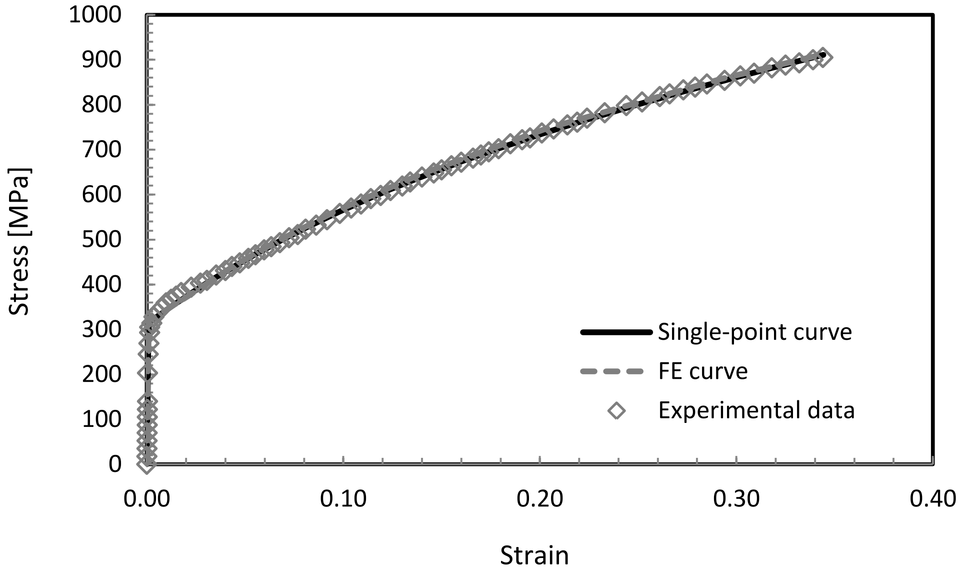


Figure 10.3 - Experimental and numerical curves for the non-linear elasto-plastic hardening model.

In Figure 10.4 and Figure 10.5 the evolution of the optimization variables during the optimization process is presented. For the single-point approach it is possible to see that for the initial decrease of the objective function value the variable that have significantly contributed was the σ^{∞}. After, during the optimization process, it seems that both σ^{∞}, δ and ζ have contributed to a light decrease of the objective function. It also seems that σ^{0} has not contributed to the final result, showing stagnation during almost all process.

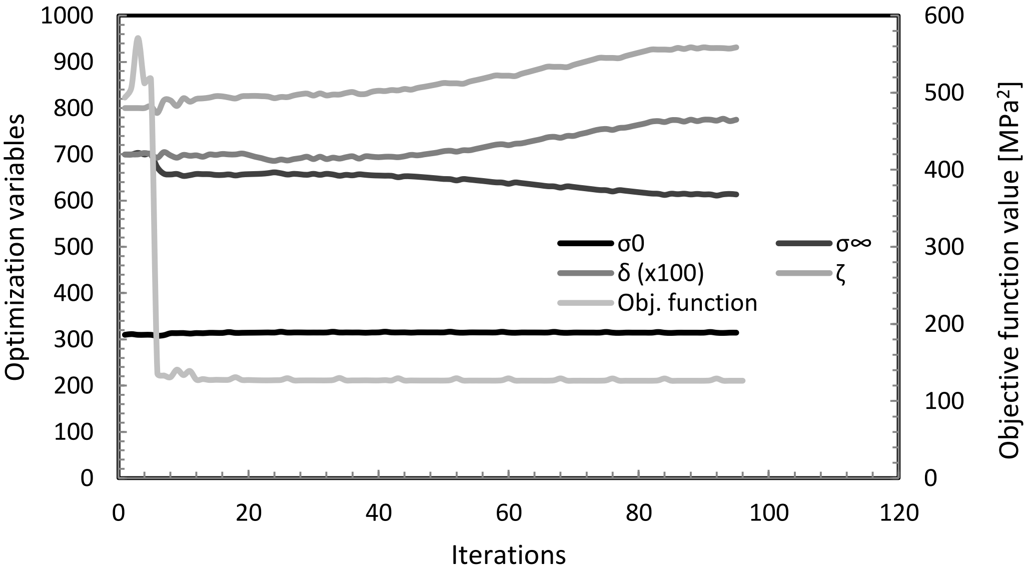


Figure 10.4 - Optimization variables evolution of the single-point approach for the non-linear elasto-plastic hardening model.

In the case of the FE analysis for the non-linear elasto-plastic hardening model, the initial and more considerable decrease happen mainly with the modification of the variables σ^{∞} and δ.

During the remaining iterations it seems that all the optimization variables lead to a light decrease of the objective function. However, σ^0 had the smallest variation, as already was observed in single-point analysis.

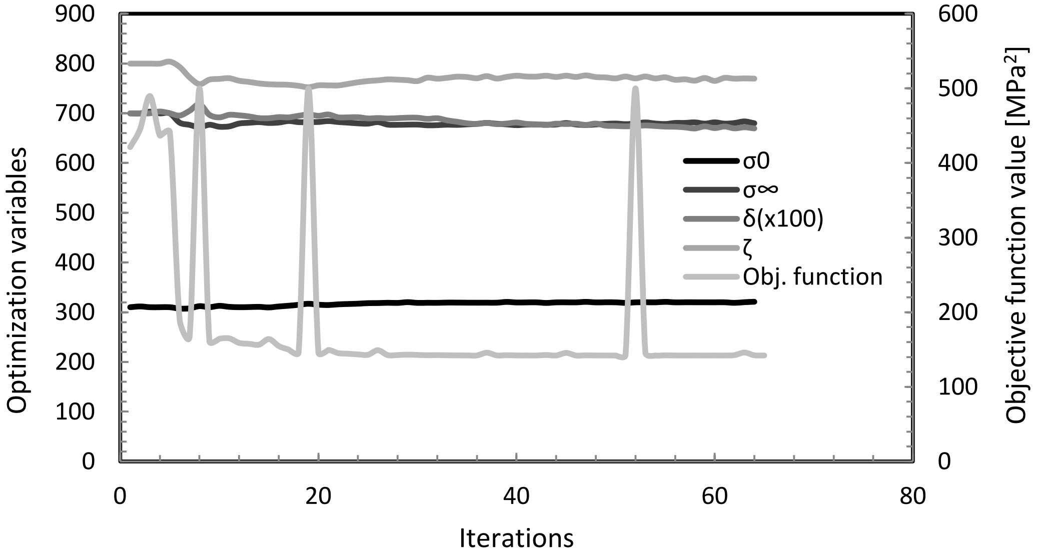


Figure 10.5 - Optimization variables evolution of the FE approach for the non-linear elasto-plastic hardening model.

In the case of the elasto-viscoplastic model with isotropic and kinematic hardening, it is also observed that the single-point analysis leads to better results when we consider the error function value and the CPU time. The boundaries considered for the model parameters were $\sigma_0 \in [50{,}230]$ MPa, $C \in [0{,}10000]$MPa, $\gamma \in [0{,}100]$, $K \in [300{,}700]$ MPa and $n \in [0{,}0.4]$.

Table 10.2 - Non-linear elasto-viscoplastic model with isotropic and kinematic hardening.

Parameters	Starting set	Single-point	FE analysis
σ_0 [MPa]	180	160.535	188.090
C [MPa]	1000	8598.673	1069.300
γ	28	99.998	28.053
K [MPa]	550	323.264	547.280
n	0.2	0.204	0.220
$S_{\text{obj1}}(\mathbf{x})$ [MPa^2]	-	1239.16	4877.49
$S_{\text{obj2}}(\mathbf{x})$ [MPa^2]	-	299.57	4744.65
$S_{\text{obj3}}(\mathbf{x})$ [MPa^2]	-	433.90	593.76
$S_{\text{obj4}}(\mathbf{x})$ [MPa^2]	-	541.32	108.24
$S_{\text{obj5}}(\mathbf{x})$ [MPa^2]	-	537.14	1638.35
$\sum_{i=1}^{n_{\text{points}}} S_{\text{obj}}(\mathbf{x})$ [MPa^2]	-	3051.09	11962.49
Iterations	-	200	40
CPU[2] [s]	-	480	28800

[2] CPU time expended in an Intel® Core™ 2 Quad CPU Q 9400 at 2.66GHz with 3.25GB of RAM.

In Figure 10.6 it is evident that the single-point analysis leads to better results in terms of objective function value. In Table 10.2 the error function values for each test is also presented, being $S_{\mathrm{obj}}(\mathbf{x})$ the error function value for the tensile test ($S_{\mathrm{obj1}}(\mathbf{x})$), the shear test ($S_{\mathrm{obj2}}(\mathbf{x})$), the cyclic test with inversion at 0.3 value of deformation ($S_{\mathrm{obj3}}(\mathbf{x})$), the cyclic test with inversion at 0.2 value of deformation ($S_{\mathrm{obj4}}(\mathbf{x})$) and the cyclic test with inversion at 0.1 value of deformation ($S_{\mathrm{obj5}}(\mathbf{x})$). In both tests it was for the tensile test that the optimized parameters lead to poorer results. Also in this case it is considered that the single-point analysis gives good improvements in terms of objective function value when compared with the FE analysis (as it possible to observe in Figure 10.7) and the CPU time is really advantageous for the single-point analysis. However, for this constitutive model, the optimum set reached for both strategies are quite dissimilar (with special enphasis in the C and K parameters) which contributes for the large discrepancy of the final objective function value. It is also noted that only the γ optimization variable (in the single-point strategy) reached the boundaries, activating the boundary restriction. In terms of optimization process, this could mean that the optimum can be outside the search universe.

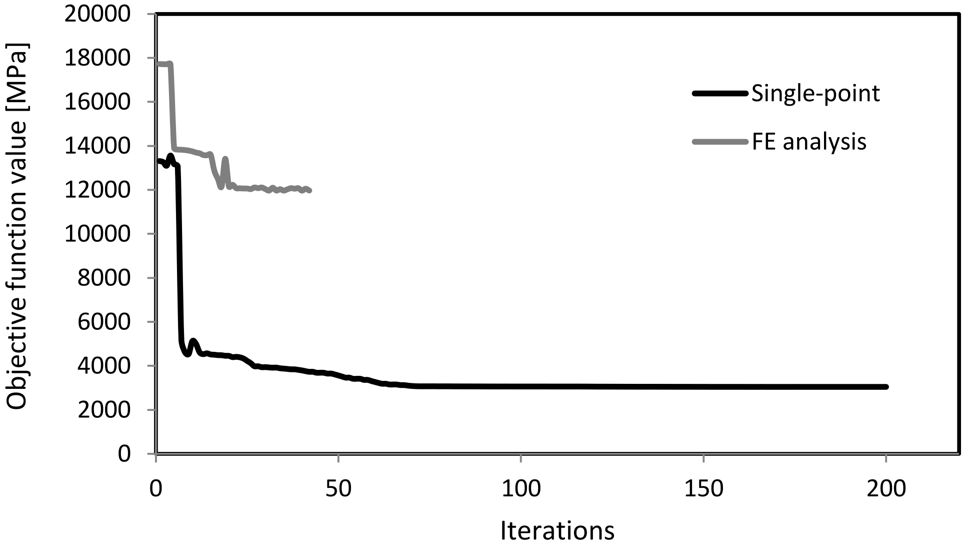


Figure 10.6 - Objective function evolution with the iteration number for the non-linear elasto-viscoplastic model with isotropic and kinematic hardening.

In Figure 10.8 to Figure 10.9 is presented the optimization variables evolution during the optimization process, for the two approaches, when applied to the non-linear elasto-viscoplastic model with isotropic and kinematic hardening. For the single-point, it is possible to observe that for the initial decrease of the objective function, the variable that more contributed was K. Afterward, the K, γ, n and C were significantly changed to obtained a lower objective function. Stagnation of the optimization process can be observed after iteration 75.

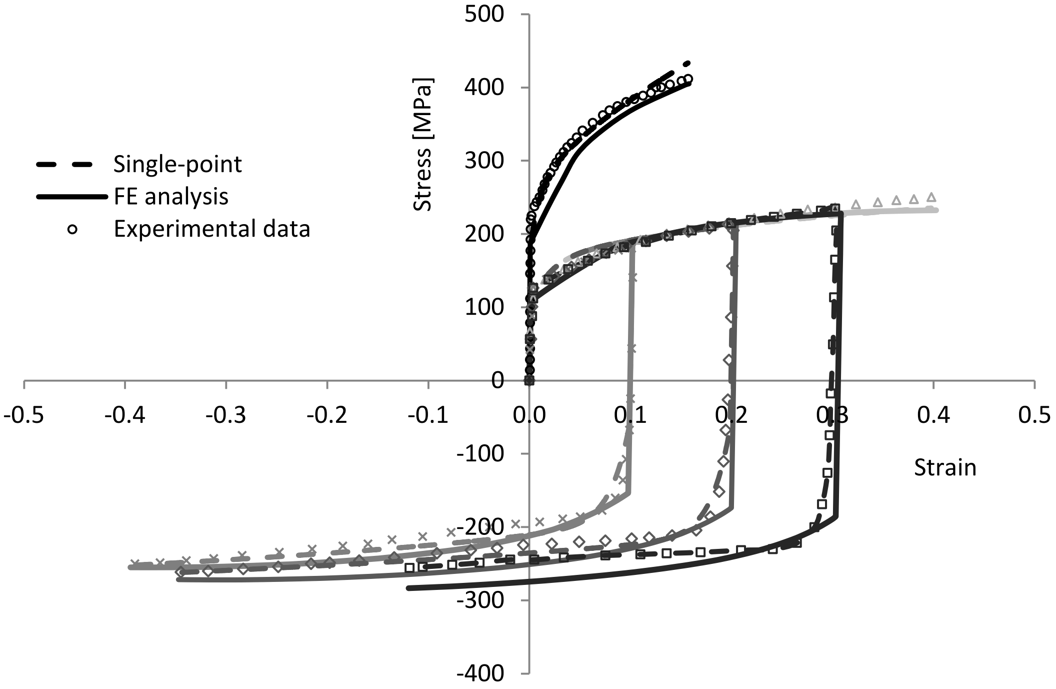


Figure 10.7 - Experimental and numerical curves for the non-linear elasto-viscoplastic model with isotropic and kinematic hardening

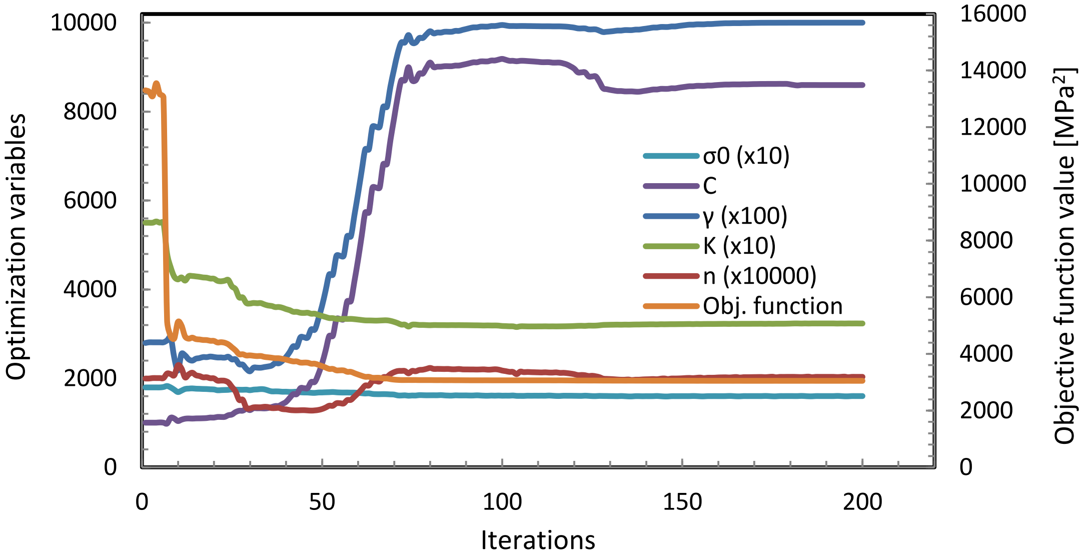


Figure 10.8 - Optimization variables evolution of the single-point approach for the non-linear elasto-viscoplastic model with isotropic and kinematic hardening.

Considering the FE approach, it is the variation of the C optimization variable that more contributes for the variation of the objective function. However, for this approach the value of the optimization variables doesn´t vary enough to conduct the objective function to lower values.

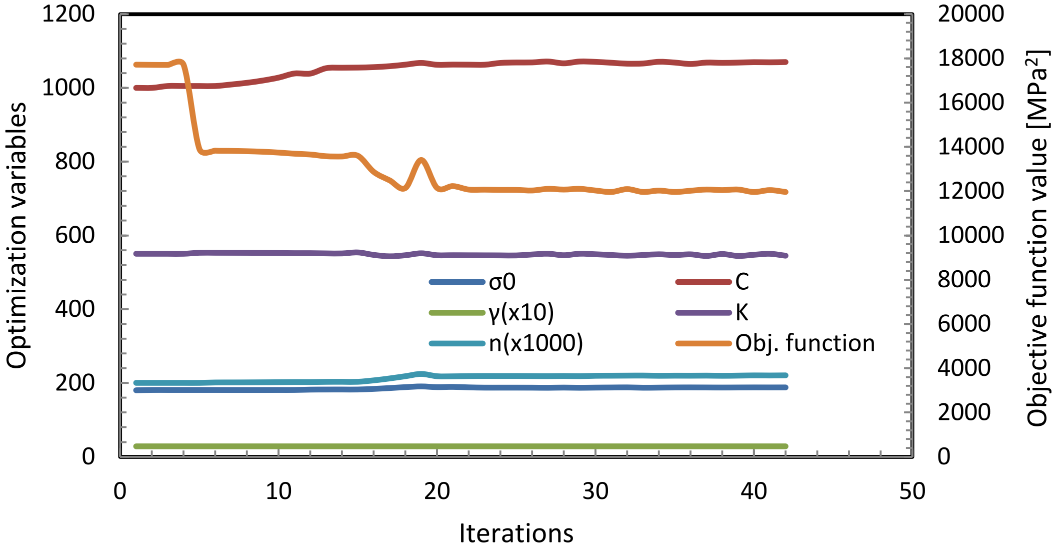


Figure 10.9 - Optimization variables evolution of the FE approach for the non-linear elasto-viscoplastic model with isotropic and kinematic hardening.

After the results presented, other study was performed. This study had consisted in considering the optimum set of material parameters obtained with the single-point as the material parameters for the FE analysis. The opposite was also considered, *i.e.*, the optimum set of material parameters obtained with the FE analysis was considered as inputs in the single-point analysis. This study was performed for both constitutive models. In Figure 10.10 it is presented the constitutive curves obtained for the non-linear elasto-plastic hardening model. The FE analysis represented in the figure is considering the FE analysis with constitutive parameters obtained with the single-point approach (the optimum set). The single-point analysis represented is the opposite. In terms of constitutive curves, both approaches lead to good results, very near to the experimental data. In terms of objective function value the FE analysis reached a value of 155.7633 MPa^2 and the single-point reached a value, of 140.5595 MPa^2. In this case quite good results were obtained and it seems that the parameters identified with a different strategy can be used, *i.e.* single-point parameters can be used in FE analysis and FE analysis parameters can be used in single-point analysis.

The same study was performed for the non-linear elasto-viscoplastic model with isotropic and kinematic hardening and the results obtained are presented in Figure 10.11. As it is possible to observe when the optimum set obtained with the FE analysis is considered as material parameters for the single-point analysis, the results fit quite well the experimental data and a objective function value of 10651.32 MPa^2 was achieved (more than 3x the value reached with the initial single-point strategy). However, for the FE analysis with the single-point optimum set, the results do not fit the experimental data, reaching a value of objective function of 49636.9354 MPa^2. Therefore, the FE strategy seems to be dependent of its approach for identifying the constitutive model parameters. This can be called a more conservative approach. The single-point analysis seems to be a more general approach, well accepting parameters identified by other approach. Consequently, for this constitutive model it is possible to conclude that the results achieved depend on the strategy considered.

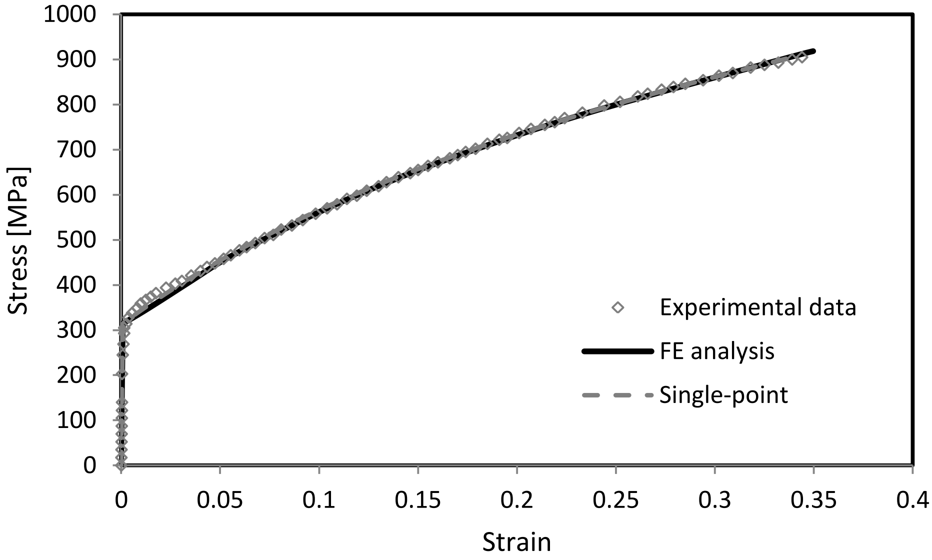


Figure 10.10 - Experimental and numerical curves for the non-linear elasto-plastic hardening model for the study with initial optimization set changed.

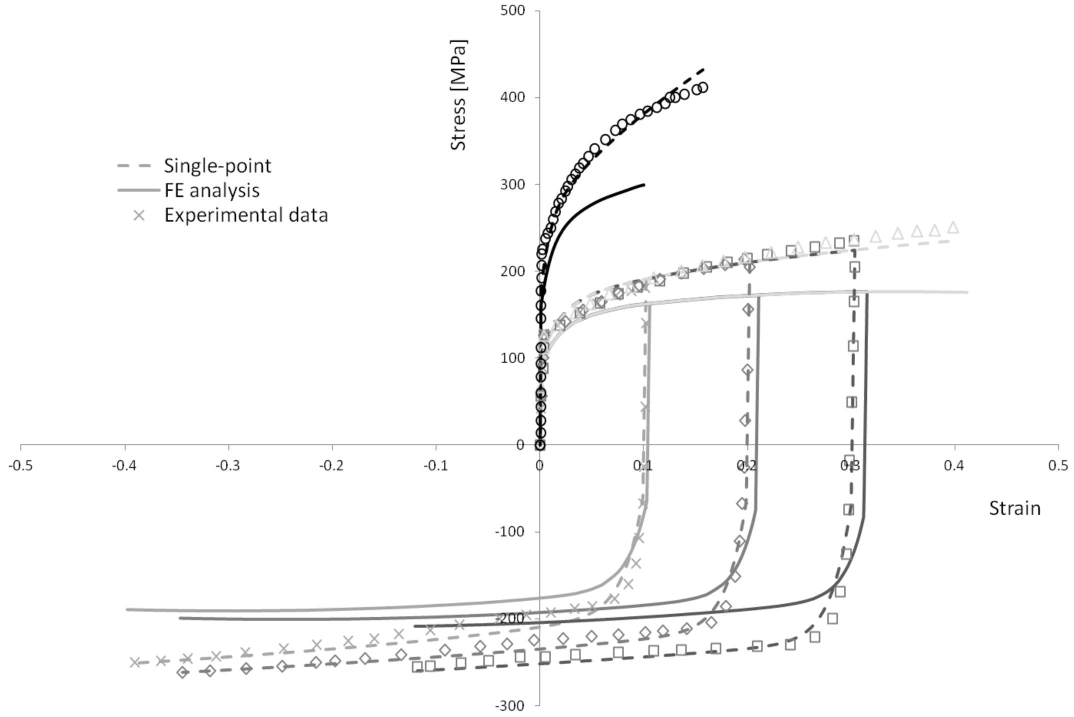


Figure 10.11 - Experimental and numerical curves for the non-linear elasto-viscoplastic model with isotropic and kinematic hardening, for the study with initial optimization set changed.

10.4 Conclusions

A comparative study between the use of the single-point and the FE analysis in the parameters determination problems was performed. The constitutive models studied were a non-linear elasto-plastic hardening model and a non-linear elasto-viscoplastic model with isotropic and kinematic hardening. For both constitutive models the single-point analysis was considered more efficient in terms of objective function value/computational cost relation. It was considered that both of the strategies presented can be applied depending on the studied problem. The FE analysis allows the user to know all the history deformation of a complex geometry and predict geometrical phenomena such as necking, springback and stress concentration. In the problems that the geometrical phenomena doesn´t exist the single-point is more appropriate considering the good relation between the CPU time and the objective function values.

It was verified that for the non-linear elasto-plastic hardening model the single-point parameters can be used in FE analysis and FE analysis parameters can be used in single-point analysis.

For the non-linear elasto-viscoplastic model with isotropic and kinematic hardening it was verified that the results achieved depend on the strategy considered.

References

[1] Andrade-Campos A (2001) Development of an optimization framework for parameter identification and shape optimization problems in engineering. International Journal of Manufacturing, Materials, and Mechanical Engineering 1:57-79.

[2] ABAQUS 6.7 (2007) User Manual, Simulia Inc, Dassault Systémes.

[3] Marquardt D (1962) An algorithm for least-squares estimation of nonlinear parameters. Journal of Society for Industrial and Applied Mathematics 11:431-441.

[4] Thuillier S, Manach P (2009) Comparison of the work-hardening of metallic sheets using tensile and shear strains paths, International Journal of Plasticity, 733-751.

[5] Lemaitre J, Chaboche J (1996) Mécanique des Matériaux Solides. Editions Dunod, Paris-France.

[6] de-Carvalho R, Valente R, Andrade-Campos A (2010) On the objective function evaluation in parameter identification of material constitutive models - Single-point or FE analysis. Proceedings of 13th ESAFORM Conference on Material Forming, Brescia, Italy

[7] de-Carvalho R, Valente R, Andrade-Campos A (2009) Optimization strategies for non-linear material parameters identification in metal forming problems. Proceedings of EUROMECH Solid Mechanics Conference 2009, Lisbon, Portugal.

IV Inverse Problems of Blank and Tool Design

Chapter 11

Blank Shape Design – Influence of the Geometry Definition

A numerical procedure for the blank shape design is described and studied. Considering the proposed methodology, the design of a carter blank is presented. The other main objective is the study of the influence of the initial geometry definition in the optimization process. This study is performed considering the NURBS formulation to model the blank upper surface that will be changed during the optimization process.

11.1 Introduction

The interest of the stamping industry in numerical simulation of sheet metal forming, including inverse engineering approaches, is increasing. This fact occurs mainly because trial and error design procedures, commonly used in the past, are no longer economically competitive. The use of simulation codes is currently a common practice in the industrial forming environment, as the results typically obtained by means of Finite Element Method (FEM) software codes are well accepted by both the industrial and scientific communities.

In general terms, sheet metal forming is a complex deformation process controlled by parameters such as blank shape, tools' geometry, sheet thickness' values, blank holding force, friction, *etc*. [1]**.** Due to its complexity, and the higher combination level of all these input variables, optimization procedures are a fundamental tool in the proper design of the process parameters, accounting for the prediction and correction of undesirable forming defects such as fracture, springback, wrinkling, shape deviations and unbalanced residual stresses [2].

Within this context, the initial blank shape can be set as one of the most important process parameters to be considered, with a direct influence on the quality of the finished part. In this sense, accounting for an initial shape design procedure can effectively reduce the final cost and the product development time associated with the plastic forming operations [2]. The optimum blank shape improves as well the deep drawing quality, thickness distribution, formability of the part and minimizes the forming defects [3]. Therefore, many initial blank design methodologies have been reported in the literature to determine optimum initial blank shape profiles for diverse constraints and objective functions [1]. It is possible to categorize some of these methods in *deformation path iteration* approaches [4], *backward tracing* methods [5-7], *ideal* forming [4,8], *inverse approach* (IA) [2,3,9,10], *slip line field* method [11-14], *geometrical mapping* [15]**,** *sensitivity analysis* method [16,17], and, finally*, trial and error methods* based on FEA [18-22].

In order to solve initial blank design optimization problems a methodology that couples a FEM software with an optimization algorithm is needed. It is then necessary to develop an interface that connects both software's [23]. This connection should be performed considering a stage called *geometry parameterization and discretization,* where the optimization variables that define the geometry are transformed into nodes (and elements) in order to be properly used in the numerical simulation

The shape (geometry) of three-dimensional pieces can be defined by their surfaces. Therefore, shape optimization or initial geometry problems can be reduced to surface definition optimization problems where optimization variables are defined concerning the selected surface type. The mathematical surface representation can be performed *explicitly*, *implicitly* or *parametrically*. The non-uniform rational B-spline (NURBS) surfaces are parametric surface representations, such as Bézier surfaces, that can be defined by a relatively small set of variables (control vertices) and are appropriate to this kind of problem [24]. Nevertheless, their use and definition can still influence the final optimized result. Therefore, in the present work this approach is adopted, and the sensitivity of the obtained results to the parametric surface is studied for initial geometry problems [23]. Doing so, the superplastic forming of a metallic carter [1] is used as benchmark problem, considering four distinct NURBS control vertices distributions and an improved objective function, and leading to a more robust and comprehensive approach when compared to preliminary results by the authors [23]. Although being a conceptual problem, the idea is that it could drive the formulation and systematization of a mathematical inverse methodology that could be extrapolated to other industrial benchmarks.

11.2 Mechanical problem

An important problem in sheet metal forming is the design of the initial blank geometry, with interest in studying the influence of the initial geometry on the optimization process. In order to perform this study, the case study of a superplastic forming of a carter [25] was selected.

In Figure 11.1 it is possible to observe the geometry of the die and the metal blank considered in the present work.

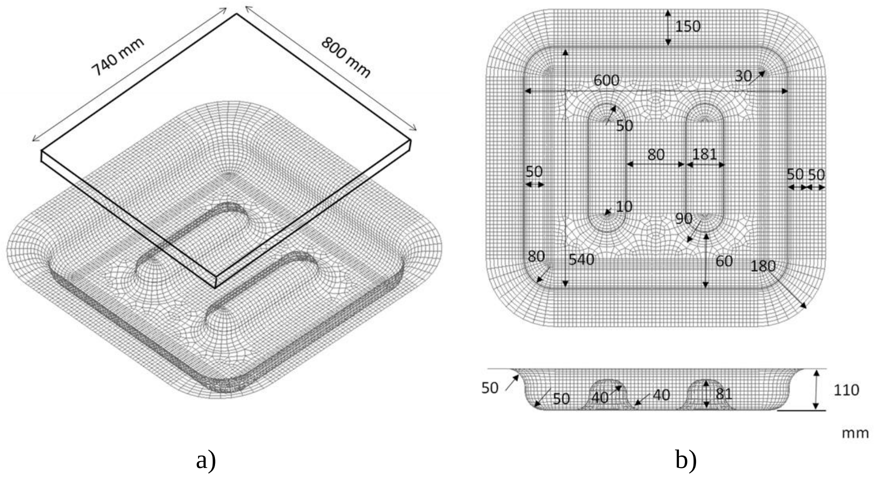


Figure 11.1 - a) Blank and b) die geometry definitions [19].

The material used in the blank is assumed to have an elasto-viscoplastic behaviour, with properties that roughly represent the 2004 based commercial superplastic aluminium alloy Supral 100 (Al-6Cu-0.4Zr) at a temperature of 470 °C. The material presents a Young's modulus of about 71 GPa and a Poisson's ratio of 0.34. The flow stress is assumed to follow a power law strain hardening function, given by [26]:

$$\dot{\bar{\varepsilon}}^{cr} = \left(A(\bar{\sigma}^{d})^{n}[(m+1)\bar{\varepsilon}^{\mathrm{cr}}]^{m}\right)^{\frac{1}{m+1}}, \tag{11.1}$$

where A, m and n are material parameters equal to 3.41x10^{-11}, 2 and 0, respectively. Also $\dot{\bar{\varepsilon}}^{\mathrm{cr}}$ represents the uniaxial equivalent creep strain rate and is equal to $\sqrt{{}^{2}/_{3}\,\dot{\varepsilon}^{cr}\!:\dot{\varepsilon}^{cr}}$, $\bar{\sigma}^{d}$ is the uniaxial equivalent deviatoric stress and $\bar{\varepsilon}^{\mathrm{cr}}$ is the equivalent creep strain.

For this problem, the blank thickness was initially considered uniform and equal to 4 mm. A maximum pressure of 100 MPa was considered. Other technological variables were not considered in the scope of the present example.

With this scenario, after the whole blank is formed it is possible to verify that the strain fields are not uniform in the whole surface, then leading to a non-homogeneous thickness distribution in the final geometry (Figure 11.2). This non-uniform thickness distribution can lead to thinning zones where subsequent rupture is prone to happen.

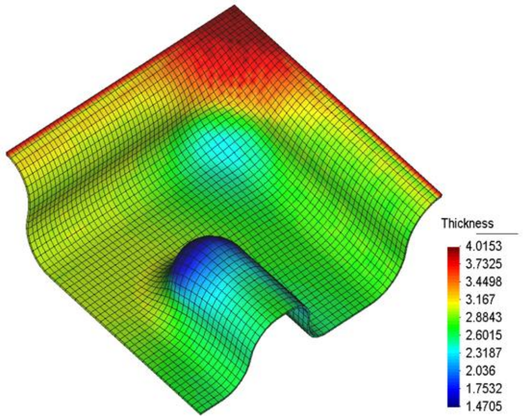


Figure 11.2 - Final thickness distribution of the FE analysis.

To avoid this problem, and in the context of the inverse methodology proposed in the present work, an initial (undeformed) blank with non-uniform thickness is conceptually proposed in order to obtain a regular thickness profile of the final component, even with non-homogeneous strains. This non-uniform thickness blank, a tailored-blank, can be achieved by means of, for instance, metal casting (foundry) or forging operations. Therefore, the final uniform thickness profile can be defined a-priori and set up to a specific value which, for this study, was chosen to be equal to 4 mm.

11.3 Sensitivity analysis on numerical parameters

A sensitivity analysis on the numerical parameters of the process assumes an important role on the understanding and improvement of a modelling process, since it is essential to determine the influence of such parameters, in order to improve the results [27]. In this section two sensitivity studies are presented, namely, the influence of the finite element size in the results of the numerical simulation of forming process, as well as the influence of the blank surface definition in the optimization process.

11.3.1 FE mesh size

The chosen finite element size is a numerical parameter that, usually, has a noticeable influence on the solution outcome. A finer mesh can improve the results accuracy when compared to a coarse mesh, however it is computationally more expensive. Therefore, an optimum balance between the finite element size and the computational requirements should be achieved [27].

To perform the present analysis, the forming process was carried out with six different meshes that differ on the total number of elements for the discretization of the blank mesh. Distinct meshes consisting of 1845, 2750, 3300, 5032, 6478 and 7380 elements with uniform distribution were considered, with one element through the thickness direction. The element type considered was C3D8 (8-node, trilinear hexahedral with full integration) [26].

Figure 11.3 shows the evolution of the pressure needed for the blank forming as a function of the z-displacement (forming direction), as measured at the blank geometric centre.

Figure 11.4 shows the pressure needed for the maximum z-displacement with the number of finite elements and Figure 11.5 represents the simulation CPU time with the number of finite elements.

Considering the graph from Figure 11.3, it is possible to conclude that the influence of the element size in the applied pressure have no significant difference for the studied meshes. The pressure needed to achieve the maximum displacement seems to have converged considering the results from Figure 11.4. However, it was for meshes 2 and 4 that less pressure was needed in order to achieve the same displacement. Based on this fact, and from the analysis of Figure 11.5 (where it is noticeable that the CPU time increases with the number of elements), mesh 2 was chosen as the reference one to be adopted in this work. The mesh 2 has 55x50x1 (2750) elements, respectively in the highest, lowest dimension and thickness.

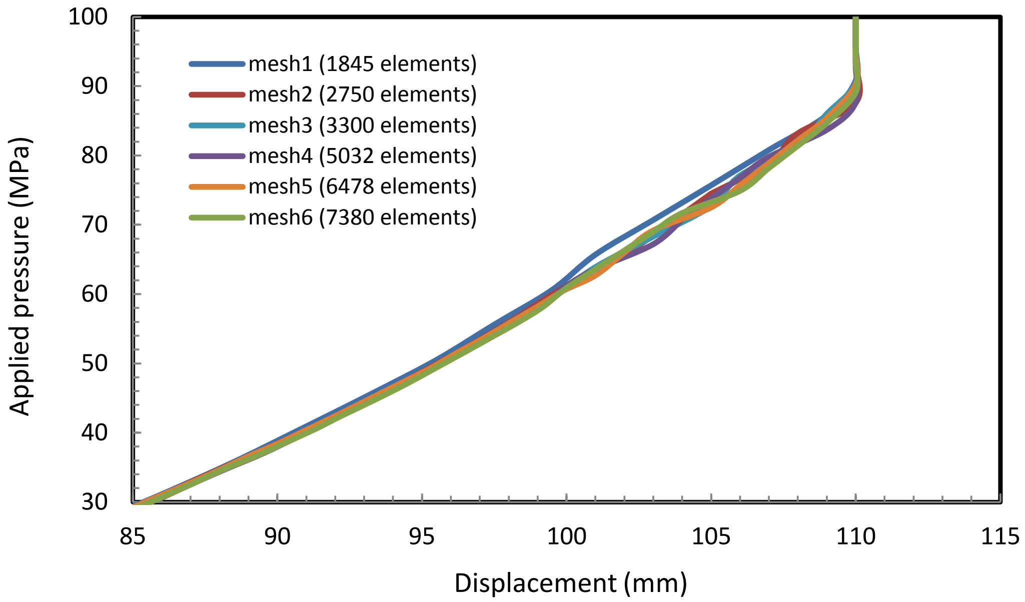


Figure 11.3 - Evolution of the applied pressure for different meshes in the carter forming process.

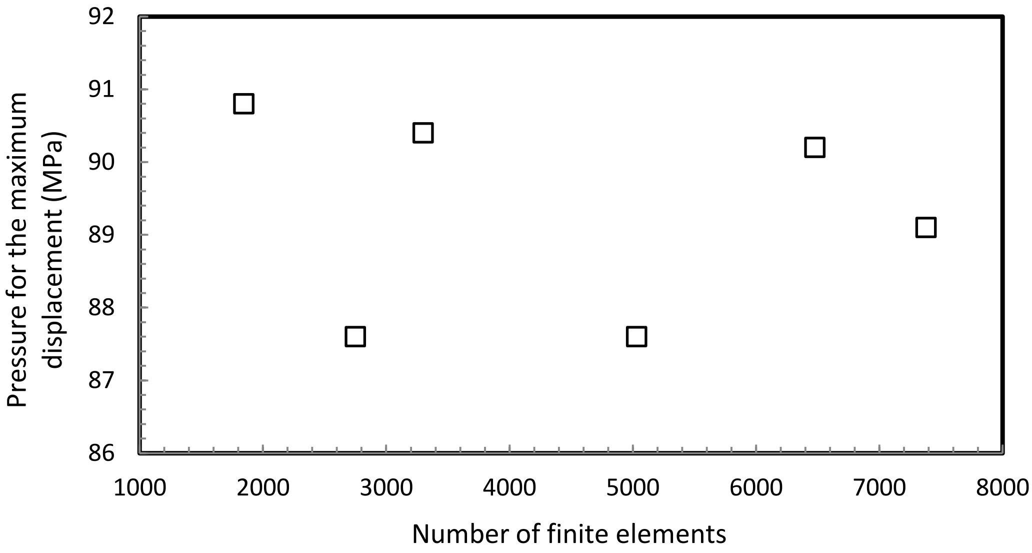


Figure 11.4 - Influence of the number of finite elements in the pressure needed for the maximum displacement.

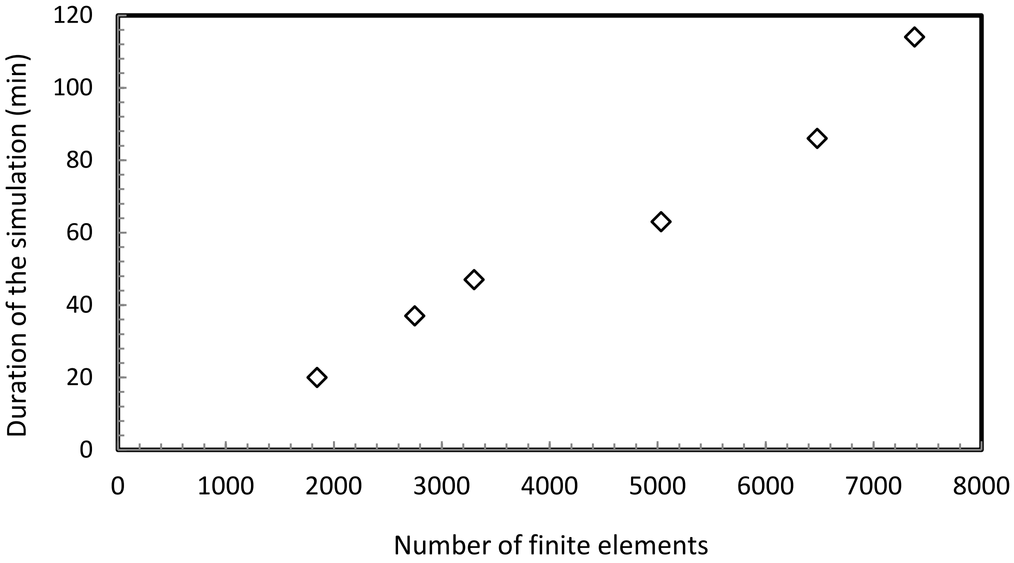


Figure 11.5 - Simulation CPU time with the number of finite elements.

11.3.2 Initial geometry definition

In order to understand the initial geometry influence in the optimization process the blank upper surface was defined by means of a NURBS surface, where the z-coordinates of specific NURBS control vertices were taken as optimization variables.

Considering the symmetry conditions of this problem, only one quarter of the die and the blank was considered during the FE simulation. The blank mesh was achieved in order to have a good description of the mechanical problem allied to the less CPU time for the FE simulation (direct problem), as detailed in the previous section. An efficient combination of these two factors is important in this sort of problems in order to reduce the time required to solve the inverse problem.

Regarding the initial geometry influence, four different studies were performed. These studies differ in the number of control vertices that formulates the NURBS surface, thus leading to different locations of the NURBS control vertices. The studies in the present work considered 9, 16, 25 and 36 optimization variables, corresponding to 16, 25, 36 and 49 control vertices, respectively, in the NURBS surface formulation. In all studies, the distribution of the control vertices was considered to be uniform in the mesh, as seen in Figure 11.6. The control vertices located in the free edges of the blank (as an example for the first study *d4,d8,d12,d13,d14,d15,d16*) are not optimized in this example. A uniform open knot vector was assumed and the weighting factors h_{ij} were considered to be equal to one in the formulation of the NURBS surface.

This parametric definition allows for a sensitivity study focusing in the influence of the number and the localization of the control vertices in the optimization process.

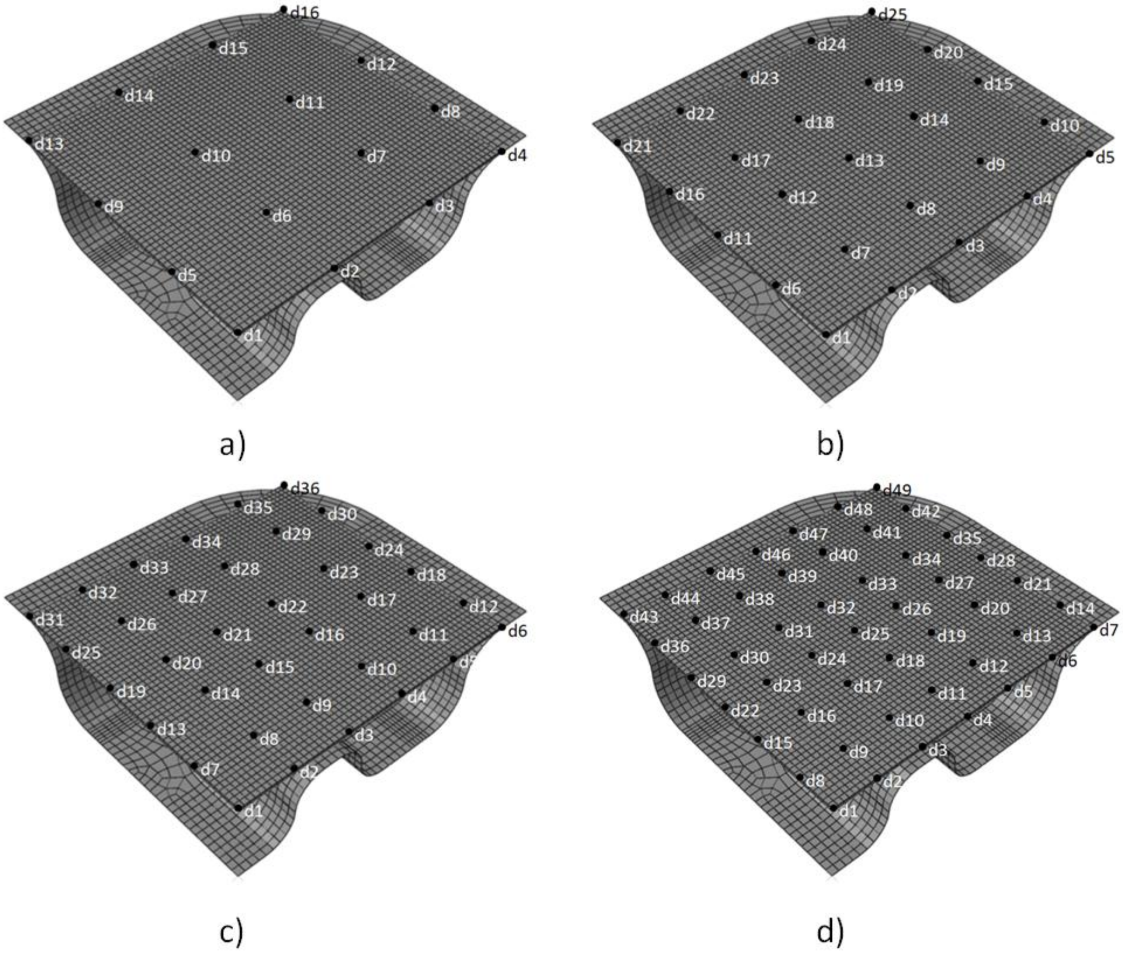

d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 d16
a)
d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 d16 d17 d18 d19 d20 d21 d22 d23 d24 d25
b)
d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 d16 d17 d18 d19 d20 d21 d22 d23 d24 d25 d26 d27 d28 d29 d30 d31 d32 d33 d34 d35 d36
c)
d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 d16 d17 d18 d19 d20 d21 d22 d23 d24 d25 d26 d27 d28 d29 d30 d31 d32 d33 d34 d35 d36 d37 d38 d39 d40 d41 d42 d43 d44 d45 d46 d47 d48 d49
d)

Figure 11.6 - NURBS control vertices distribution: a) 16 control vertices, b) 25 control vertices, c) 36 control vertices and d) 49 control vertices.

The algorithm applied in the NURBS surface implementation is listed in Table 11.1.

11.4 Principle of the blank optimization procedure

The presented optimization procedure was performed using a combination of a finite element commercial program and an optimization software code. The FE analysis was performed with Abaqus® [26], while the optimization process was carried out with the *SDL* optimization software [28]. The communication between these two programs was ensured by a interface developed in FORTRAN, where the new geometry is parameterized and then the relevant input variables written in the finite element input file. This numerical procedure can run with no operator intervention once it has started. Figure 11.7 illustrates the numerical procedure performed in order to achieve the optimum blank shape, which is based on inverse and incremental approach in combination with the optimization algorithm.

Table 11.1 - NURBS algorithm used in the carter forming process [18].

```
Specify the number of control vertices in the $u$ and $w$ directions.

Specify the order of the surface in each of the $u$ and $w$ directions.
Specify the number of isoparametric lines in each of the $u$ and $w$ directions.
Specify the control net.
Calculate the uniform open knot vector in the $u$ direction.
Calculate the uniform open knot vector in the $w$ direction.
For each parametric $u$ value
    Calculate the basis functions $N_{i,k}(u)$.
    For each parametric value $w$
        Calculate the basis functions $M_{j,l}(w)$.
        Calculate the Sum function.
        For each control vertex in the $u$ direction
            For each control vertex in the $w$ direction
                Calculate the surface point $Q(u,w)$.
            end loop
        end loop
    end loop
end loop
```

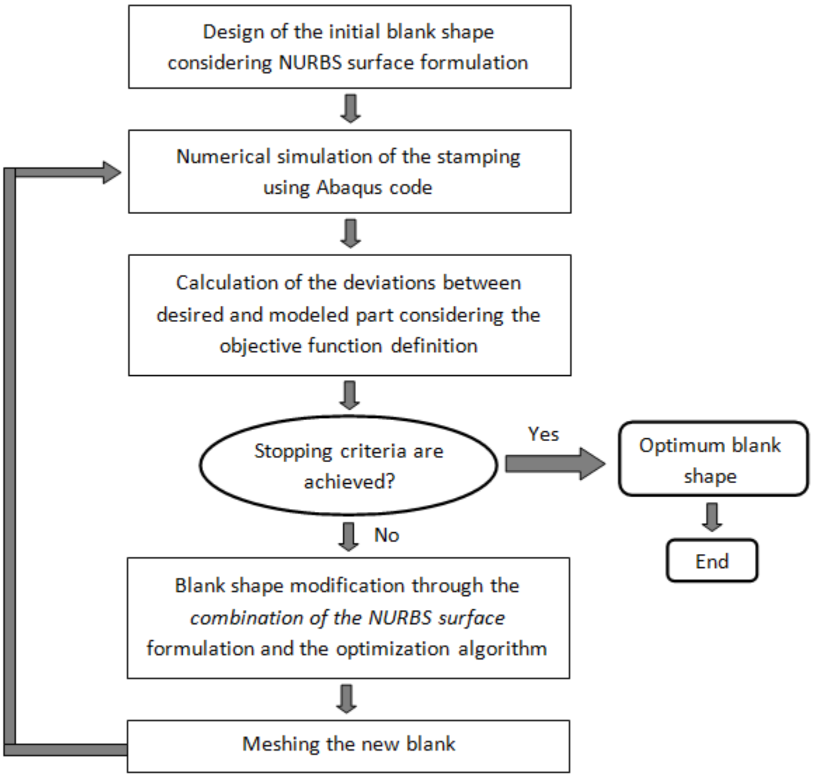

Design of the initial blank shape considering NURBS surface formulation
Numerical simulation of the stamping using Abaqus code
Calculation of the deviations between desired and modeled part considering the objective function definition
Stopping criteria are achieved?
Yes
Optimum blank shape
End
No
Blank shape modification through the *combination of the NURBS surface* formulation and the optimization algorithm
Meshing the new blank

Figure 11.7 - Initial blank shape optimization procedure.

The initial process parameters (tools geometry, mechanical properties of the blank sheet, blank holder force, *etc*.) are constant during the optimization process. Also, and as stated before, an initial value for the blank thickness is given (4 mm) which, when combined with the NURBS surface formulation, will allow to iteratively re-design the initial blank shape.

The parameterized initial blank shape is then exported to the FE code Abaqus® in order to perform the numerical simulation of the stamping plastic forming process, in this case with the use of an implicit algorithm (Abaqus® Standard). After the first simulation, the calculation of the deviations between desired and modelled part is performed considering the objective function. If the stopping criterion is achieved then all the process could be stopped and the optimum initial blank shape is assumed to have been found. Otherwise the blank shape is modified through the combination of the NURBS surface formulation and the optimization algorithm. Afterwards, the new blank is remeshed and used for the second simulation procedure. The same calculation is repeated for the following steps until the stopping criterion is finally achieved.

The success of the optimization procedure strongly depends on the correct definition of the objective function, which turns to be essential to the efficient determination of the optimization variables effectively minimizing the objective function [28]. One of the most used objective functions consists on the sum of squares and, for this specific case the problem can be defined as:

$$\min \qquad F = \frac{\sum_{i=1}^{n_{\text{snodes}}} (e_i - e_{\text{opt}})^2}{n_{\text{snodes}}} \quad , \tag{11.2}$$

$$\text{subjected to:} \qquad 1 < e_{\text{ini}} < 9$$

in which n_{snodes} is the number of surface nodes, e_i are the observable relevant variables (in the present context, the final thickness of each element of the blank) and e_{opt} corresponds to the optimum (required) final thickness (that is 4 mm). Also, e_{ini} is the set of the optimization variables that represent the initial thickness of the blank, i.e. the thickness before the forming process. Figure 11.8 shows the procedure adopted for the final thicknesses calculation.

In this work, a Levenberg-Marquardt gradient algorithm [29] was used to minimize the objective function. Considering the least-squares type of the LM algorithm, the objective function choosen (sum of squares, where $n_{\text{snodes}} > n_{\text{op.variables+1}}$) guarantees the convergence of the optimization problem. It was considered as the stooping criterion a maximum number of 300 iterations or, as an alternative, a stagnation value of 1x10^{-30}, in terms of objective function value between each iteration. The objective function sensibilities are calculated by means of forward finite differences and the perturbation size considered is 5x10^{-3}.

11.5 Results and discussion

In this section the results that synthesize the current study are presented and discussed. In Figure 11.9 it is possible to observe the evolution of the normalized objective function value with the iteration number, for the different analyses. In Figure 11.10 it is presented the evolution of the optimization variables and the objective function with the optimization process, for the study with 9 optimization variables.

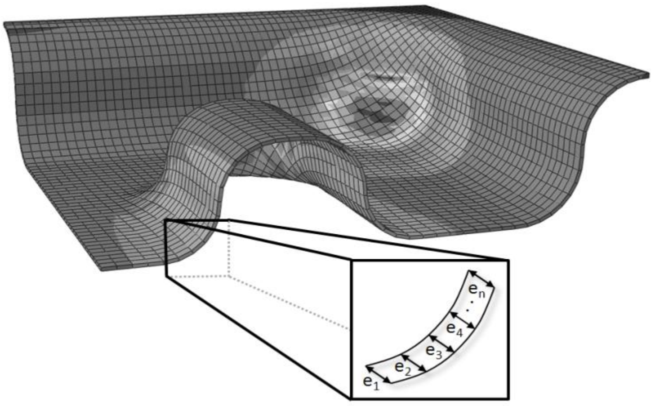


Figure 11.8 - Final thicknesses calculation.

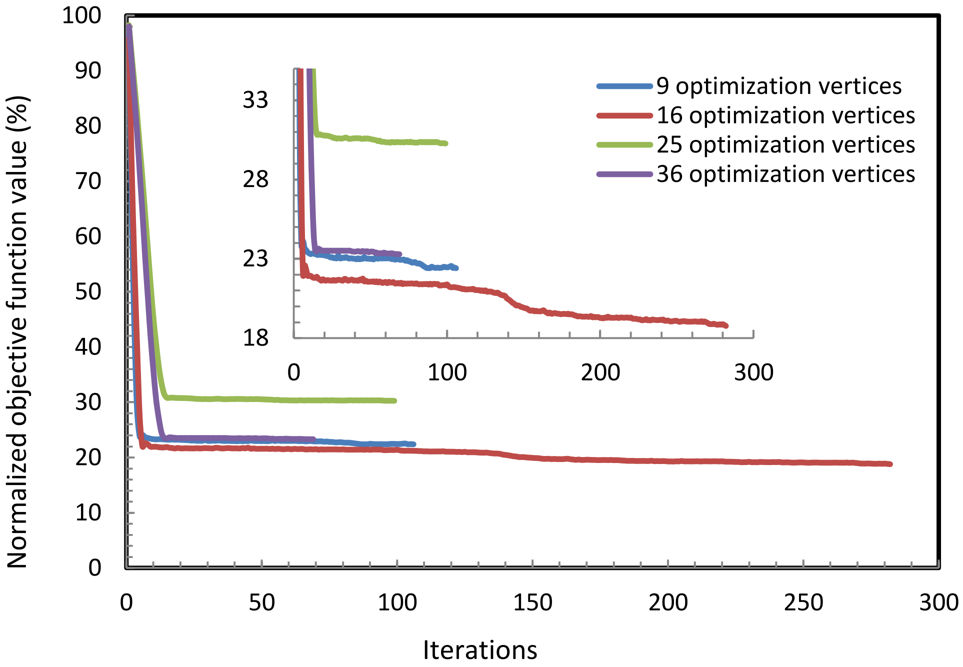


Figure 11.9 - Evolution of the normalized objective function value.

In Table 11.2 the values of the optimum variables and the respective objective function, for each study, are presented.

Regarding the results presented in Figure 11.9, as well as the values of the objective function from Table 11.2, it is possible to verify that the four studies considered have achieved good results in terms of objective function values. The objective function values show decreases that vary from 69.7%, for the 25 optimization vertices, to 81.2%, for the 16 optimization vertices, relative to the initial trial simulation (considering a blank with a uniform 4 mm thickness).

It is possible to observe that the best value of objective function was achieved for the study with 16 optimization variables. However, for this study, the optimization method has needed twice as many iterations to reach stagnation. Also, a tendency for the evolution of the objective function, with the optimization variables number, was not clearly verified.

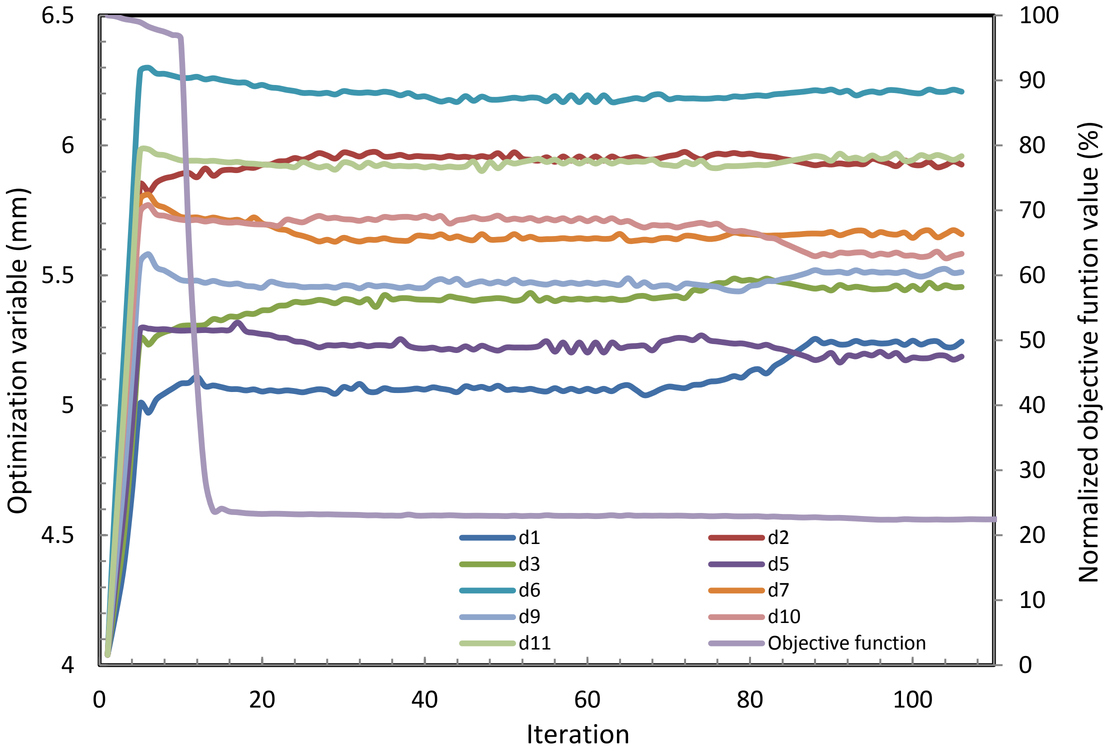


Figure 11.10 - Evolution of the optimization variables z-coordinate for the study with 9 optimization variables.

Table 11.2 - Values of the optimization variables (in millimetres) and the objective function for the best iteration.

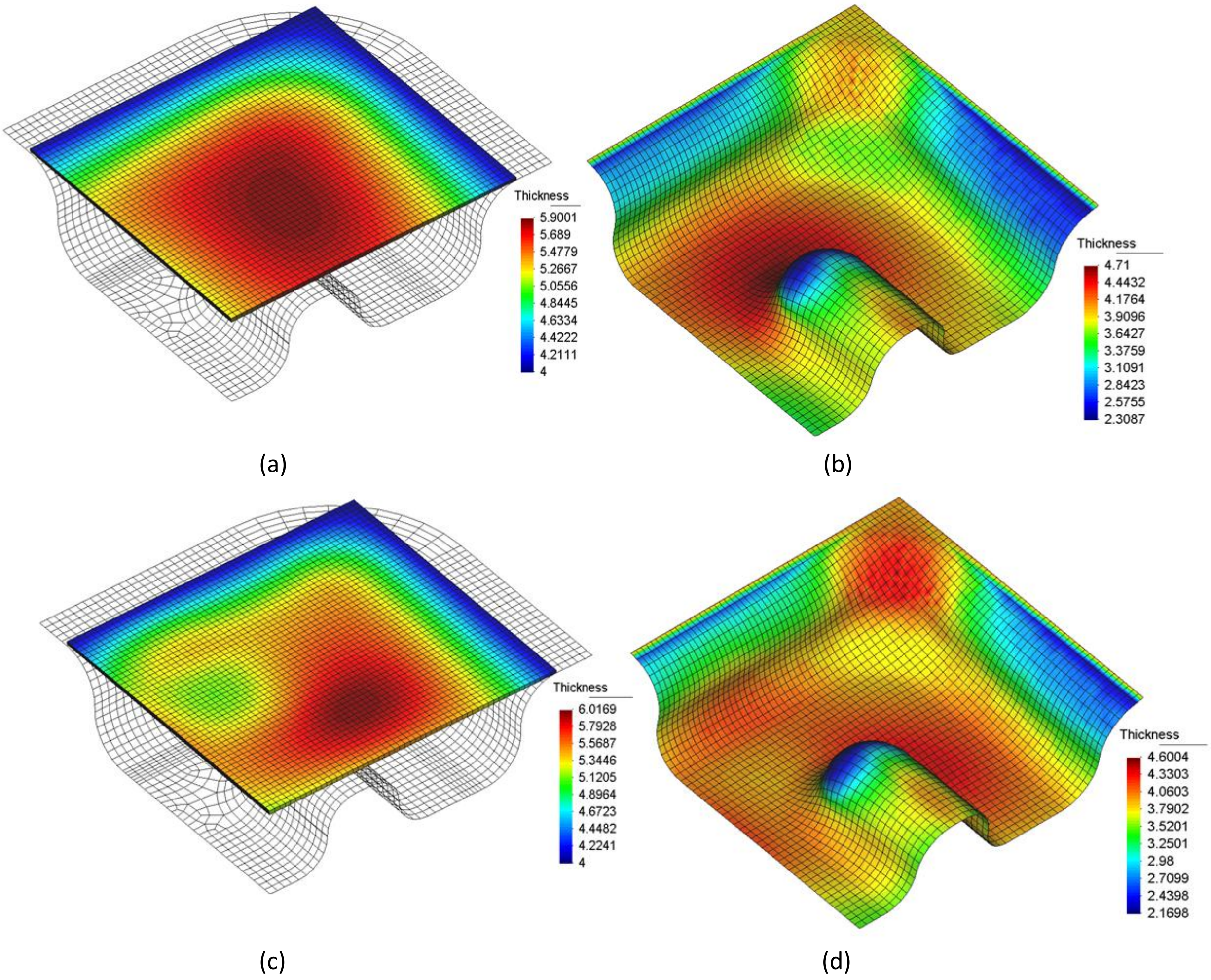
Op.var.	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	...
9vertices	5.2448	5.9266	5.4551	-	5.1874	6.2075	5.659	-	5.5125	5.5826	5.9584	-	-	...
16vertices	5.1889	5.2901	5.5974	5.3058	-	5.5495	6.0632	6.4116	5.2089	-	5.5134	4.4786	5.8311	...
25vertices	3.7866	4.4368	6.0688	4.3885	3.5395	-	4.6669	5.9201	6.8653	6.1069	4.8017	-	6.2191	...
36vertices	4.5273	4.0089	4.7542	4.7814	5.6966	6.0772	-	5.0606	4.5223	7.8016	6.6657	4.9714	4.9598	...

Op.var.	d14	d15	d16	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26	...
9vertices	-	-	-	-	-	-	-	-	-	-	-	-	-	...
16vertices	5.4825	-	4.9893	6.0191	5.1159	5.5671	-	-	-	-	-	-	-	...
25vertices	5.5936	5.0101	5.777	5.7575	-	6.0147	5.0224	5.8394	3.2367	6.8665	-	4.4438	4.2894	...
36vertices	4.0746	7.4977	4.8431	4.2511	6.2052	-	6.7544	4.3991	5.3244	6.8045	5.3746	5.5439	-	...

Op.var.	d27	d28	d29	d30	d31	d32	d33	d34	d35	d36	d37	d38	d39	F (%)
9vertices	-	-	-	-	-	-	-	-	-	-	-	-	-	22.4
16vertices	-	-	-	-	-	-	-	-	-	-	-	-	-	18.8
25vertices	5.7769	4.9559	5.1153	-	-	-	-	-	-	-	-	-	-	30.3
36vertices	4.6045	4.0831	5.9371	4.002	5.6371	4.0436	-	6.6808	4.4406	6.3958	5.1132	6.1315	3.7955	23.3

The evolution of the optimization variables with the iteration number was presented only for the study considering 9 optimization variables (see Figure 11.10), being the evolution of the optimization variables for the other studies similar. It is possible to observe that the highest variation in this values occur in the first iterations, leading to the highest decrease of the objective function. It is also possible to observe that after these initial iterations the optimization variables remain approximately constant until approximately iteration 85, where the optimization variables change significantly. However, this variation didn´t lead to considerable changes in the objective function value.

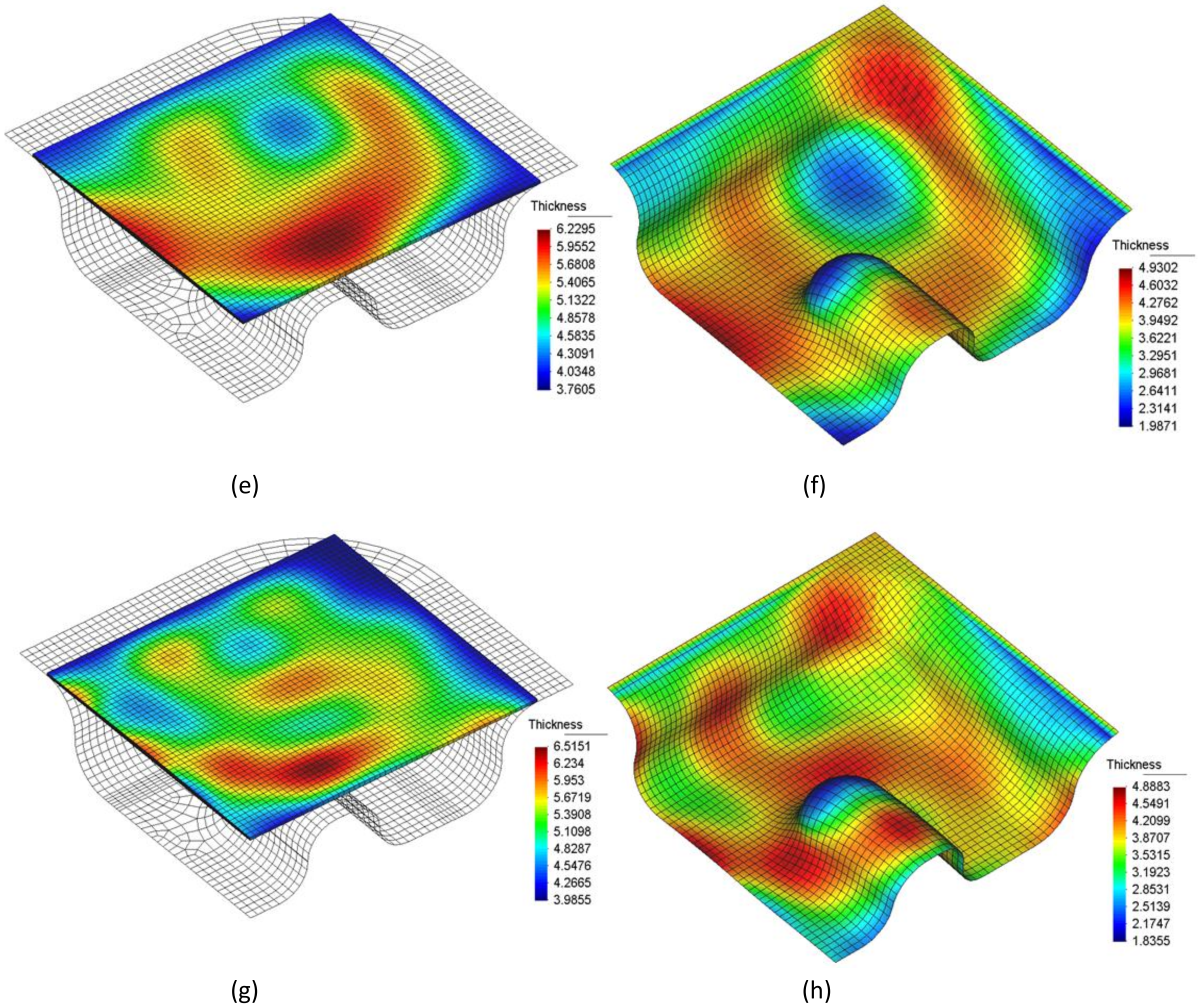
Figure 11.11 and Figure 11.12 presents the contour graphs for the thickness of the blank (in millimetres). Both initial and final thicknesses are presented for the best iteration of each study. In Figure 11.13, the blank thickness range evolution with the number of optimization variables is presented. Thickness range was considered to be the difference between the higher and the lower values of thickness for a specific blank.



(a) (b)

(c) (d)

Figure 11.11 - Blank thickness of the best iteration for the: a) initial and b) final blank for 9 optimization variables, c) initial and d) final blank for 16 optimization variables.



(e) (f)

(g) (h)

Figure 11.12 - Blank thickness of the best iteration for the: e) initial and f) final blank for 25 optimization variables, g) initial and h) final blank for 36 optimization variables.

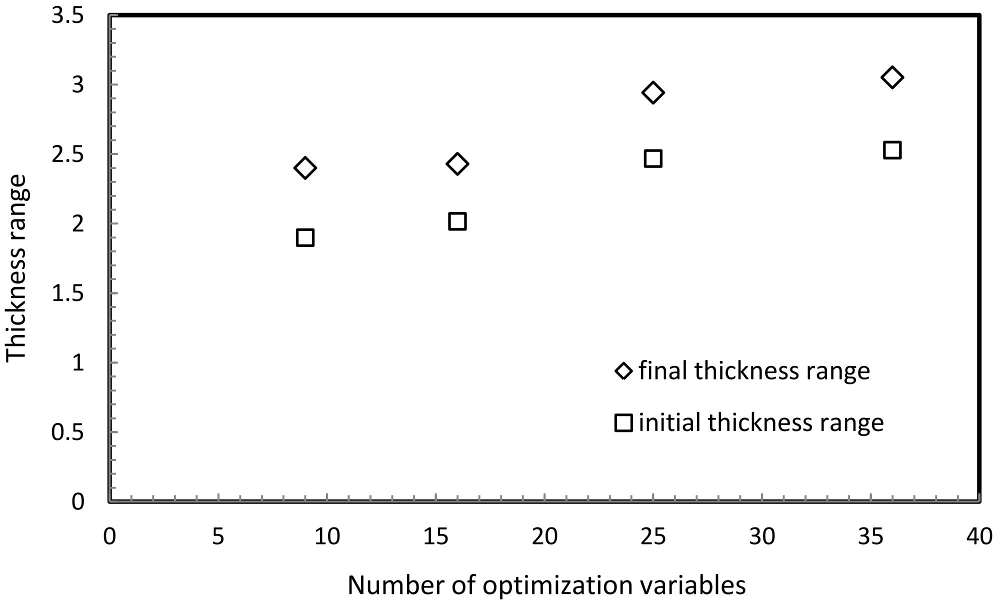


Figure 11.13 - Blank thickness range of the best iteration.

From the results presented in Figure 11.11 and Figure 11.12, it is possible to conclude that increasing the number of optimization variables and, consequently, the number of control vertices allows the formulation of a more complex shape for the initial surface of the blank. Furthermore, in Figure 11.13 it is possible to observe that if the number of optimization variables increases, then the thickness range also increases for the initial and final blanks. This fact was expected considering that increasing the optimization variables the complexity of the surface can also increase.

Considering the final thickness of the optimized blanks (Figure 11.11 and Figure 11.12 b), d), f) and h)) and comparing those with the final thickness of the blank that have an initial thickness of 4 mm (Figure 11.2) it is possible to observe that the minimum thickness value had increased for all the optimized blanks. This minimum point is located in the same region for the four studies, as it is possible to observe in Figure 11.11 and Figure 11.12.

It was for the study considering 9 optimization variables that the highest values of the minimum thickness were observed. However, comparing this fact with the results concerning the objective function value evolution, it is seen that when dealing with 16 optimization variables then a minimum value of objective function was achieved.

It is possible to conclude that the increase from 9 to 16 optimization variables leads to better results, when considering the value of the objective function. The increase to 25 and 36 optimization variables, on the other side, has lead to a more complex and flexible initial surfaces of the blank. However, and taken into account the objective function results, no improvements were inferred. This fact may occur due to the fact that, when increasing the number of optimization variables, the vertices are not granted to be located in the same places for the distinct meshes, considering that they can be in more critical areas.

These results and discussion are for this specific problem. Each stamping problem has different thinning zones where subsequent rupture is prone to happen, and therefore ideal optimization variables location varies between different problems. For other mechanical problems this method can be considered, however the initial geometry definition should be analysed in detail.

11.6 Conclusions

A sensitivity study was performed considering the influence of the finite element mesh size in the forming process and the initial geometry definition in the optimization process. Considering the mesh studied, a blank mesh that has a good description of the mechanical problem allied to the less CPU time for the FE simulation was achieved.

The influence of the geometry definition in a carter forming process was presented. Four studies were performed with differences in the number and location of control vertices that formulate the NURBS surface. The studies considered 9, 16, 25 and 36 optimization variables that correspond respectively to 16, 25, 36 and 49 control vertices in the NURBS surface formulation. In all the studies the location of the control vertices were considered to be uniform in the mesh. The four studies had achieved good values for objective function, having a minimum decrease of 69.7% for the 25 optimization vertices and a maximum decrease of 81.2% for the 16 optimization vertices, relative to the initial simulation that considered a blank with a uniform 4 mm thickness. It was for the study with 16 optimization variables that the best value of objective function of 18.8% was achieved.

References

[1] Padmanabhan R, Oliveira M, Baptista A, Alves J, Menezes L (2009) Blank design for deep drawn parts using parametric NURBS surfaces. Journal of Materials Processing Technology 209:2402-2411.

[2] Azaouzi M, Belouettara S, Rauchsa G (2011) A numerical method for the optimal blank shape design. Materials & Design 32:756-765.

[3] Fazli A, Arezoo B (2012) A comparison of numerical iteration based algorithms in blank optimization. Finite Elements in Analysis and Desing 50:207-216.

[4] Park S, Yoon J, Yang D, Kim Y (1999) Optimum blank design in sheet metal forming by the deformation path iteration method. International Journal of Mechanical Sciences 41:1217-1232.

[5] Ku T, Lim H, Choi H, Hwang S, Kang B (2001) Implementation of backward tracing scheme of the FEM to blank design in sheet metal forming. Journal of Materials Processing Technology 111:90-97.

[6] Kang B, Lee J, Choi H (1997) Extension for the backward tracing scheme of the rigid-plastic FEM in three-dimensional deformation. Journal of Manufacturing Science and Engineering 119:556-562.

[7] Choi H, Park M, Kang B (1998) Design of a piercing hole in coining process by the three-dimensional backward tracing scheme of the FEM. Journal of Materials Processing Technology 81:113-120.

[8] Chung K, Yoon J, Richmond O (2000) Ideal sheet forming with frictional constraints. International Journal of Plasticity 16:595-610.

[9] Naceur H, Guo Y, Batoz J (2004) Blank optimization in sheet metal forming using an evolutionary algorithm. Journal of Materials Processing Technology 151:183-191.

[10] Wang J, Goel A, Yang F, Gau J (2009) Blank optimization for sheet metal forming using multi-step finite element simulations. International Journal of Advanced Manufacturing Technology 40:709-720.

[11] Karima M (1989) Blank development and tooling design for drawn parts using a modified slip line field based approach. Journal of Engineering for Industry 111:345-350.

[12] Chen X, Sowerby R (1996) Blank development and the prediction of earing in cup drawing. International Journal of Mechanical Sciences 8:509-516.

[13] Parsa M, Matin P, Mashhadi M (2004) Improvement of initial blank shape for intricate products using slip line field, Journal of Materials Processing Technology 145:21-26.

[14] Kuwabara T, Si W (1997) PC-based blank design system for deep-drawing irregularly shaped prismatic shells with arbitrarily shape flange, Journal of Materials Processing Technology 63:89-94.

[15] Sowerby R, Duncan J, Chu E (1986) The modeling of sheet metal stampings. International Journal of Mechanical Sciences 28:415-430.

[16] Shim H, Son K, Kim K (2000) Optimum blank shape design by sensitivity analysis. Journal of Materials Processing Technology 104:191-199.

[17] Son K, Shim H (2003) Optimal blank shape design using the initial velocity of boundary nodes. Journal of Materials Processing Technology 134:92-98.

[18] Pegada V, Chun Y, Santhanam S (2002) An algorithm for determining the optimal blank shape for the deep drawing of aluminium cups. Journal of Materials Processing Technology 125-126: 743-750.

[19] Azaouzi M, Naceur H, Delamézière A, Batoz J, Belouettar S (2008) An heuristic optimization algorithm for the blank shape design of high precision metallic parts obtained by a particular stamping process. Finite Element in Analysis and Design 44:842-850.

[20] Azaouzi M, Delamézière A, Batoz J, Belouettar S, Sibaud D (2008) Design of the blank contour for high precision metallic parts obtained by stamping. Proceedings of the 7th International Conference on Numerical Simulation of 3D Sheet Metal Forming Processes, NUMISHEET, Switzerland, Interlaken 705-708.

[21] Correia A, Alves J, Oliveira M, Padmanabhan R, Menezes L (2010) Towards a blank shape optimization: enhancement of procedure and FE solver. Proceedings of 2nd International Conference on Engineering Optimization, Lisbon, Portugal.

[22] Padmanabhana R, Oliveira M, Baptista A, L.Alves J, Menezes L (2009) Numerical study on the influence of initial anisotropy on optimal blank shape. Finite Elements in Analysis and Design 45:71-80.

[23] de-Carvalho R, Silva S, Valente R, Andrade-Campos A (2011) The geometry definition influence in inverse analysis - application to carter forming process. Proceedings of the 14th International ESAFORM Conference on Material Forming, Belfast, Ireland.

[24] Rogers D (2001) An Introduction to NURBS with Historical Perspective, Academic Press.

[25] ABAQUS 6.7 (2007) User Manual, Simulia Inc, Dassault Systémes.

[26] Kleinermann P, Ponthot J (2003) Parameter identification and shape/process optimization in metal forming simulation. Journal of Materials Processing Technology 139:521-526.

[27] Oliveira M, Padmanabhan R, Baptista A, Alves J, Menezes L (2009) Sensitivity study on some parameters in blank design. Materials & Design 30:1223-1230.

[28] de-Carvalho R, Valente R, Andrade-Campos A (2010) On the objective function evaluation in parameter identification of material constitutive models - Single-point or FE analysis. Proceedings of the 13th ESAFORM Conference on Material Forming, Brescia, Italy.

[29] Marquardt D (1962) An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics 11:431-41.

Chapter 12

Shape Optimization of a Forging Process

The shape optimization of a two-stage forging process is presented. In order to achieve a straight cylinder after the forging, two different approaches are analyzed. In the first one, the initial geometry of the cylinder is optimized and, in the other one, the shape of the first stage tool is optimized. To parameterize the free surface of the cylinder different methods are presented. Furthermore, in order to define the tool in the last example, also different parameterizations are presented.

12.1 Introduction

As in all mechanical processes, the achievement of final high-quality products is the keyword in metal forging processes [1]. Forging process belongs to the metal forming process category and, in this process, the change in the shape of the workpiece is made through the application of compressive forces [2]. Generally, in the design of a forging process, the only factors that are known are the material and the final shape of the component. Therefore, the engineer will have to design a process that makes the desired part subjected to limitations of shape, properties, cost, time, *etc.*. If the final component shape is complex, the workpiece won´t be deformed in one stage [2]. To solve this problem, two different approaches can be considered: the initial shape optimization of the workpiece or the optimization of a preform tool. The advantages of the initial shape optimization process were mentioned in the previous chapter. The preform tool design assumes as the most important step for product quality control [3]. It is considered that a reasonable preform tool shape reduces the raw material cost and improves material flow, reducing manufacturing cost and eliminating following processes. Consequently, preform tool shape optimization is still of considerable interest [1]. Several works in this preform tool optimization thematic can be found in [1,2,4-6].

The present chapter appears in the tool shape optimization thematic. The mechanical problem considered is the two stage forging of a cylindrical billet, one example merely academic, however that allows to validate the optimization methodologies that could be applied in real optimization processes [7,8].

It is well known that after the compression of a cylindrical billet, it gets a barrel shape. This phenomenon can be observed in the experimental study carried out by Santos *et al*. [8]. Aiming to avoid this phenomenon and get a straight cylindrical billet after the forging, two different approaches are presented. In the first one, the initial geometry of the billet is optimized and in the other one the shape of the first stage tool of a two stage forging process is optimized. These two inverses methodologies will be presented and discussed.

This is a problem that allows to create an inverse methodology to be used in tool optimization problems. However, it is known that the barrel shape problem in forging can be also technologically controlled by lubrication, by the roughness between the compression plates and by the difference of temperatures between the two compression plates, in the case of hot forging. When a perfect cylindrical billet, after the forging, is wanted it is also possible to use technological processes such as, for instance, precision forging. This process allows the reduction or elimination of finishing secondary operations and heat treatments, optimizing the raw material used, reducing the energy consumption and reducing or eliminating the costs associated to the transformation of the waste [9].

12.2 Initial shape optimization of a cylindrical billet in a two stage forging process

12.2.1 Mechanical problem

The present example consists in the compression of an axisymmetric cylindrical billet up to 60% of its initial 100 mm height [7,8]. The main mechanical properties of this problem are the same as the ones considered in section 6.3.1 and are presented in Table 12.1.

Table 12.1 - Mechanical properties of the billet [8].

Mechanical properties	
Density [kg/m3]	2710
Young´s modulus [GPa]	71
Yield stress [MPa]	100
Poisson´s ratio [-]	0.33
Friction coefficient [-]	0.1

The aim of this example is to modify the initial geometry of the billet in order to achieve a straight cylindrical billet after forging. The example is graphically described in Figure 12.1. The mesh considered in this problem is presented in Figure 12.2.

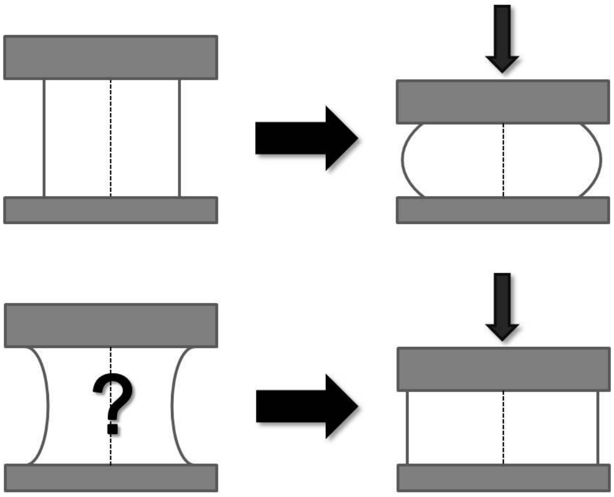


Figure 12.1 - Initial geometry optimization of a cylindrical billet [6].

12.2.2 Objective function definition

As it was mentioned in previous chapters the optimization procedure strongly depends on the correct definition of the objective function. Once again the objective function used was based in the work of Igor Grešovnik [7], where the objective function tries to find the minimum difference between the billet after the forming process and a straight cylindrical billet. This can be achieved by the minimization of the following objective function [7]:

$$D(\bar{u},\bar{v}) = D_u + D_v = \sum_{i=1}^{M} \frac{(u_i - \bar{u})^2}{\bar{u}^2} + \sum_{i=1}^{N} \frac{(v_i - \bar{v})^2}{\bar{v}^2} \ , \tag{13.1}$$

in which u_i are the coordinates of the nodes on the contact surface of the billet and v_i are the coordinates of the nodes on the free surface of the billet, and M and N are respectively the number of nodes on each surface of the billet. The values $\bar{u}$ and $\bar{v}$ are the average values of the correspondent coordinates and can be computed as:

$$\bar{u} = \sum_{j=1}^{M} \frac{u_j}{M} \quad \text{and} \quad \bar{v} = \sum_{j=1}^{N} \frac{v_j}{N} \, . \tag{13.2}$$

In the first approach, as the tools are both flats, D_u is equal to zero. This variable has significance for the second approach where the irregular shape of the first tool can conduct to a non-flat upper surface of the billet in the end of the forging process.

12.2.3 Parameterization influence

In the present section, the results obtained are presented and also a study of the parameterization influence is conducted. In order to minimize the previous objective function, a parameterization of the free surface of the billet was performed. A Levenberg-Marquardt gradient algorithm was used to minimize the objective function. It was considered that the algorithm has converged if a maximum number of 200 iterations or a stagnation value of 1x10^{-60} is achieved. In Figure 12.2 the 21 parameterized nodes are presented.

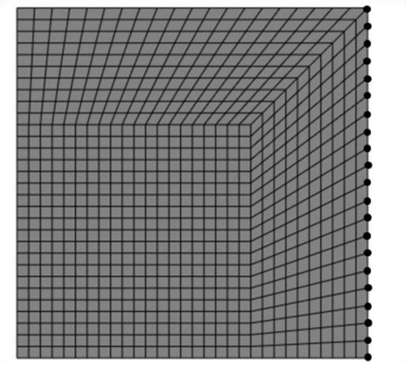


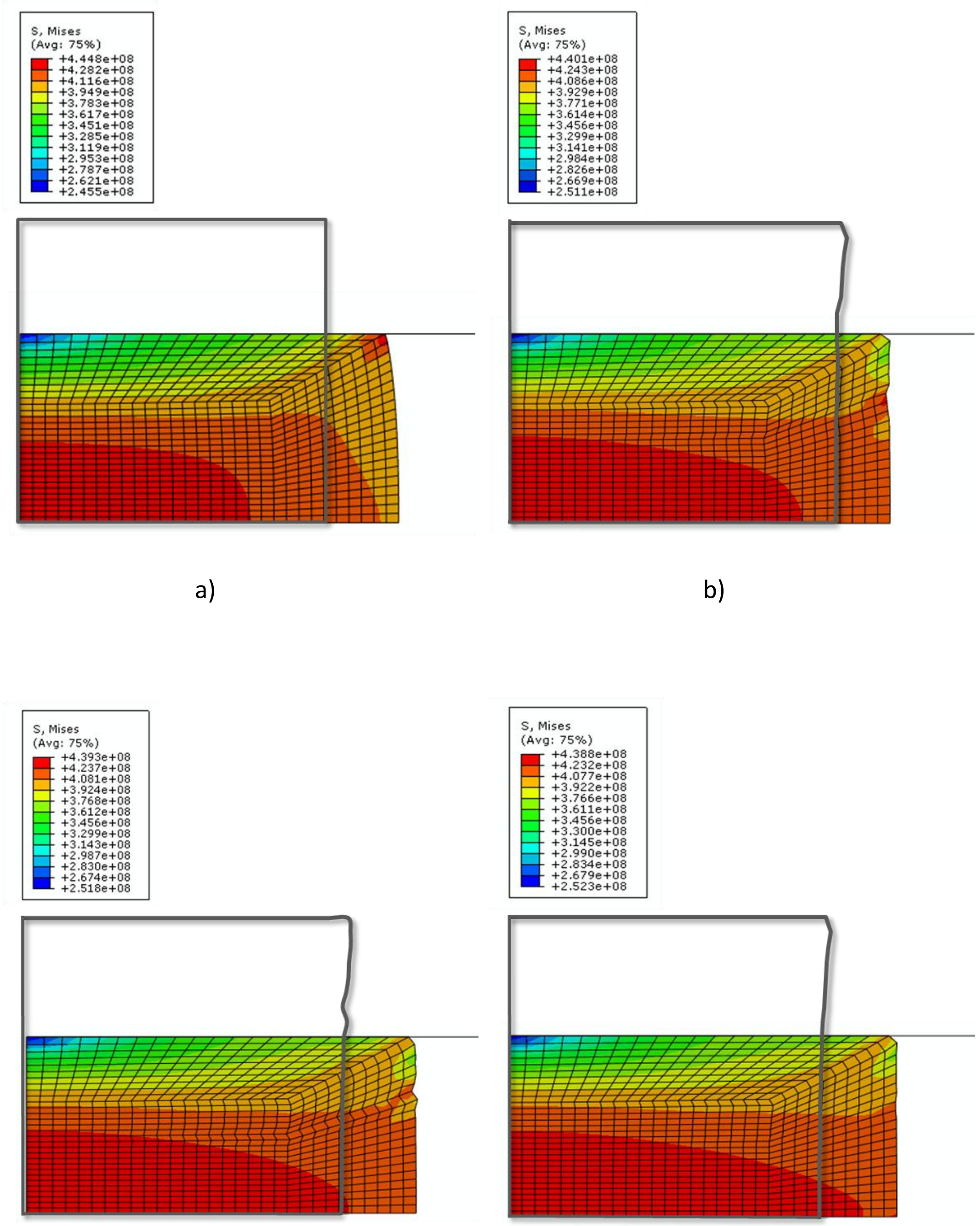
Figure 12.2 - The 21 parameterized nodes.

For the optimization process three different methods were used. The first method consists in the use of the x-coordinate of the 21 parameterized nodes as optimization variables. The results obtained for this method are presented in Figure 12.3 and this method will be referred as the *set of points method* for a better understanding. The initial (gray line) and the final results are presented for four different iterations during the optimization process. The iterations presented are the first, the optimum iteration and two iterations in between this two.

Following the same logic, in Figure 12.4 the results obtained considering for the parameterization a cubic Bézier curve are presented. In this case the optimization variables are the x-coordinates of four points equally spaced in the billet height. These four points are the control vertices that define the Bézier curve that passes through the 21 nodes mentioned before. This method will be referred as the *cubic Bézier method*.

Considering the same logic, in Figure 12.5 it is possible to observe the results when a cubic NURBS is considered for the parameterization. In this case, four points are also used, however the optimization variables are eight, four x-coordinates and four NURBS weights. This method will be referred as the *cubic NURBS method*.

In order to compare the results obtained with the three different parameterizations, it is possible to observe the evolution of the objective function during the optimization process in Figure 12.6 and Figure 12.7. Figure 12.7 has the same information than Figure 12.6, however has a zoom where the lowest values in terms of objective function value are reached. In Figure 12.8 it is possible to compare the three final geometries obtained.



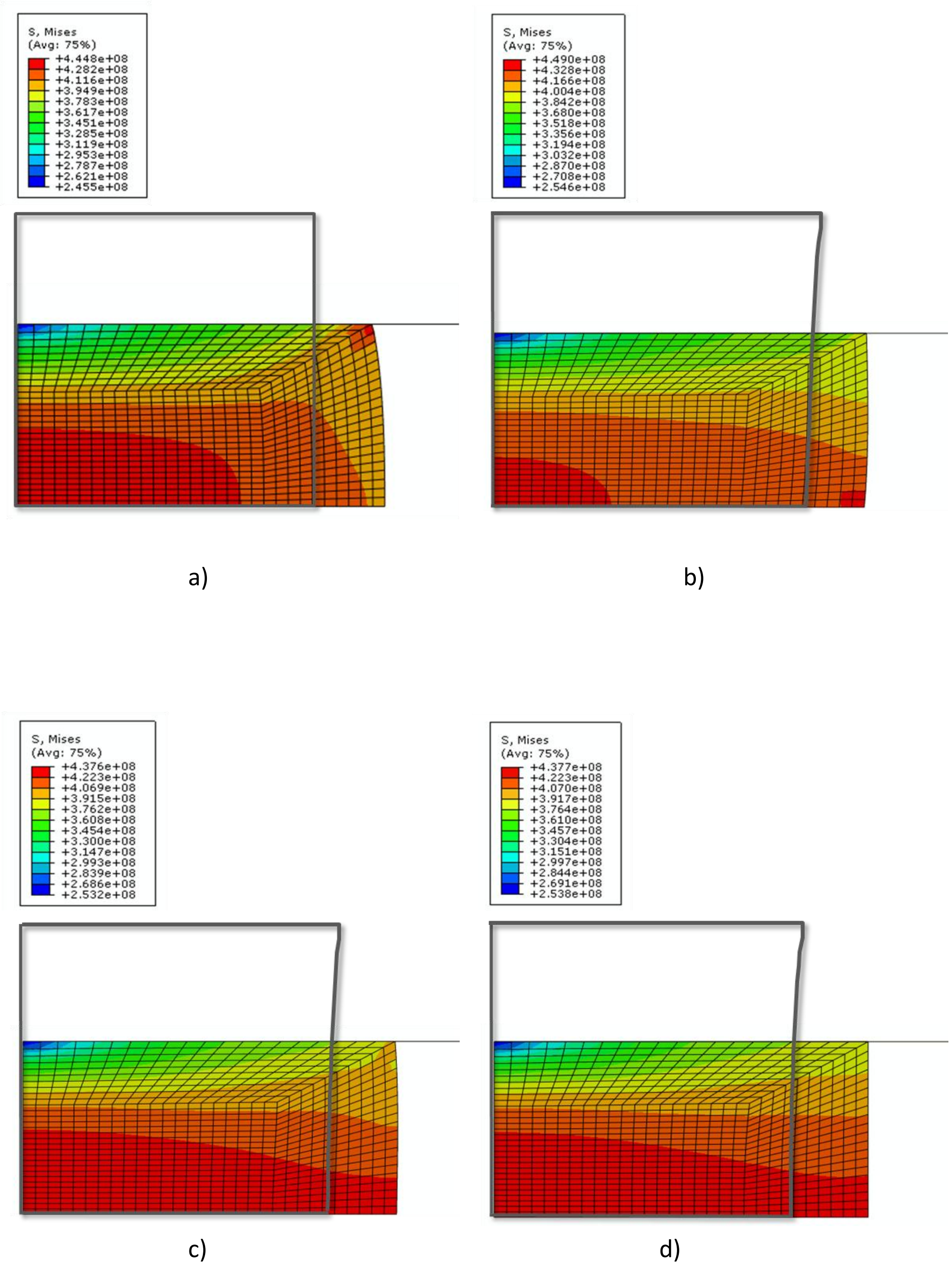
a)

b)

c)

d)

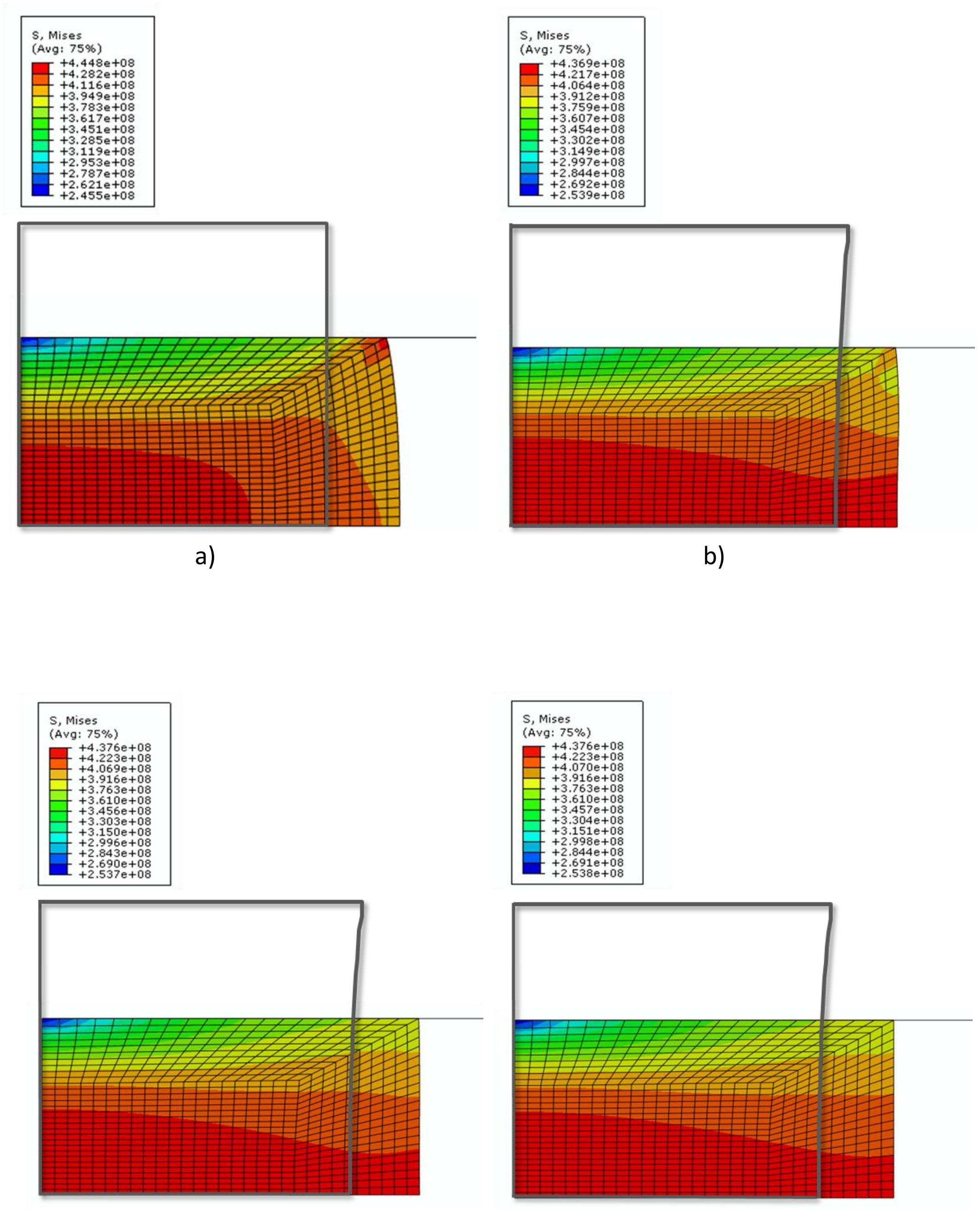
Figure 12.3 - Initial and final shape for the *set of points method*: a) iteration 0, b) iteration 22, c) iteration 28 and iteration 33 (optimum iteration).



a) b)

c) d)

Figure 12.4 - Initial and final shape for the cubic Bézier method: a) iteration 0, b) iteration 7, c) iteration 14 and iteration 20 (optimum iteration).



a) b)

c) d)

Figure 12.5 - Initial and final shape for the cubic NURBS method: a) iteration 0, b) iteration 7, c) iteration 14 and iteration 20 (optimum iteration).

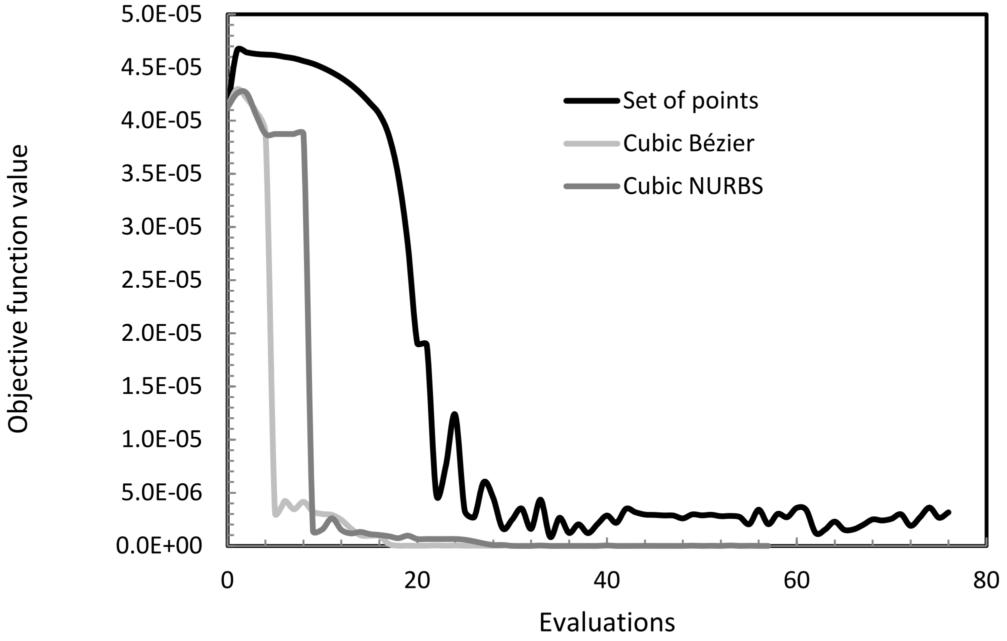


Figure 12.6 - Evolution of the objective function value in function of the evaluations number.

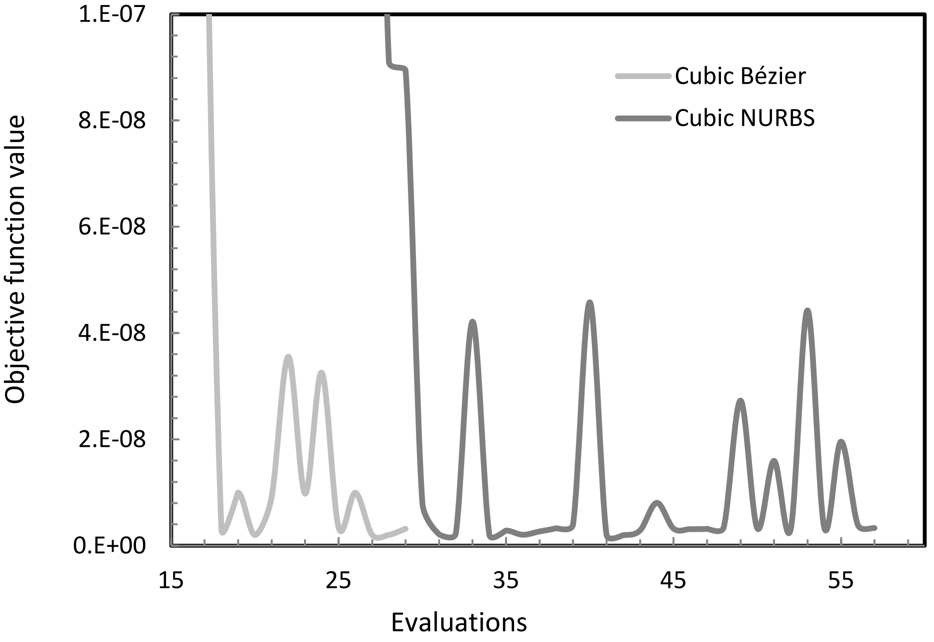


Figure 12.7 - Zoom of the evolution of the objective function value in function of the evaluations number.

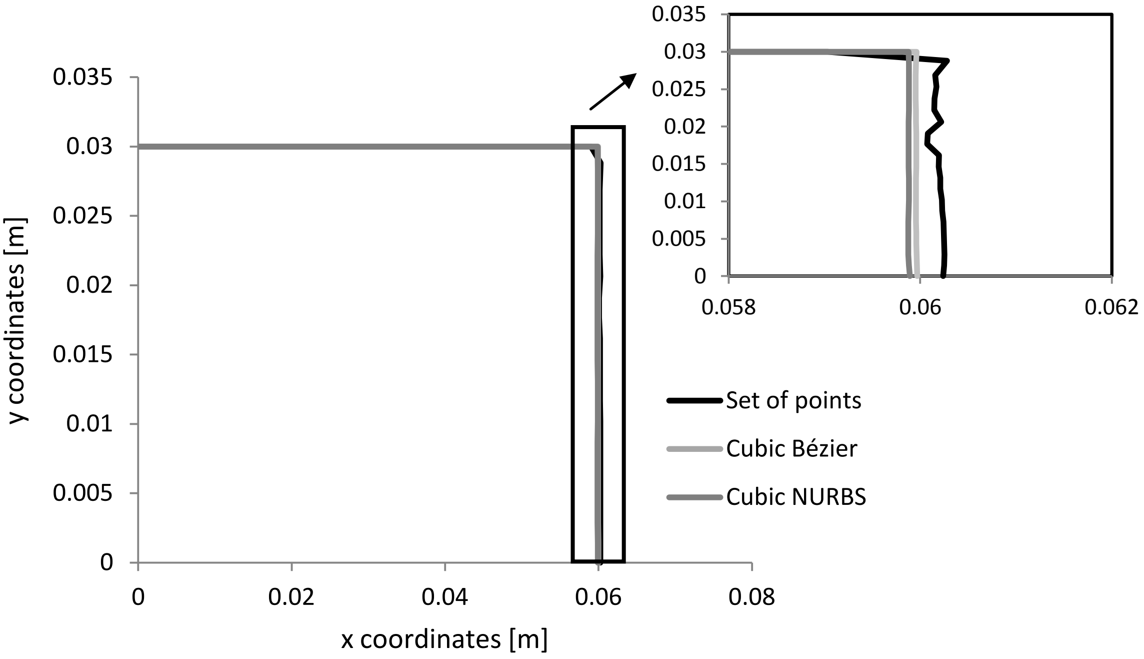


Figure 12.8 - Final geometries of the cylindrical billet, after forging, for the three methods.

In Table 12.2 and Table 12.3 the main results are presented. In Table 12.3 the results are shown for the three methods, considering for the *set of points* methodology 21 optimization variables that are correspondent to the 21 x node coordinates. In the case of the *cubic Bézier* methodology, the four optimization variables are the four control vertices, and in the *cubic NURBS* methodology the eight optimization variables are the four control vertices and the four correspondent weights.

Table 12.2 - Results for the three methods considered in the optimization of the initial cylindrical billet.

Parameterization method	Optimum objective function value	Number of iterations until the optimum	CPU time until the optimum[1] [s]
Set of points	8.2729x10^{-7}	35	525
Cubic Bézier	2.0x10^{-9}	21	315
Cubic NURBS	1.95x10^{-9}	43	627

[1] CPU time expended in an Intel® Core™ 2 Quad CPU Q 9400 at 2.66GHz with 3.25GB of RAM.

Table 12.3 - Optimum sets for the three methods considered in the optimization of the initial cylindrical billet.

	Set of points [m]	*Cubic Bézier* [m]	*Cubic NURBS*
X_1	0.49678 x10^{-1}	0.49440 x10^{-1}	0.49376 x10^{-1}m
X_2	0.49688 x10^{-1}	0.49425 x10^{-1}	0.49331 x10^{-1}m
X_3	0.49701 x10^{-1}	0.49780 x10^{-1}	0.50151 x10^{-1}m
X_4	0.49710 x10^{-1}	0.51390 x10^{-1}	0.51321 x10^{-1}m
X_5	0.49731 x10^{-1}		0.99889
X_6	0.49758 x10^{-1}		1.00086
X_7	0.49783 x10^{-1}		1.002070
X_8	0.49828 x10^{-1}		1.001657
X_9	0.49871 x10^{-1}		
X_{10}	0.49935 x10^{-1}		
X_{11}	0.50001 x10^{-1}		
X_{12}	0.50094 x10^{-1}		
X_{13}	0.50087 x10^{-1}		
X_{14}	0.50220 x10^{-1}		
X_{15}	0.50473 x10^{-1}		
X_{16}	0.50542 x10^{-1}		
X_{17}	0.50732 x10^{-1}		
X_{18}	0.50888 x10^{-1}		
X_{19}	0.51132 x10^{-1}		
X_{20}	0.51362 x10^{-1}		
X_{21}	0.50725 x10^{-1}		

12.2.4 Discussion and conclusions

Considering the presented results, it is possible to observe that the minimum value in terms of objective function was reached for the *cubic NURBS method* and is 1.95x10^{-9}. However, this is the most expensive method when it is considered the computational time dispended to reach the minimum.

Comparing the *cubic Bézier method* with the *cubic NURBS method* it is possible to verify that both methods reach close values of objective function and the *cubic Bézier method* needs only approximately half of the time than the *cubic NURBS method* to reach the minimum. This fact may occur because the *cubic NURBS method* has twice optimization variables, that allows a more flexible curve at the cost of much more time for finding the optimum. In geometric terms, considering the results presented in Figure 12.4 and Figure 12.5, it is possible to observe that for these two approaches the final shape of the billet is a straight billet, being this the main objective of this procedure.

The results obtained with the *set of points method* reach also quite good results although not so good as the other methods. This fact is due to the simple parameterization method used. This fact can be observed in Figure 12.3, where it is possible to see that the initial free surface of the billet is not smooth leading to a non-straight surface after the forging.

In conclusion, for a specific problem, if a method that only focus in the better results of the objective function is needed, the *cubic NURBS method* should be selected. However, if the computer efficiency should be also accounted for, the *cubic Bézier method* should be chosen.

12.3 First stage tool optimisation in a two stage forging process

12.3.1 Mechanical problem

The main difference between this mechanical problem and the previous one is the method used to find the straight cylindrical billet after the forming process. In this problem the shape of the initial billet remains constant during all the simulations. However, to achieve a straight cylindrical billet after the forming process, a shape optimization process of the first stage tool was applied. The second tool is a flat tool that gives the final shape of the billet as it is shown in Figure 12.9.

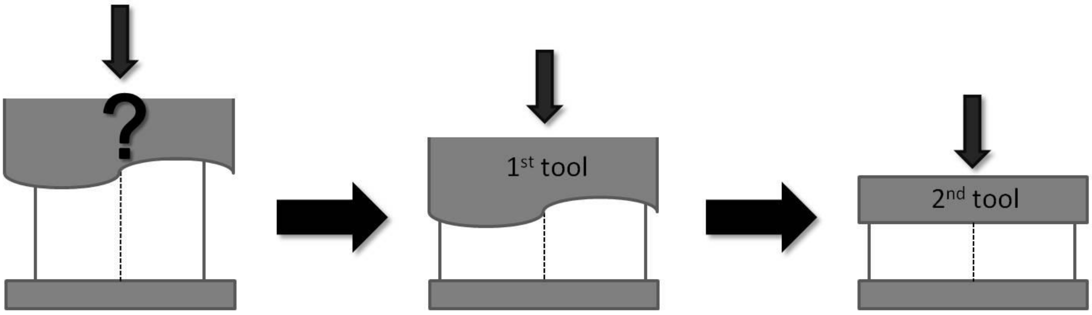


Figure 12.9 - Schema of the second approach.

The material properties for this problem remain the same as considered for the previous mechanical problem. Table 12.4 and Figure 12.10 present the geometry of this problem. The RP marked in Figure 12.10 is the reference point, *i.e.*, is a point which the tool is assigned and the displacement conditions of the tool are applied to this point.

Table 12.4 - Tool and billet dimensions.

Dimensions	
Tools radius, R [m]	0.075
Reference point position, h_1 [m]	0.030
Optimization zone, Δp [m]	0.015
Cylinder height, h [m]	0.050
Cylinder radius, r [m]	0.050

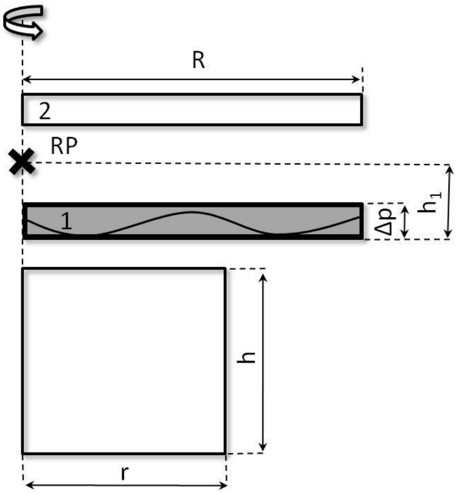


Figure 12.10 - Geometry problem.

12.3.2 Optimization procedure

In the present problem the main objective is to optimize the first tool shape. Two different parametric definitions of the curve were considered, such as a Bézier cubic curve and a NURBS cubic curve. Both curves were validated in the chapter 6. For both approaches the optimization procedure first starts with an initial set of parameters that formulate the parametric curve. This parametric formulation is implemented in Abaqus® by means of the RSURFU user subroutine. Then, the Abaqus® simulation of the two stage forging problem is performed. Subsquently, the results are analysed and the deviations between the desired and the numerical data are computed considering the previous objective function. After this calculation, the stopping criteria are analysed, ending the process if verified. However, if the stopping criteria aren´t verified a new set of parameters is considered in the parametric definition of the first tool and this procedure cyclically continues until the stopping criteria are achieved. This process is presented in the flowchart of Figure 12.11.

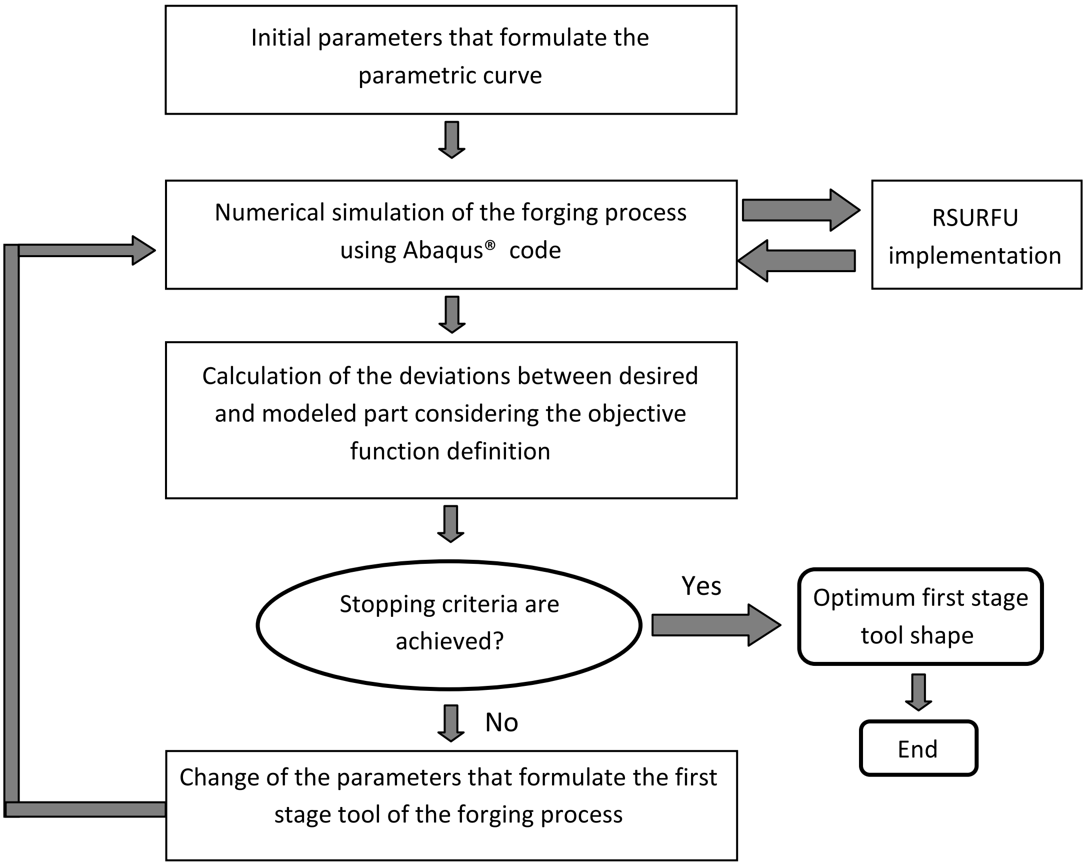

Initial parameters that formulate the parametric curve
Numerical simulation of the forging process using Abaqus® code
RSURFU implementation
Calculation of the deviations between desired and modeled part considering the objective function definition
Stopping criteria are achieved?
Yes
Optimum first stage tool shape
End
No
Change of the parameters that formulate the first stage tool of the forging process

Figure 12.11 - Flowchart of the presented forging optimization process.

12.3.3 Results for the Bézier subroutine

After the Bézier RSURFU validation, the optimization of the first stage tool was performed. Four different approaches were studied. The first one, called *4 optimization variables* have four optimization variables that are the x-coordinates of the four control points of the cubic Bézier curve. For this case the displacement of the first stage tool is constant and equal to -0.025m. The other three approaches use five optimization variables: four x-coordinates of the control points and the displacement of the first tool. The difference between these three approaches is the starting value of the displacement variable (d) in the optimization process. The values considered were -0.015m, -0.020m and -0.025m. This study considering three different values of initial displacement was considered in order to understand how the initial displacement of the first tool influences all the optimization process. In order to perform this optimization process a Levenberg-Marquardt gradient algorithm was used to minimize the objective function. The objective function is the same as the one defined in section 12.2.2 section. A maximum number of 200 iterations or a stagnation value of 1x10^{-60} are considered as stopping criteria.

In Figure 12.12 the evolution of the objective function for the four different approaches is represented and Figure 12.13 is a zoom from Figure 12.12 that allows a better view of the evolution of the objective function for the four approaches.

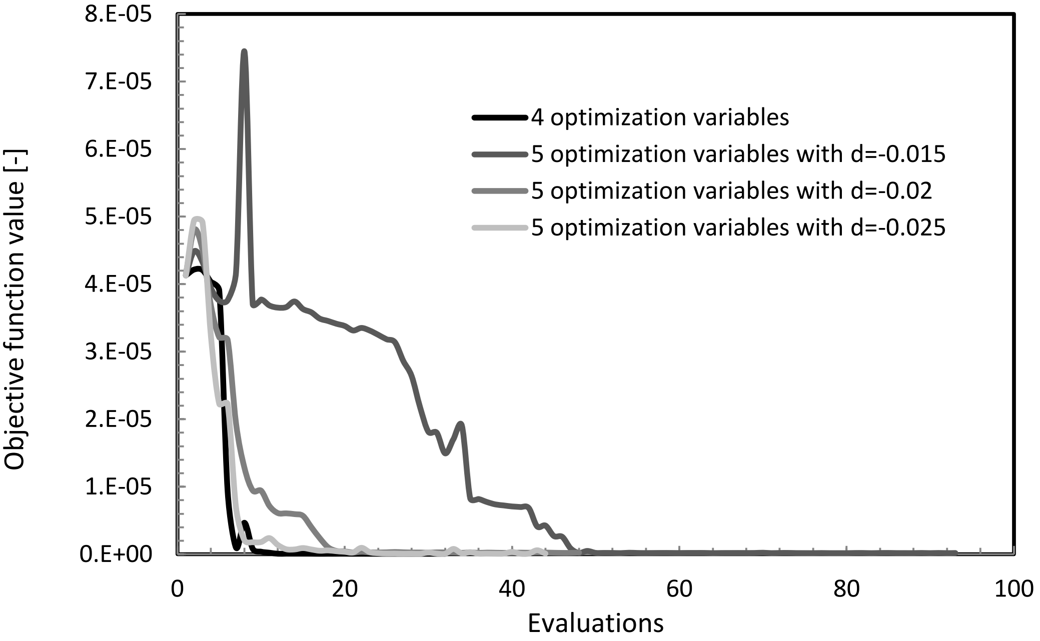


Figure 12.12 - Evolution of the objective function value in function of the evaluations number.

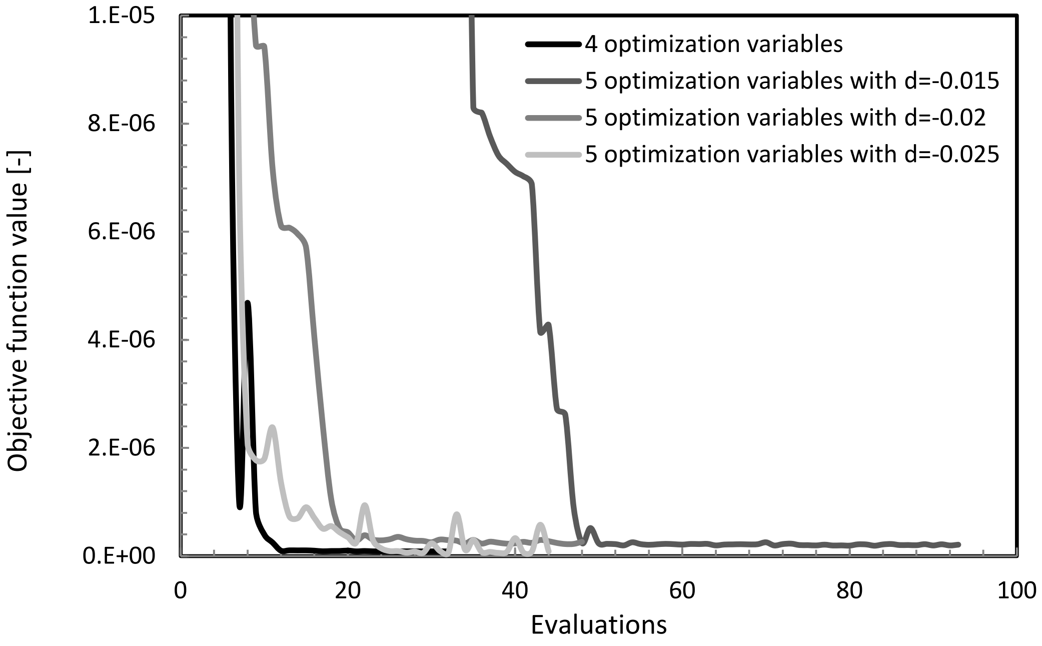


Figure 12.13 - Evolution of the objective function value in function of the evaluations number.

From Figure 12.14 to Figure 12.29 the results obtained for each method, in different iterations, during the optimization process are presented. The von Mises equivalent stress is also ploted.

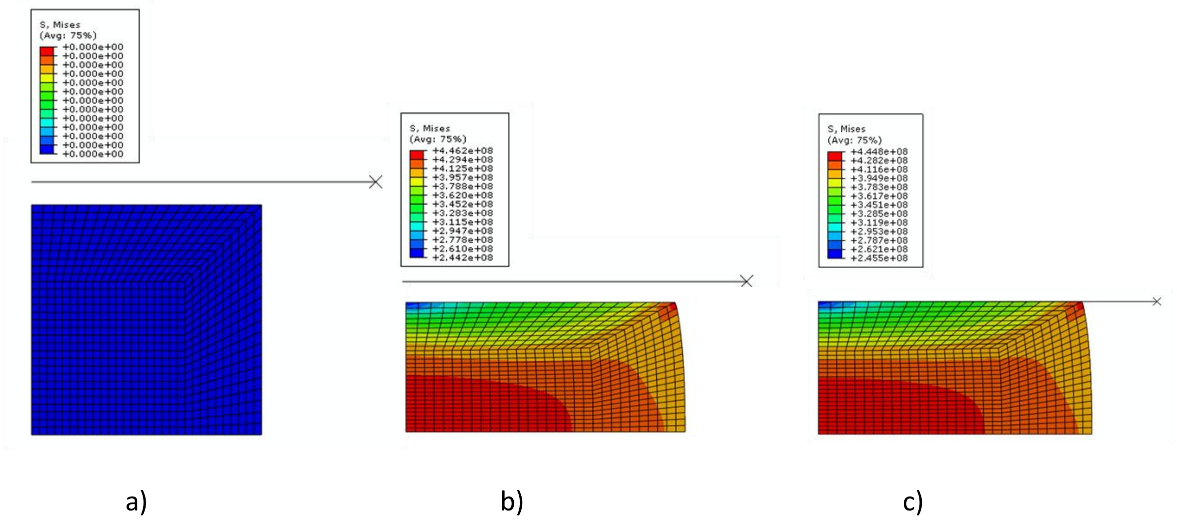
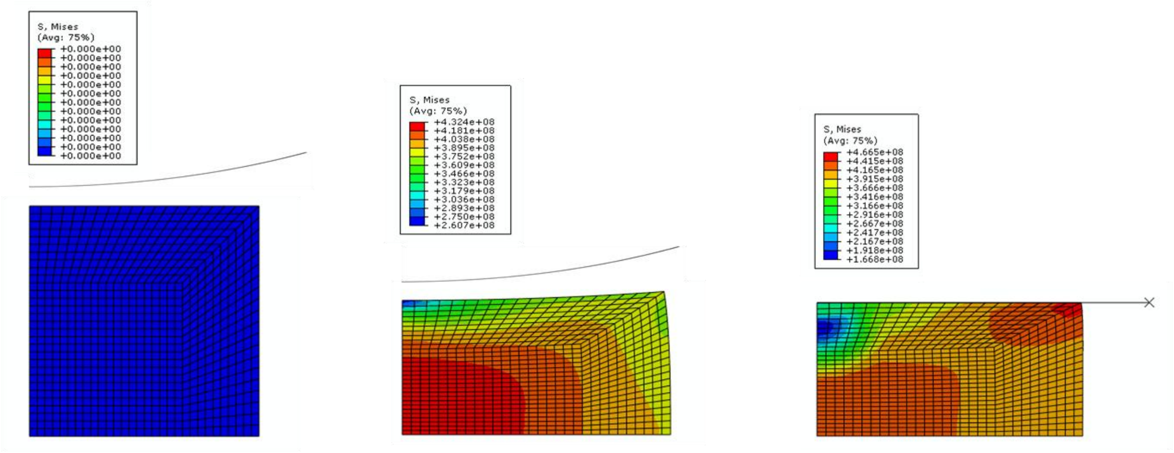
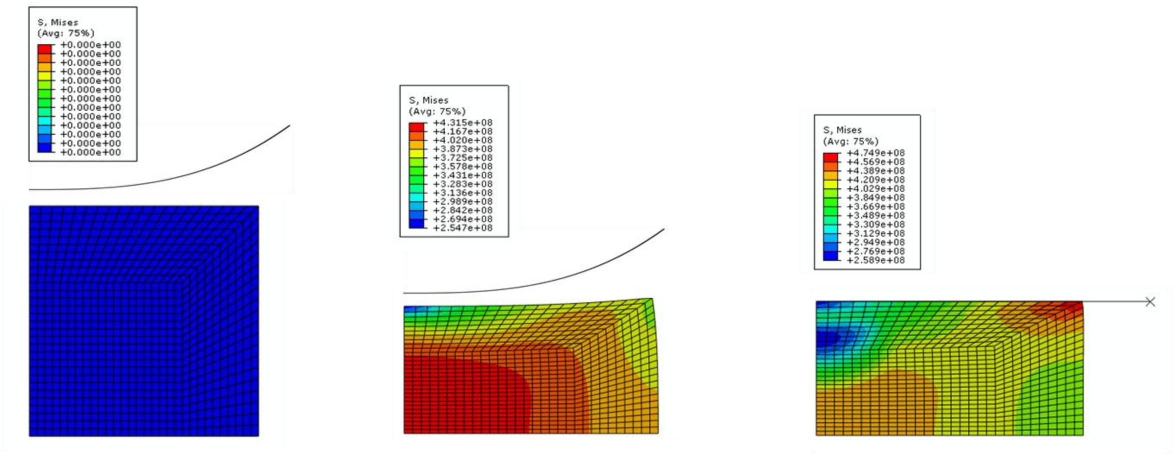


Figure 12.14 - Initial and final shapes of the billet and the tools for the 4 optimization variables in iteration 0.



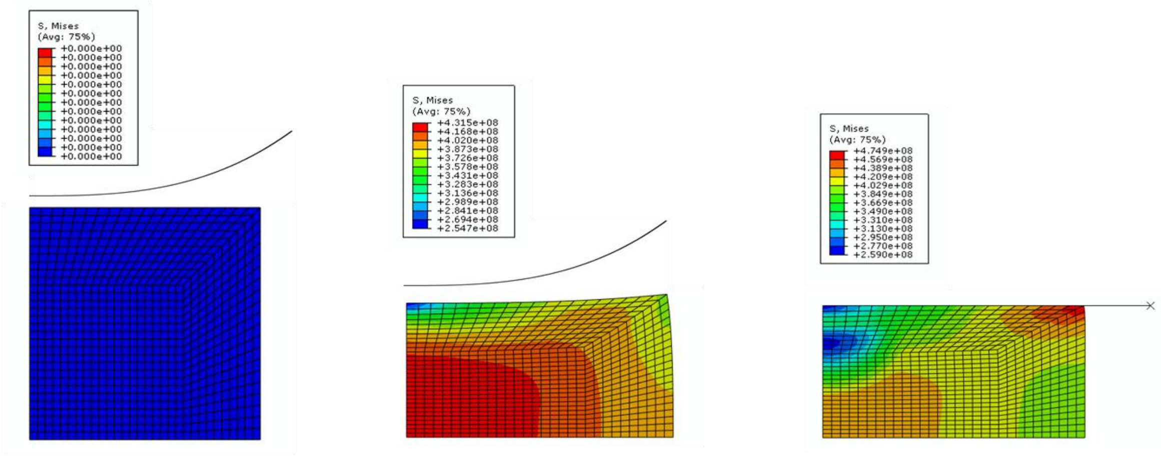
a) b) c)

Figure 12.15 - Initial and final shapes of the billet and the tools for the 4 optimization variables in iteration 10.



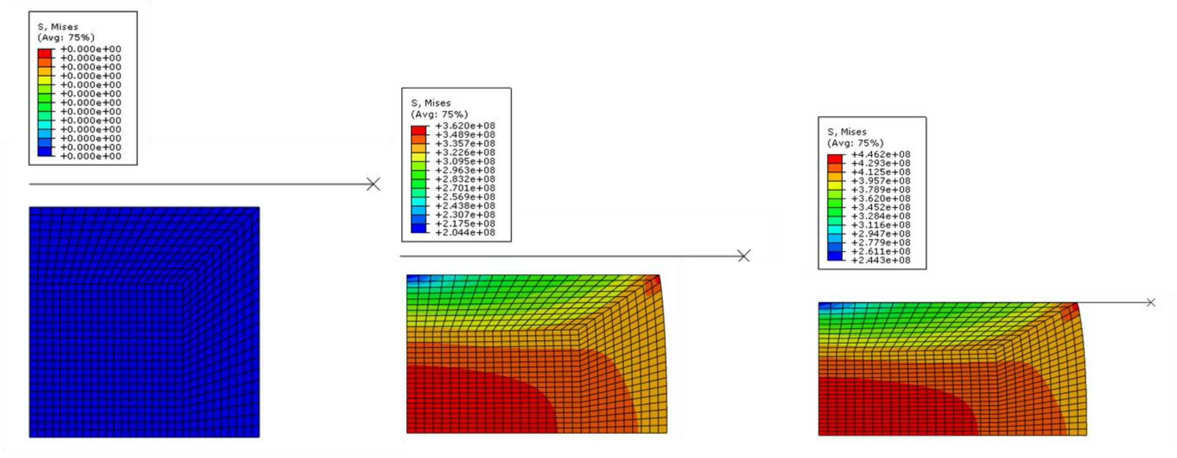
a) b) c)

Figure 12.16 - Initial and final shapes of the billet and the tools for the 4 optimization variables in iteration 18.



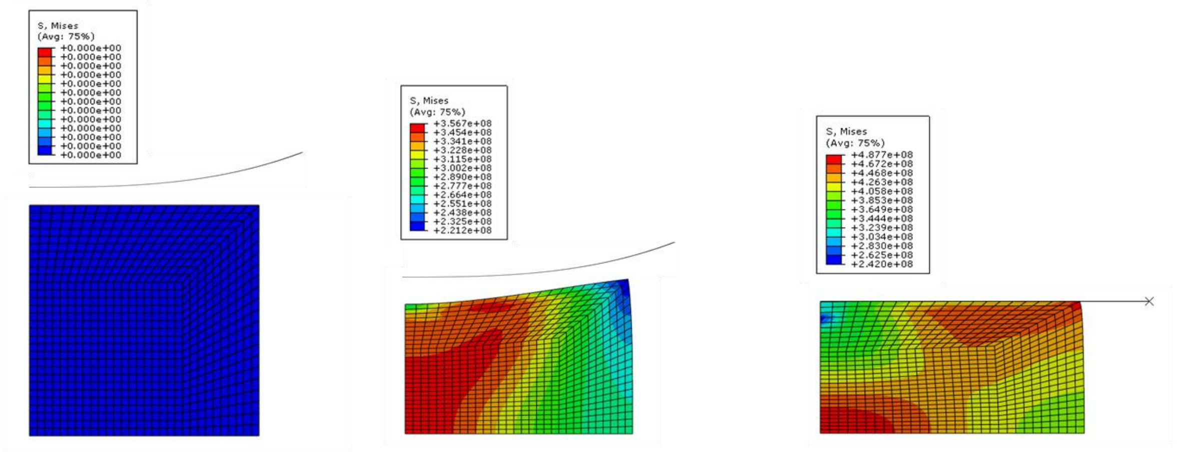
a) b) c)

Figure 12.17 - Initial and final shapes of the billet and the tools for the 4 optimization variables in iteration 26.



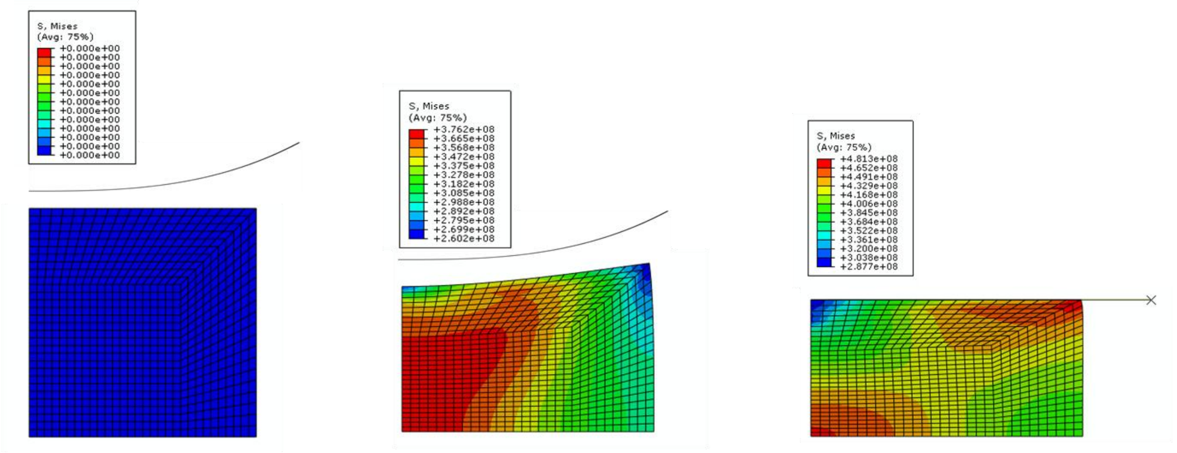
a) b) c)

Figure 12.18 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 0 with initial displacement of -0.015 m.



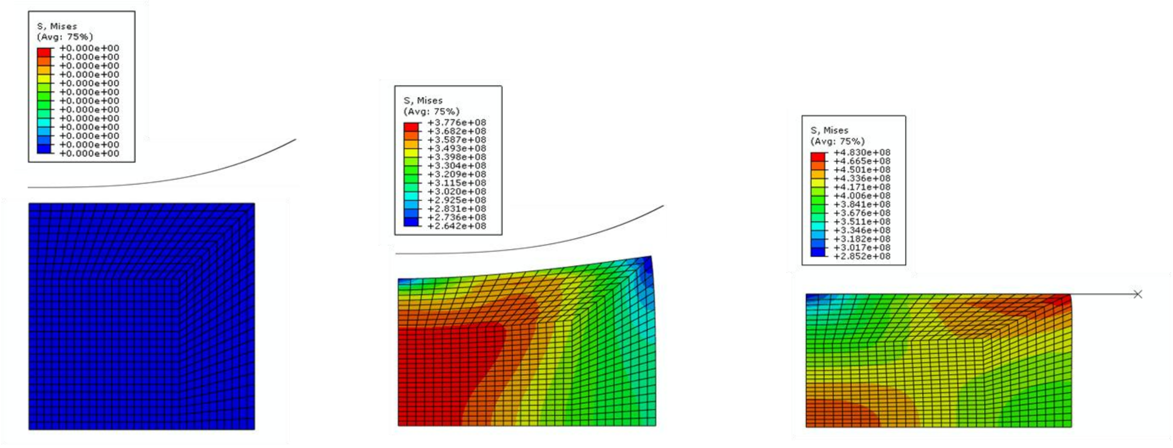
a) b) c)

Figure 12.19 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 25 with initial displacement of -0.015 m.



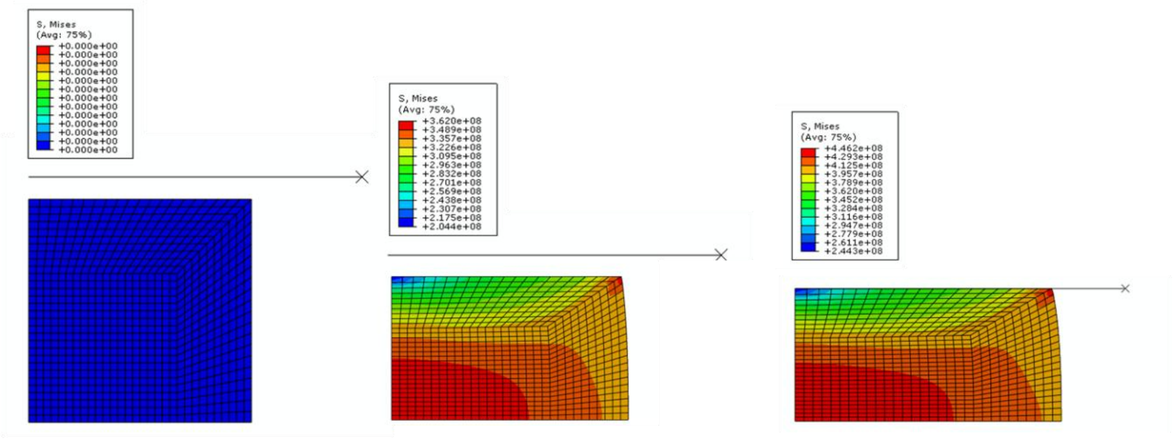
a) b) c)

Figure 12.20 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 55 with initial displacement of -0.015 m.



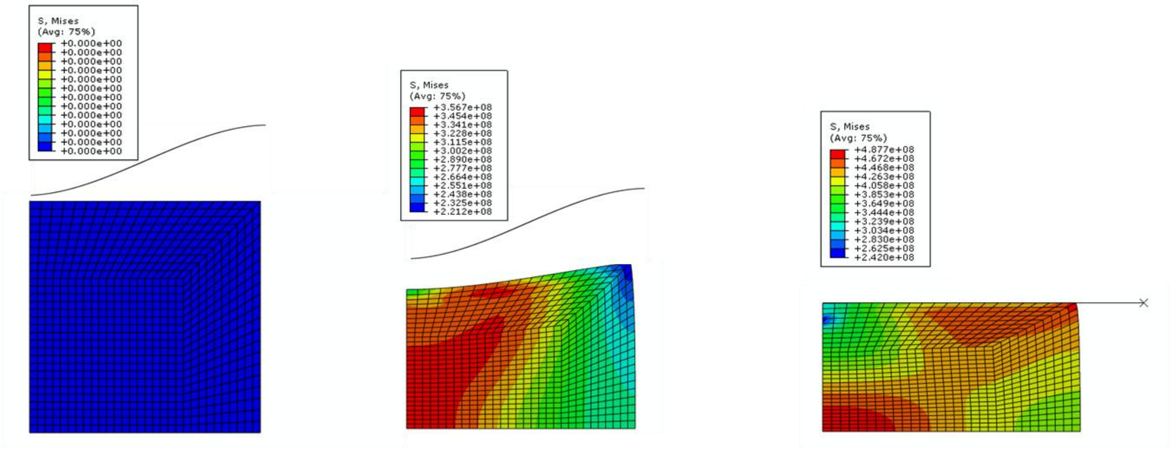
a) b) c)

Figure 12.21 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 83 with initial displacement of -0.015 m.



a) b) c)

Figure 12.22 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 0 with initial displacement of -0.02 m.



a) b) c)

Figure 12.23 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 15 with initial displacement of -0.02 m.

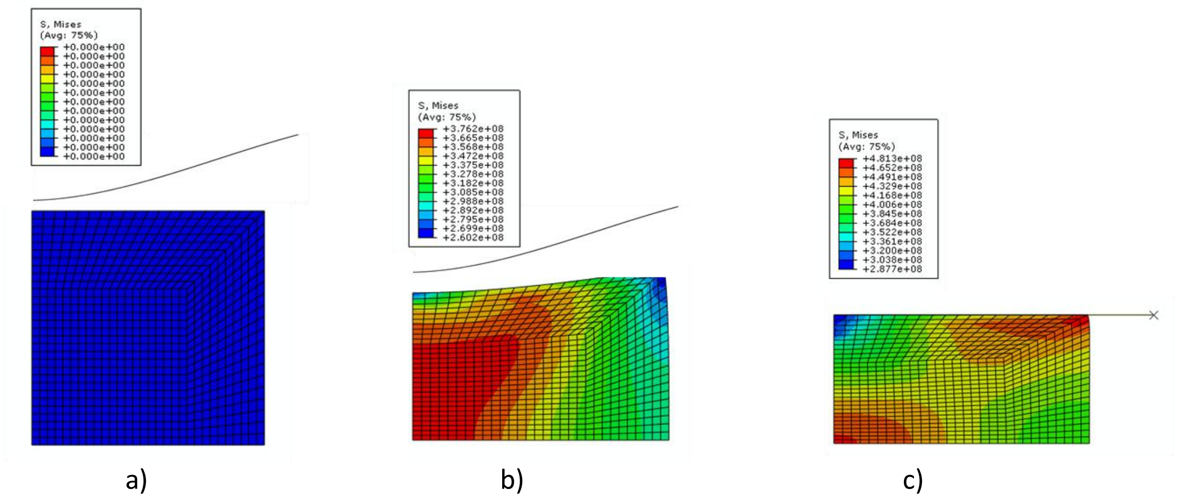


Figure 12.24 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 30 with initial displacement of -0.02 m.

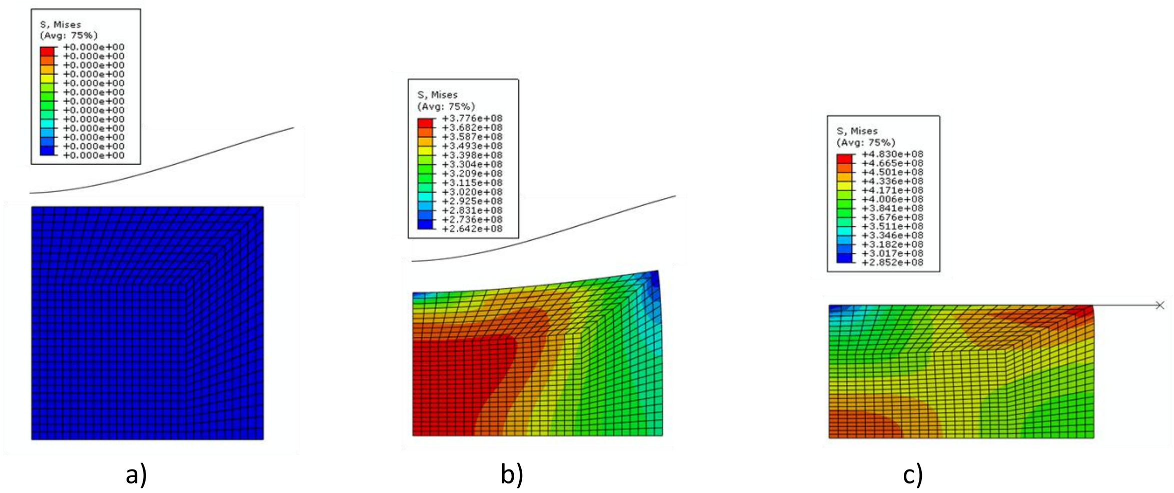


Figure 12.25 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 46 with initial displacement of -0.02 m.

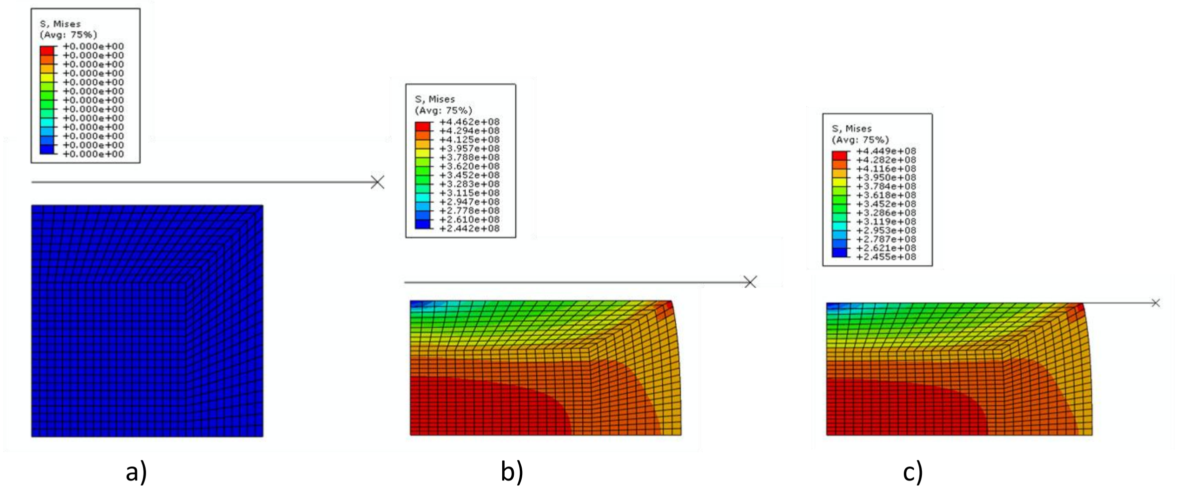


Figure 12.26 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 0 with initial displacement of -0.025 m.

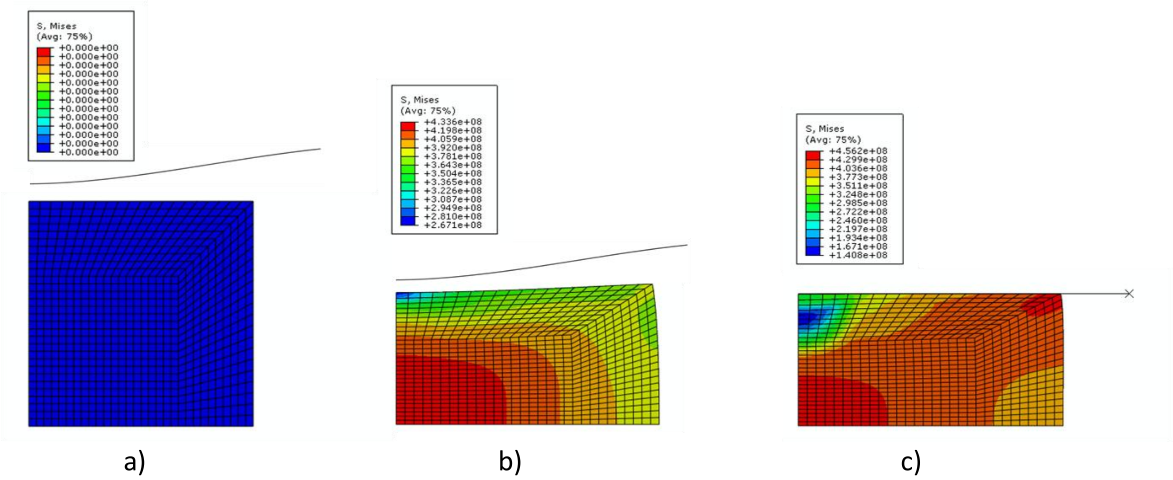


Figure 12.27 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 10 with initial displacement of -0.025 m.

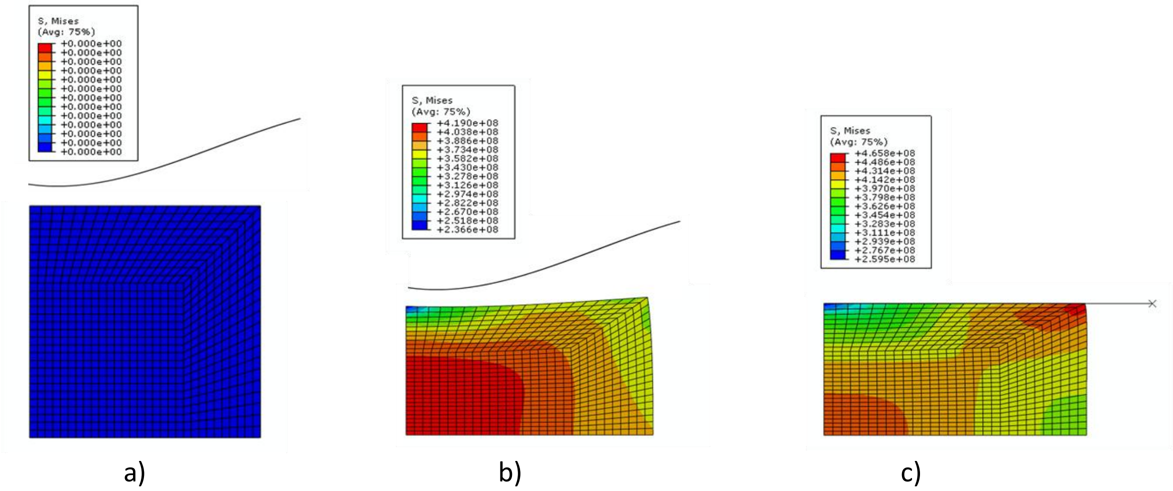


Figure 12.28 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 20 with initial displacement of -0.025 m.

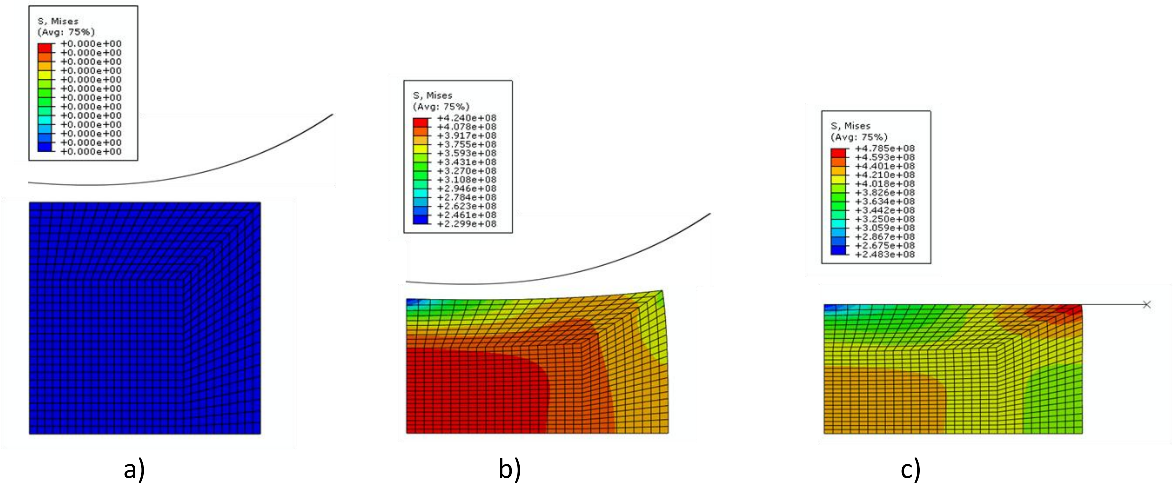


Figure 12.29 - Initial and final shapes of the billet and the tools for the 5 optimization variables in iteration 32 with initial displacement of -0.025 m.

In Figure 12.30 to Figure 12.33 the shape of the first tool is presented for some iterations during the optimization process.

In Figure 12.34 the optimum first tool shapes for the four approaches considered are represented.

In Table 12.5 and Table 12.6 the most important results obtained with the different four approaches for the optimization with the Bézier cubic curve implementation are systematized.

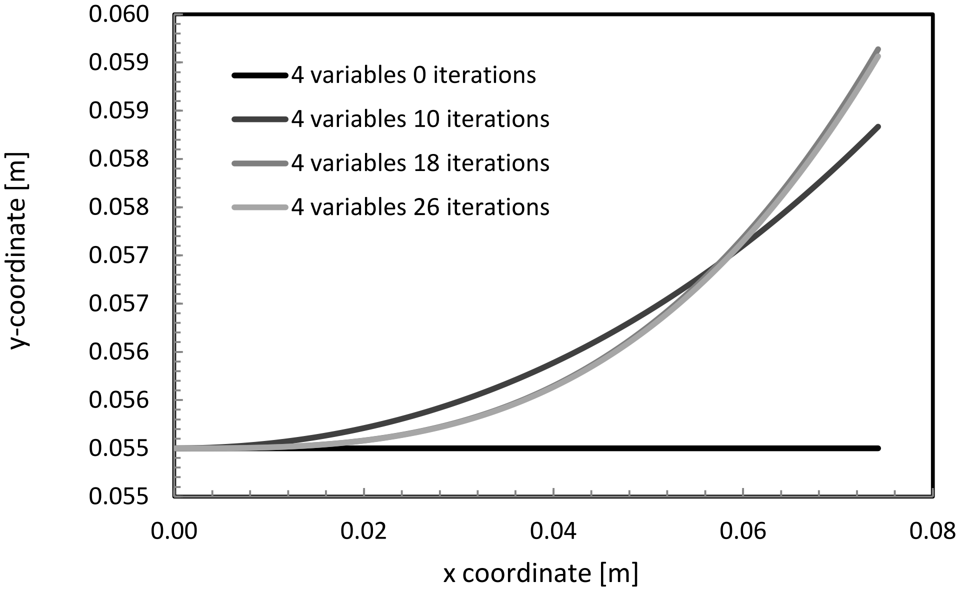


Figure 12.30 - Shape of the first tool for the 4 variables approach.

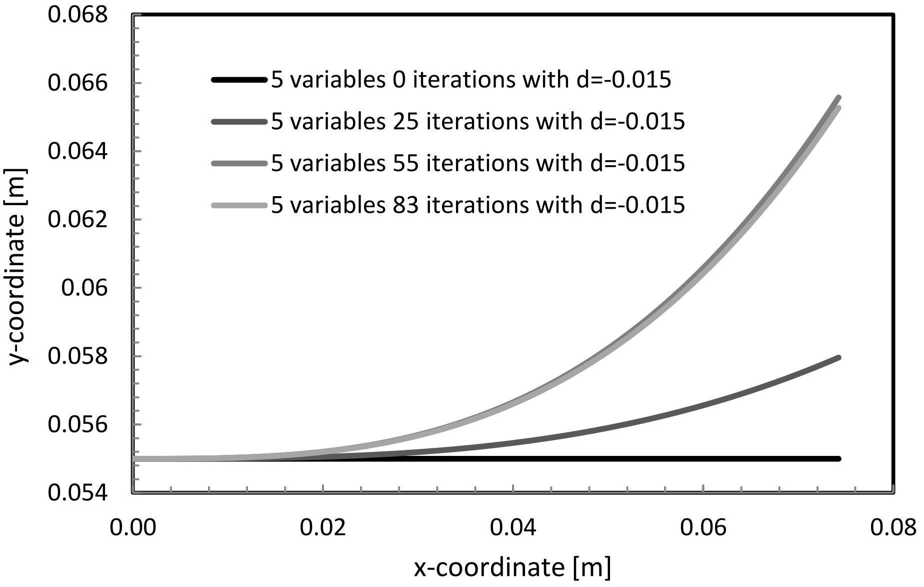


Figure 12.31 - Shape of the first tool for the 5 variables approach with starting displacement of -0.015 m.

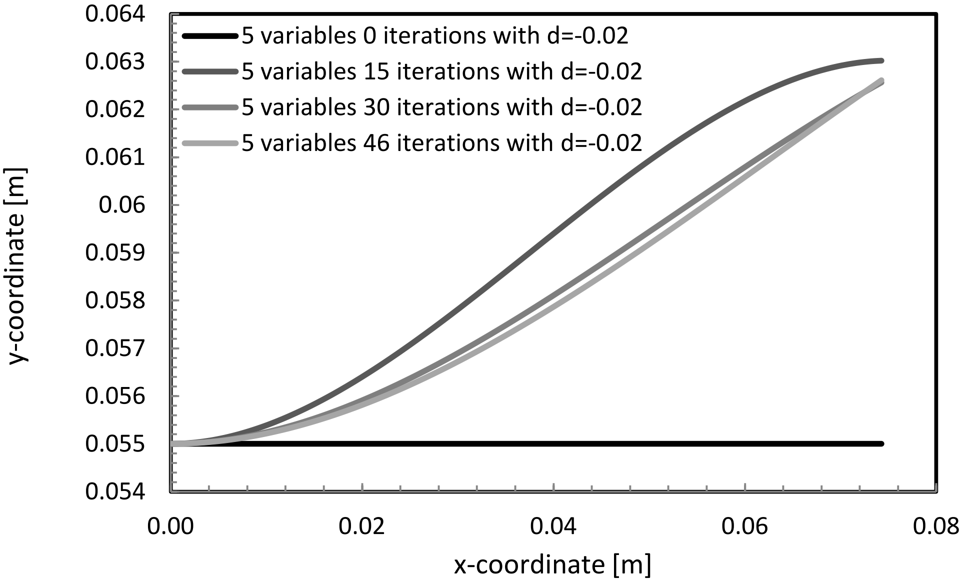


Figure 12.32 - Shape of the first tool for the 5 variables approach with starting displacement of -0.02 m.

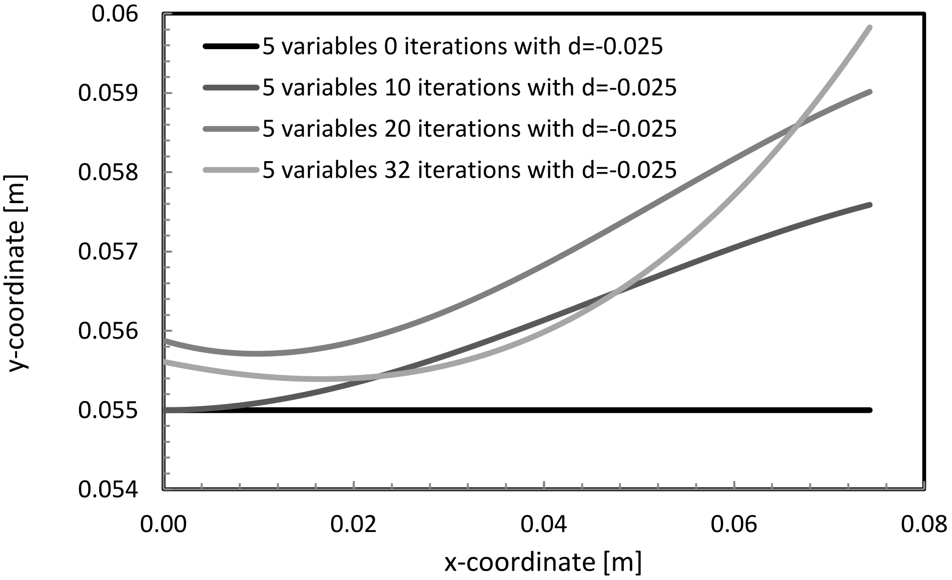


Figure 12.33 - Shape of the first tool for the 5 variables approach with starting displacement of -0.025 m.

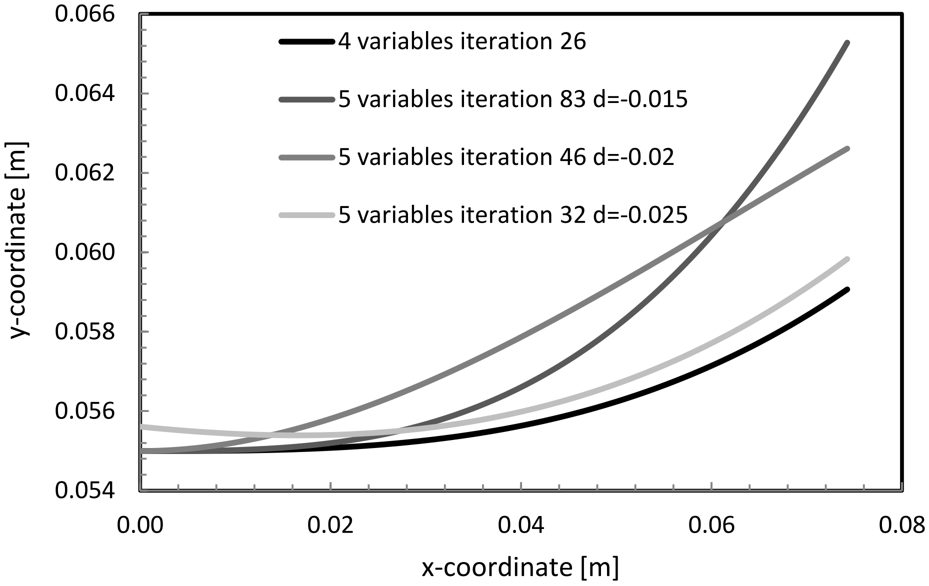


Figure 12.34 - Optimum shape of the first tool for the 4 approaches considered.

Table 12.5 - Results for the Bézier RSURFU implementation.

Parameterization method	Optimum objective function value [-]	Number of iterations until the optimum	CPU time until the optimum[2] [s]
4 variables	$8.8827\text{x}10^{-8}$	26	1294
5 var. 0.015	$1.8950\text{x}10^{-7}$	83	4133
5 var. 0.02	$2.2235\text{x}10^{-7}$	46	2290
5 var. 0.025	$6.1434\ \text{x}10^{-8}$	32	1594

Table 12.6 - Best parameters sets for the Bézier RSURFU implementation.

Parameterization method	x_1 [m]	x_2 [m]	x_3 [m]	x_4 [m]	x_5 [m]
4 variables	0.055	0.055	0.055	0.05918774	-
5 var. 0.015	0.055	0.055	0.055	0.06559008	-0.02160044
5 var. 0.02	0.055	0.055	0.06299601	0.06302654	-0.02105060
5 var. 0.025	0.05560831	0.055	0.05542591	0.05996363	-0.025

[2] CPU time expended in an Intel® Core™ 2 Quad CPU Q 9400 at 2.66GHz with 3.25GB of RAM

12.3.4 Results for the NURBS subroutine

After the NURBS RSURFU validation (see chapter 6) the optimization of the first tool shape was performed. The shape of the first tool was implemented considering the NURBS RSURFU (see Table A2 in Appendix).

Four different approaches were considered, being the first one, called *8 optimization variables*. This approach have 8 optimization variables that are the 4 x-coordinates of the control points and the 4 correspondent weights, being the descendent displacement of the first stage tool constant and equal to 0.025 m. The other three approaches have 9 optimization variables, the 8 considered in the last method, and in addition, the displacement of the first stage tool. The difference between these three approaches is the starting value of the displacement in the optimization process. The starting displacement (d) considered were -0.015 m, -0.020 m and -0.025 m. In order to perform this optimization process a Levenberg-Marquardt gradient algorithm was used to minimize the objective function. The objective function considered was defined in section 12.2.2. As convergence conditions, it was considered a maximum number of 200 iterations or a stagnation value of 1x10^{-60}. In Figure 12.35 the objective function value in function of the evaluations number for the four different approaches is represented. Figure 12.36 shows a zoom of Figure 12.35 that allows to see how the objective functions behave near the zero.

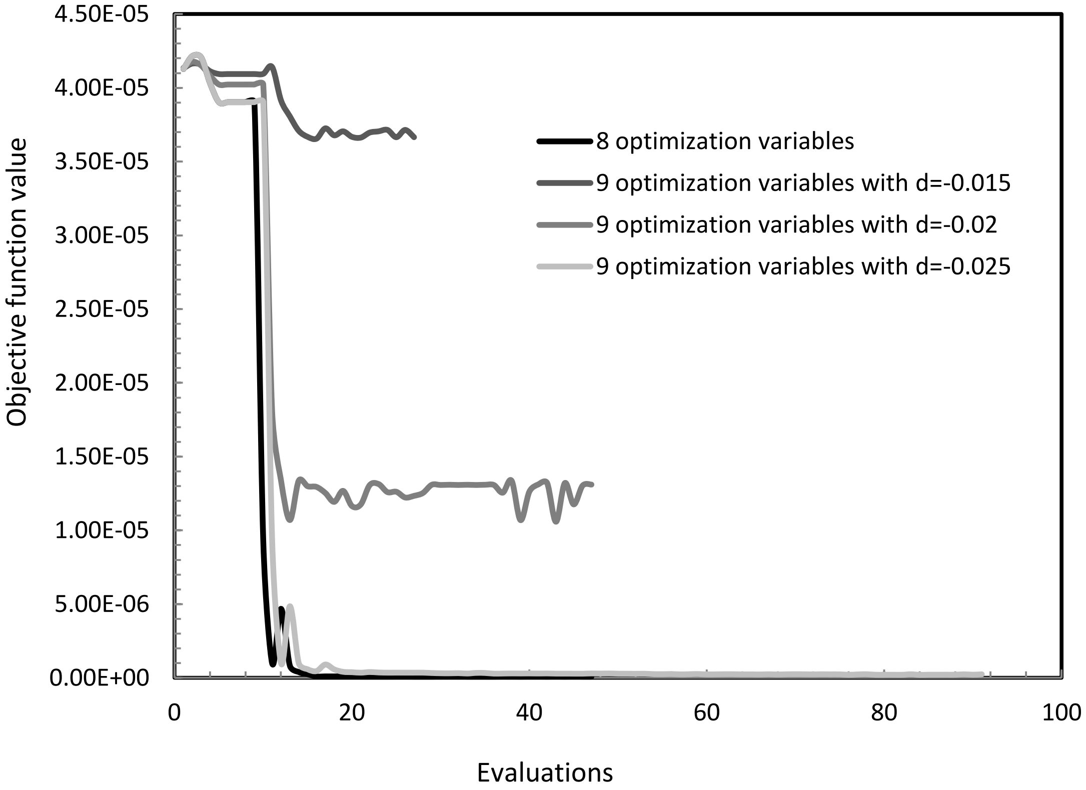


Figure 12.35 - Objective function value in function of the evaluations number.

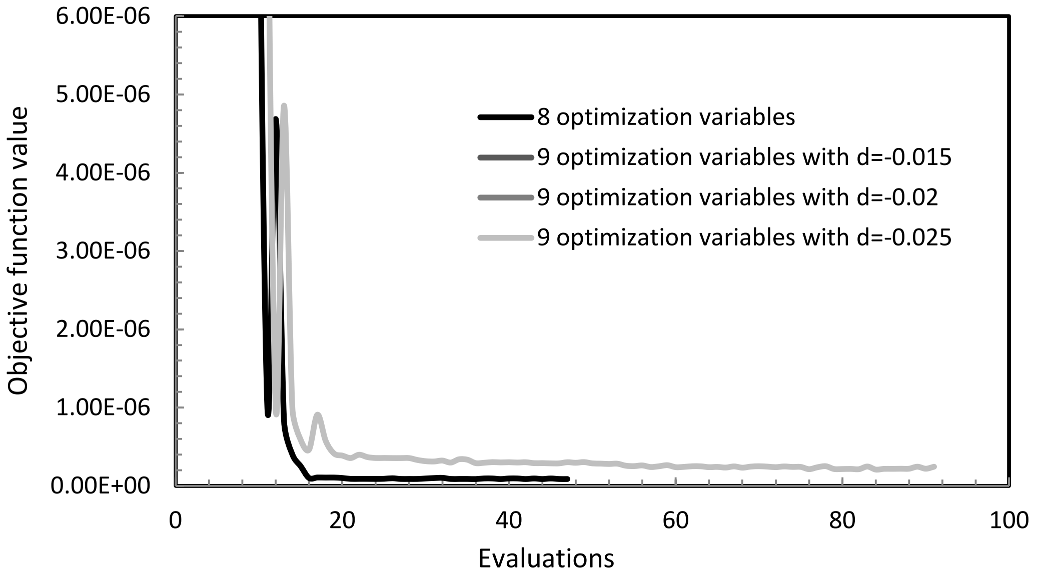
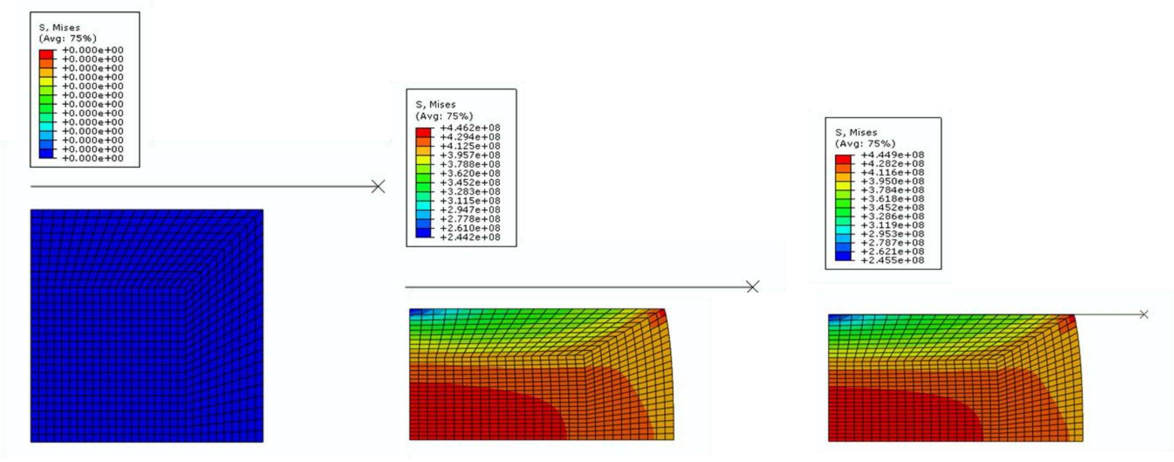


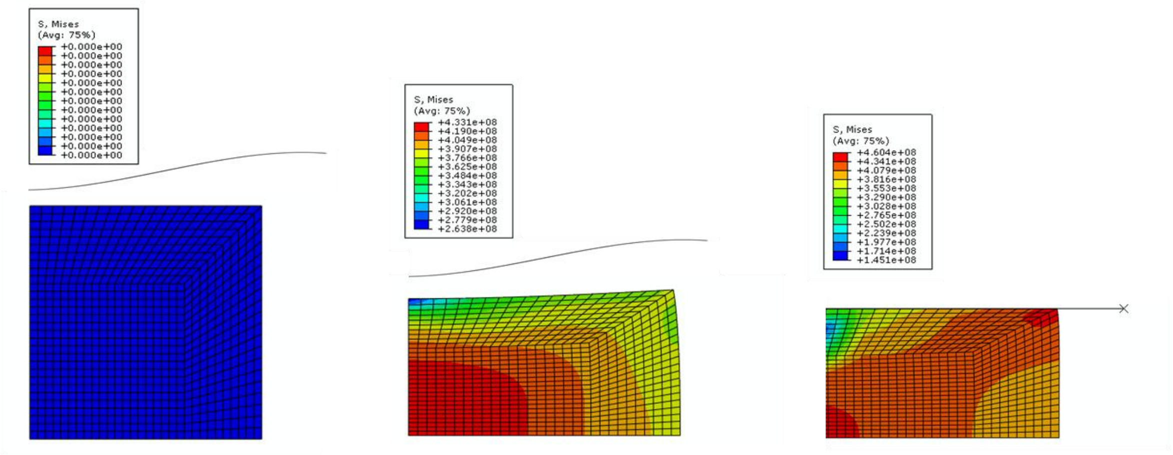
Figure 12.36 - Objective function value in function of the evaluations number.

The results obtained for each method, in different iterations during the optimization process, can be seen in figures 12.37-52. The von Mises equivalent stress can also be observed in these figures.



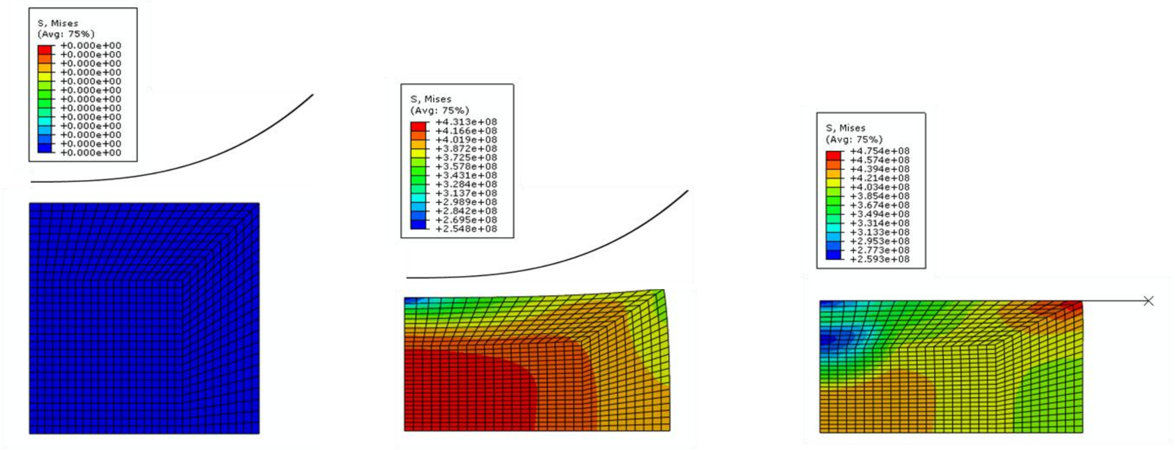
a) b) c)

Figure 12.37 - Initial and final shapes of the billet and the tools for the 8 optimization variables in iteration 0.



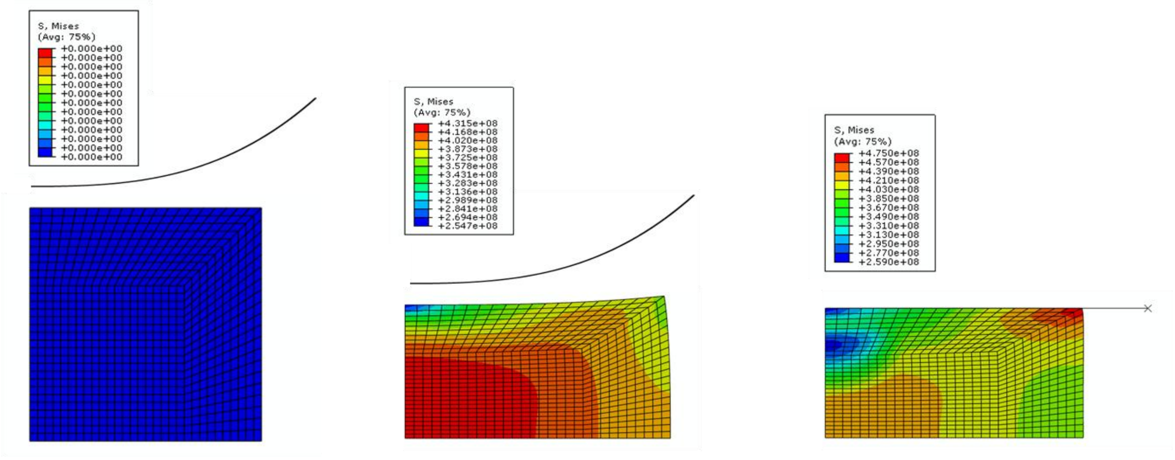
a) b) c)

Figure 12.38 - Initial and final shapes of the billet and the tools for the 8 optimization variables in iteration 11.



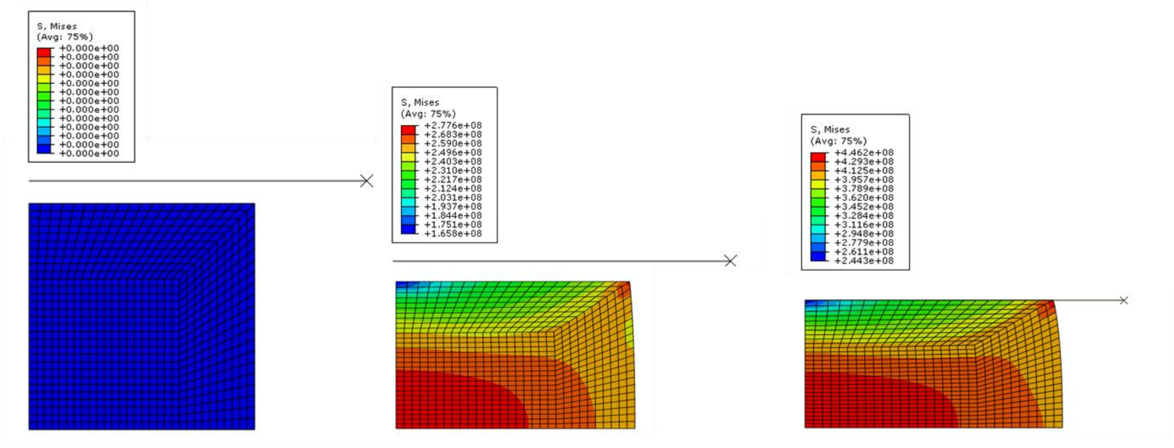
a) b) c)

Figure 12.39 - Initial and final shapes of the billet and the tools for the 8 optimization variables in iteration 20.



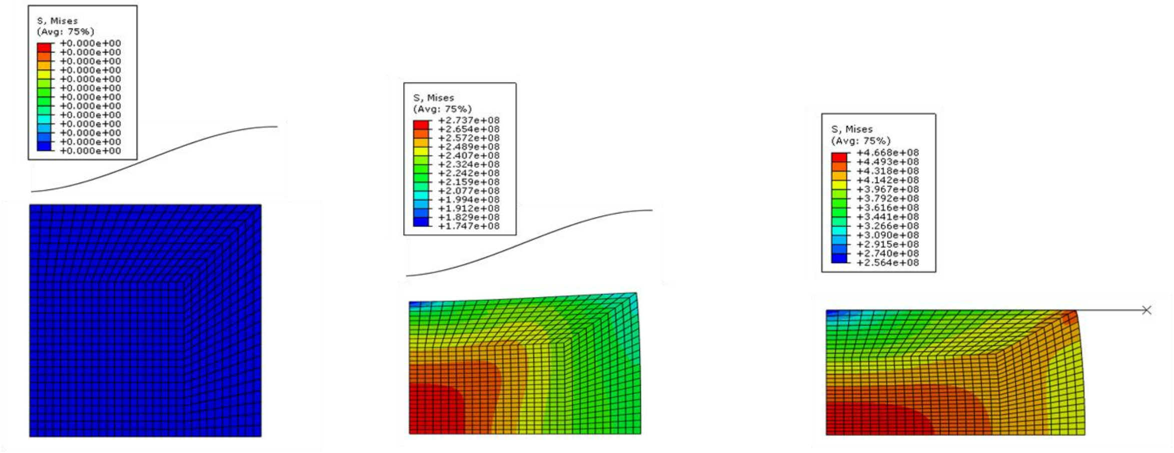
a) b) c)

Figure 12.40 - Initial and final shapes of the billet and the tools for the 8 optimization variables in iteration 36.



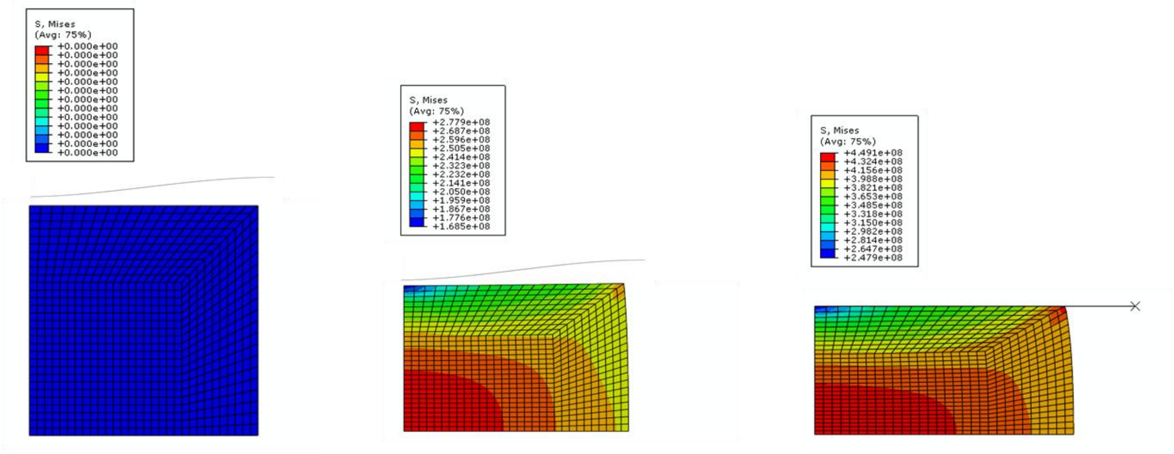
a) b) c)

Figure 12.41 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 0 with initial displacement -0.015 m.



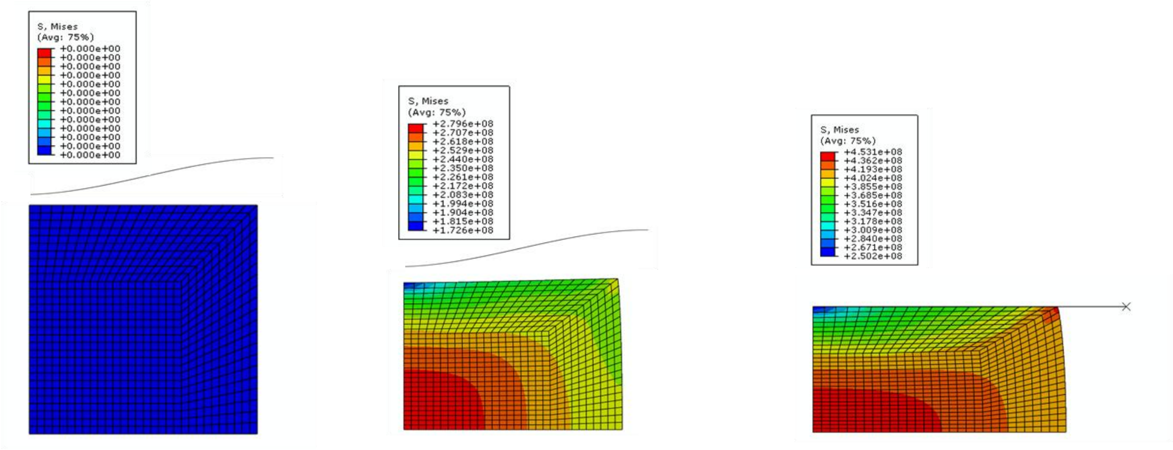
a) b) c)

Figure 12.42 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 11 with initial displacement of -0.015 m.



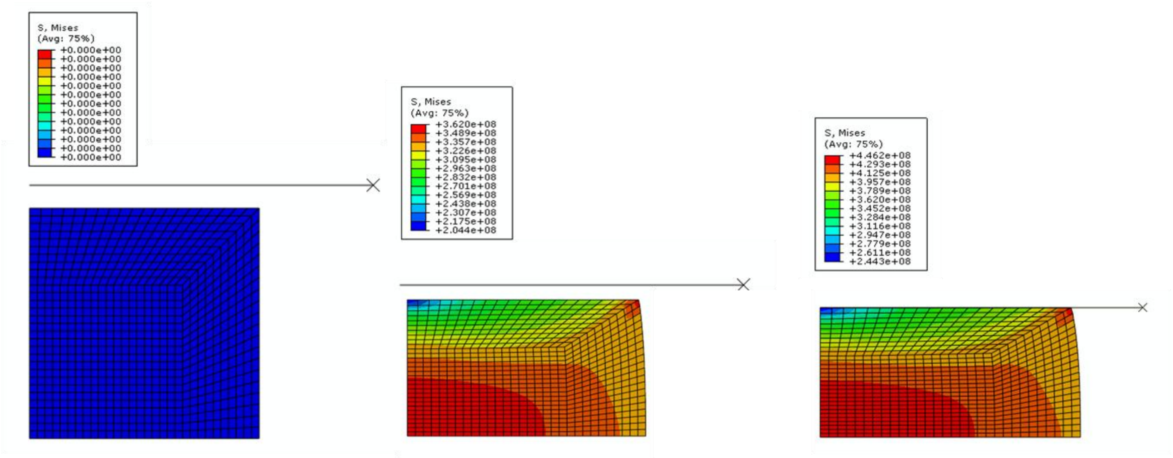
a) b) c)

Figure 12.43 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 14 with initial displacement of -0.015 m.



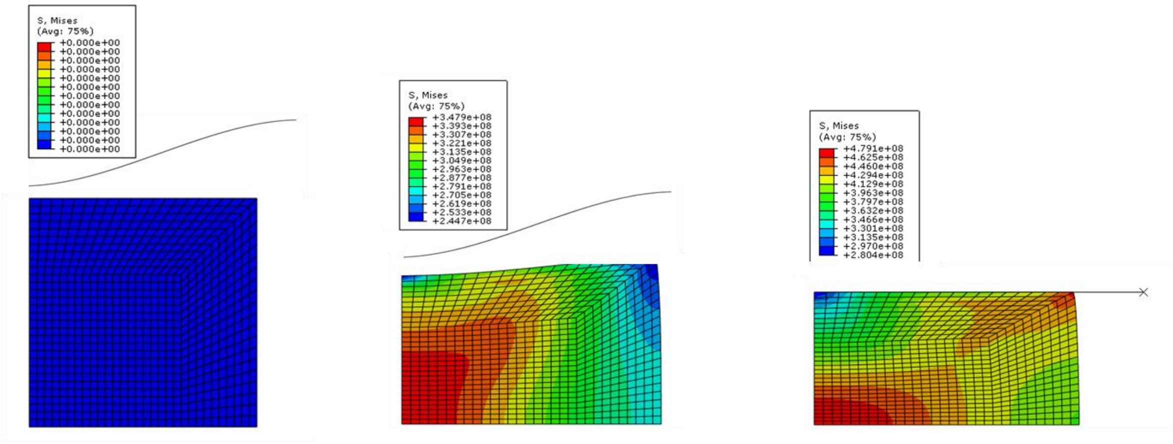
a) b) c)

Figure 12.44 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 16 with initial displacement of -0.015 m.



a) b) c)

Figure 12.45 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 0 with initial displacement of -0.02 m.



a) b) c)

Figure 12.46 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 15 with initial displacement of -0.02 m.

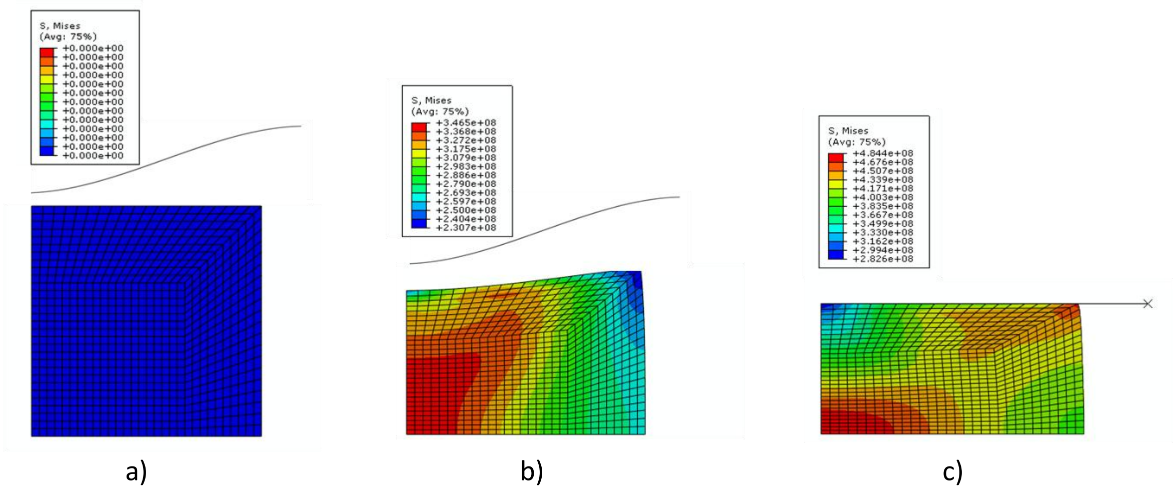


Figure 12.47 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 30 with initial displacement of -0.02 m.

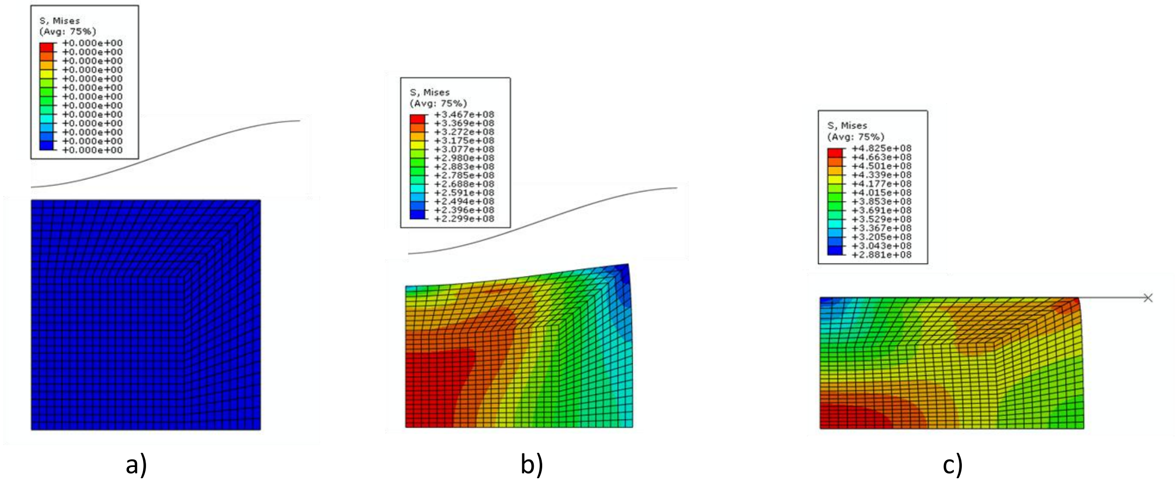


Figure 12.48 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 43 with initial displacement of -0.02 m.

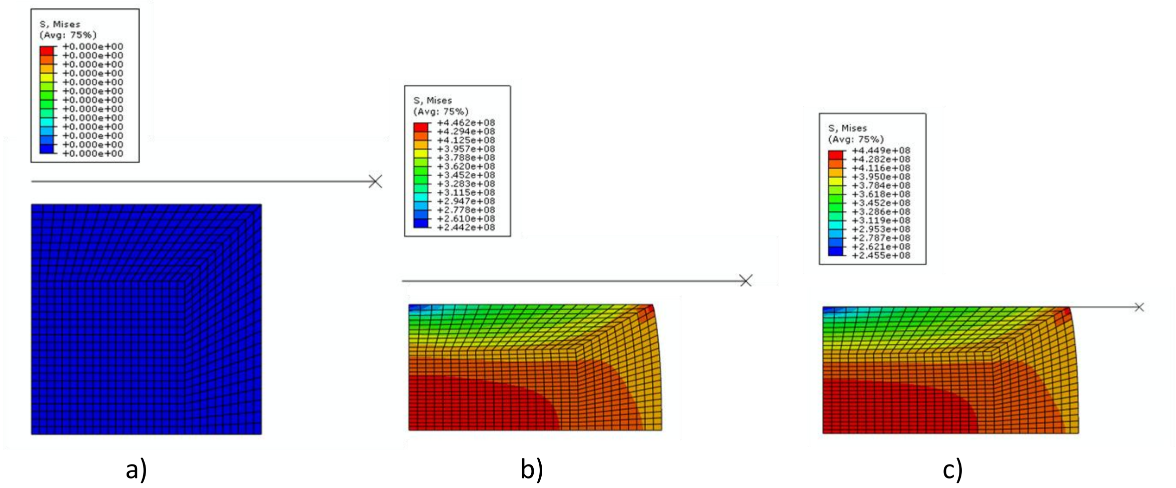
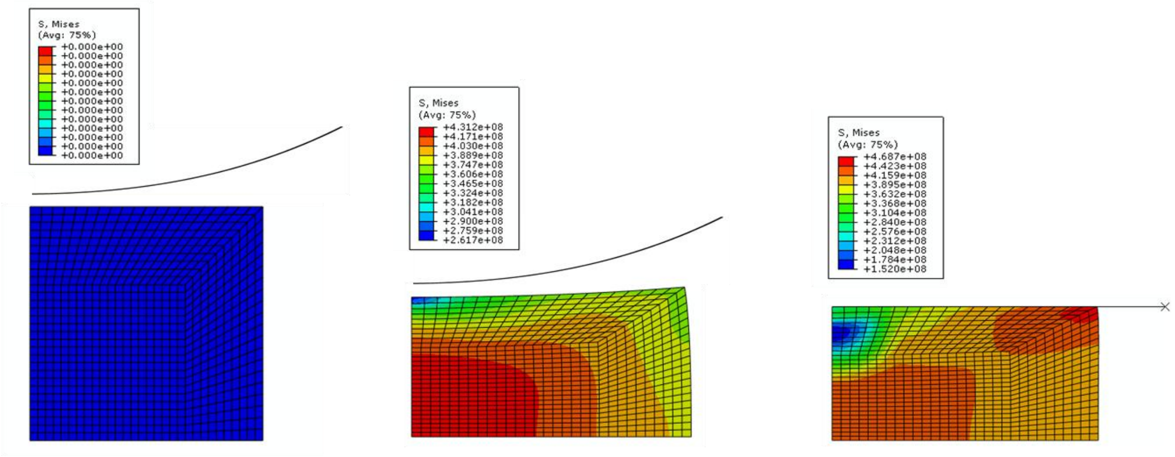
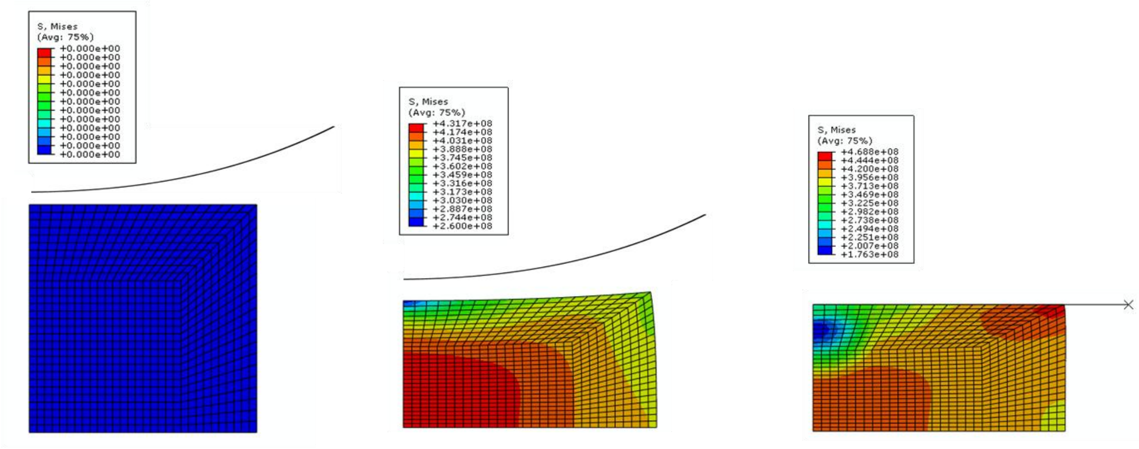


Figure 12.49 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 0 with initial displacement of -0.025 m.



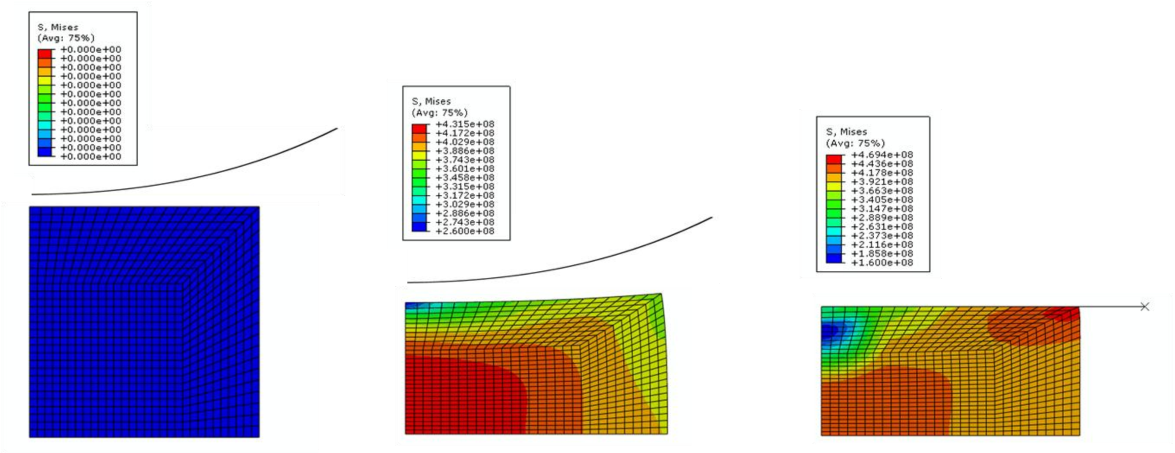
a) b) c)

Figure 12.50 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 25 with initial displacement of -0.025 m.



a) b) c)

Figure 12.51 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 55 with initial displacement of -0.025 m.



a) b) c)

Figure 12.52 - Initial and final shapes of the billet and the tools for the 9 optimization variables in iteration 84 with initial displacement of -0.025 m.

The shape of the first tool for the different iterations during the optimization process is presented in figures 13.52 to 13.55.

In Figure 12.57 the optimum first tool shapes for the four approaches considered are represented.

The most significant results obtained with the different four approaches are systematized in Table 12.7 and Table 12.8.

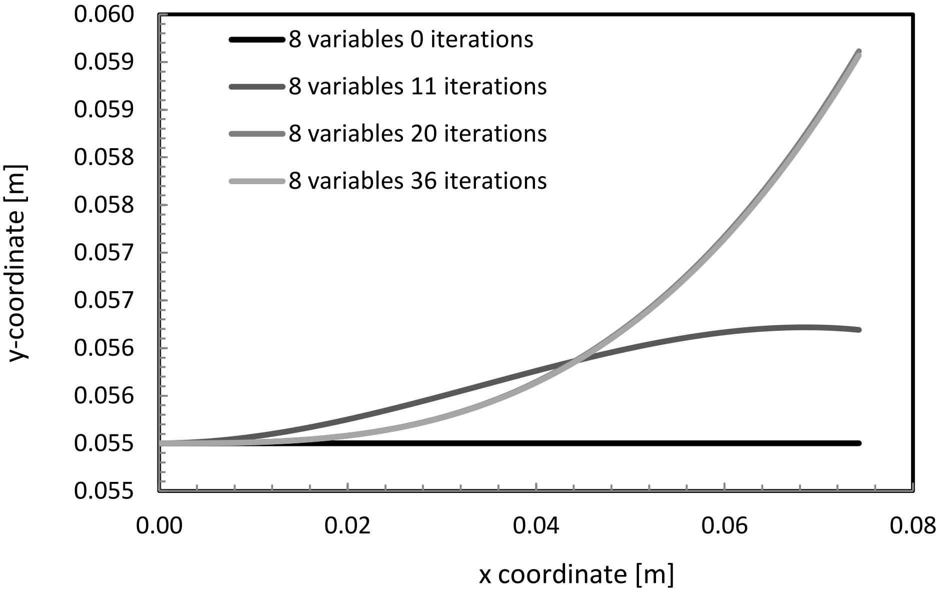


Figure 12.53 - Shape of the first tool for the 8 variables approach.

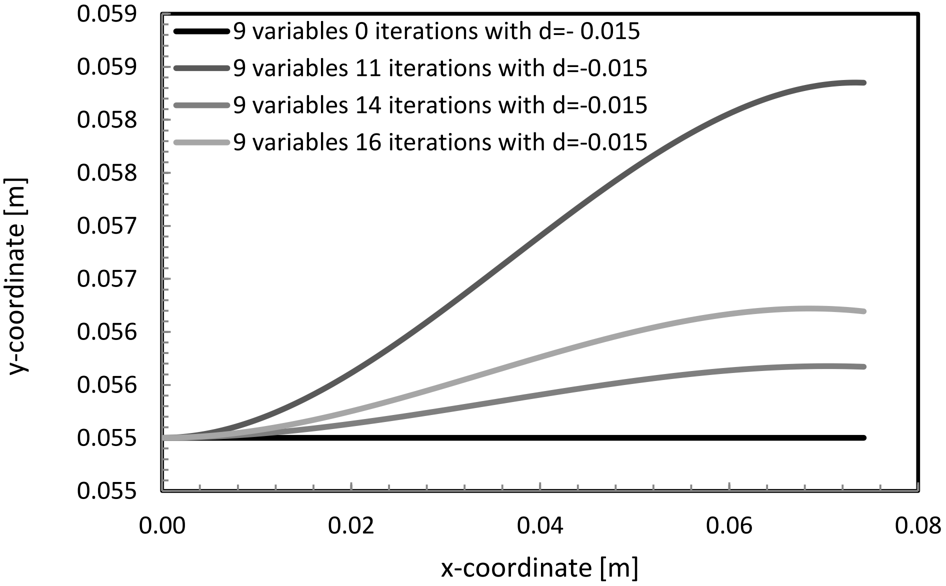


Figure 12.54 - Shape of the first tool for the 9 variables approach with starting displacement of -0.015 m.

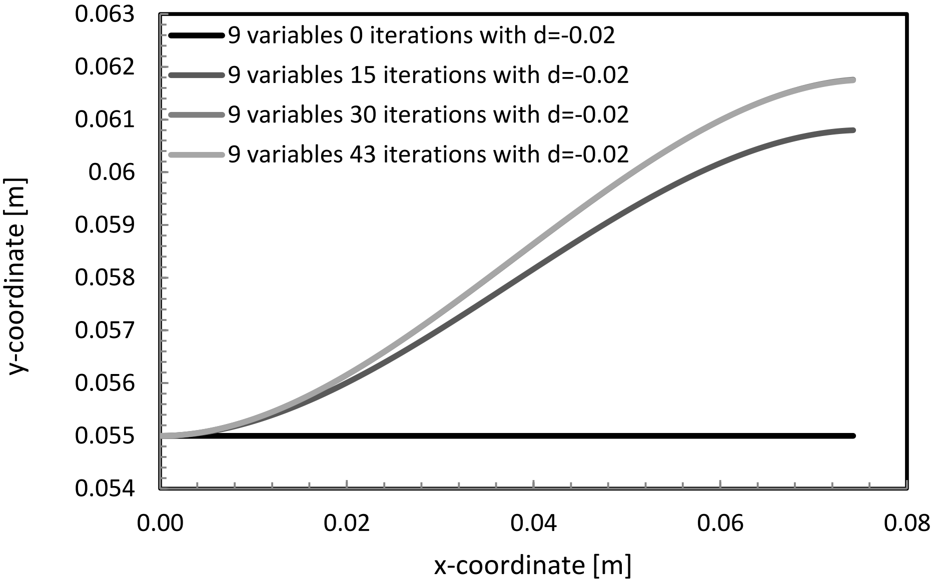


Figure 12.55 - Shape of the first tool for the 5 variables approach with starting displacement of -0.02 m.

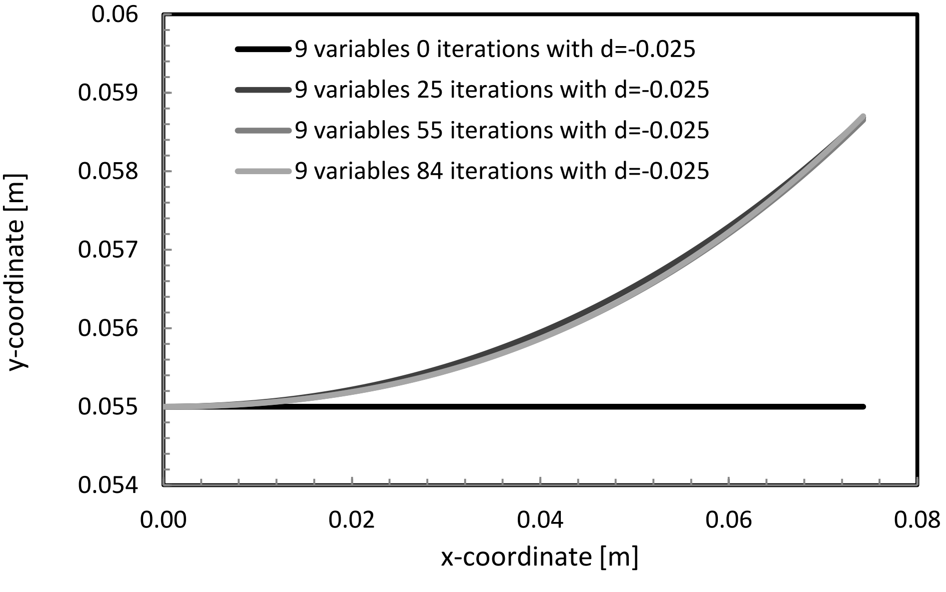


Figure 12.56 - Shape of the first tool for the 5 variables approach with starting displacement of -0.025 m.

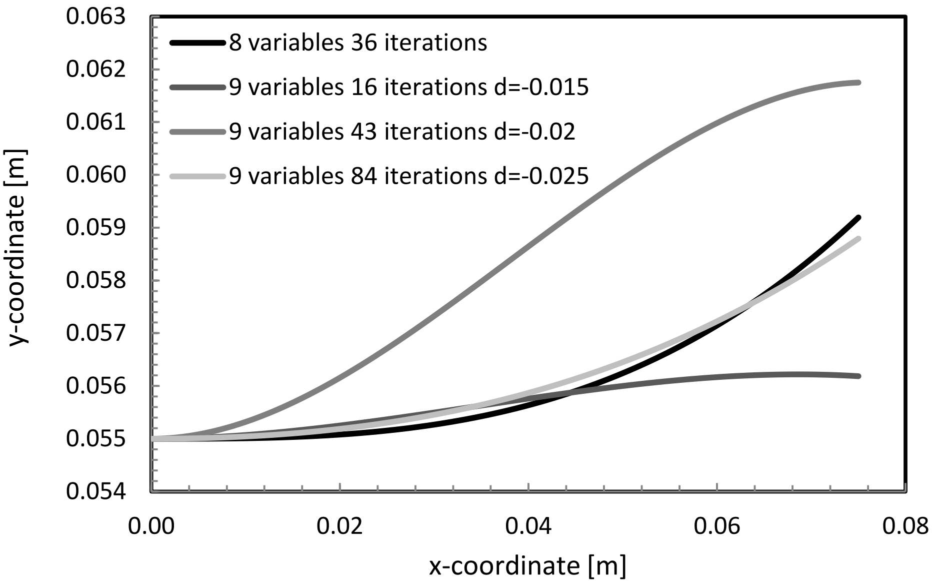


Figure 12.57 - Optimum shape of the first tool for the 4 approaches considered.

Table 12.7 - Results for the NURBS RSURFU implementation.

Parameterization method	Optimum objective function value	Number of iterations until the optimum	CPU time until the optimum[3] [s]
8 variables	$8.8648x10^{-8}$	36	2160
9 var. 0.015	$3.6550x10^{-5}$	16	960
9 var. 0.02	$1.0577x10^{-5}$	43	2580
9 var. 0.025	$2.0989 x10^{-7}$	84	5040

12.3.5 Discussion and conclusions

In Figure 12.12 and Figure 12.13 the evolution of the objective function values in function of the evaluations number for the Bézier RSURFU is presented. Considering these results and the ones systematized in the Table 12.5 it is possible to conclude that the method with five variables and initial displacement of -0.015 was the only method that didn´t lead to good results in terms of CPU time expended. However, all the four approaches lead to very good results when considering the results in terms of objective function value. When considering the objective function value the better results were obtained for the approach with five optimization variables with the starting value of displacement of -0.025 m.

[3] CPU time expended in an Intel® Core^{TM} 2 Quad CPU Q 9400 at 2.66GHz with 3.25GB of RAM.

Table 12.8 - Best parameters sets for the Bézier RSURFU implementation.

Parameterization method	x_1 [m]	x_2 [m]	x_3 [m]	x_4 [m]	x_5 [m]	x_6 [m]	x_7 [m]	x_8 [m]	x_9 [m]
8 variables	0.055	0.055	0.055	0.05919277	1.000355	1.001272	1.001142	0.9999793	-
9 var. 0.015	0.055	0.055	0.05645975	0.05618449	1.0	1.0	1.0	1.0	-0.01528399
9 var. 0.02	0.055	0.055	0.06158015	0.06175001	0.9997298	1.0	1.0	1.0	-0.02004371
9 var. 0.025	0.055	0.055	0.05573808	0.05879218	0.9995975	1.001160	1.002941	1.000853	-0.025

Considering the results obtained, one should choose the approach with five optimization variables and -0.025 starting displacement if wants a method that better minimize the considered objective function. Furthermore, the approach with four optimization variables should be considered in cases that a good balance between the objective function value and the computational time is desired.

In Figure 12.34 is represented the optimum shape of the first tool for the four studies conducted. As it was mentioned before all strategies lead to good resukts in terms of objective function value, however the optimum shapes are different. This fact shows that this optimization problem have a multiplicity of solutions.

In Figure 12.35 and Figure 12.36 is presented the objective function value in function of the evaluations number for the NURBS RSURFU. Considering these results and the ones presented in Table 12.7 it is possible to conclude that was for the method with 8 optimization variables that better results in terms of objective function value were reached. In this case it was considered that it was the 8 variables method that reached better results for a good balance between CPU time expended and objective function value.

With these results it is possible to conclude that the increase of the curve complexity from a Bézier curve to a NURBS curve didn´t lead to better results.

When comparing these results with the ones obtained for the initial geometry optimization of the billet it is possible to conclude that the initial geometry optimization of the billet generically lead to better results in terms of objective function and is less expensive in terms of computational cost.

As it was verified for the Bézier approach, in the NURBS approach, different optimum first tool shapes were obtained for the four methodologies. This fact underlines the existence of multiple solutions of the presented optimization problem.

In Figure 12.58 it is possible to observe a comparison between the better result achieved for the tool optimization approach and the one obtained by Igor Grešovnik [7]. In terms of objective function value the results obtained can´t be compared because they are dissimilar. This fact may occur because no correct information about the friction coefficient was found. In geometric terms both solutions lead to the main objective, reaching a straight cylindrical billet after the forging.

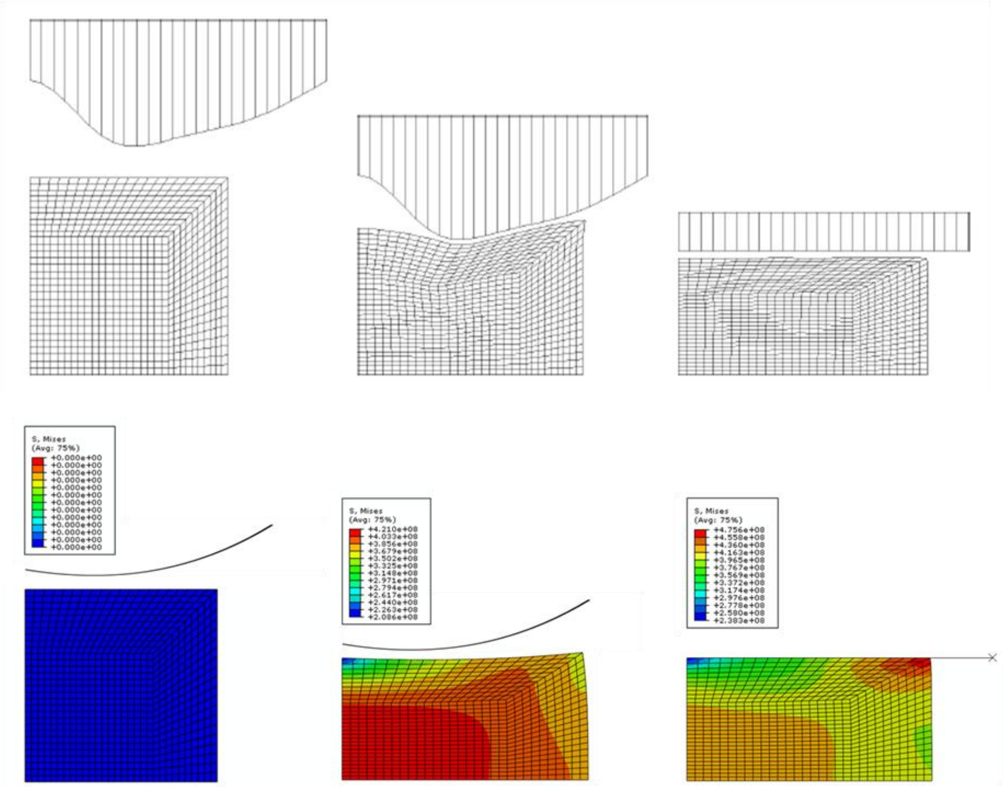


Figure 12.58 - Results comparison between the better result achieved for the tool optimization approach and the one obtained by Igor Grešovnik [7].

References

[1] Tang Y, Zhou X, Chen J (2008) Preform tool shape optimization and redesign based on neural network response surface methodology. Finite Elements in Analysis and Design 44:462-471.

[2] Castro C, Sousa L, António C, César de Sá J (2001) An efficient algorithm to estimate optimal perform die shape parameters in forging. Engineering computations 18:1057-1077.

[3] Vieilledent D, Fourment L (2001) Shape optimization of axisymmetric preform tools in forging using a direct differentiation method. International Journal for Numerical Methods in Engineering 52:1301-1321.

[4] Chenot J, Massoni E, Fourment L (1996) Inverse problems in finite element simulation of metal forming processes. Engineering Computations 13:190-225.

[5] António C, Castro C, Sousa L (2004) Optimization of metal forming processes. Computer and Structures 82:1425-1433.

[6] Kleinermann J (2000) Identification parametrique et optimisation des procedes de mise a forme par problemes inverses. PhD Thesis. University of Liége (in French).

[7] Grešovnik I (2000) A general purpose computational shell for solving inverse and optimization problems – Applications to metal forming processes. PhD Thesis. University of Wales.

[8] Santos A, Duarte J, Reis A, Barata da Rocha A, Neto R, Paiva R (1999) Finite element simulation of closed die forging for prediction of material behaviour and optimization of process. Proceedings of the Second ESAFORM Conference on Material Forming, Portugal.

[9] Rodrigues J, Martins P (2005) Tecnologia Mecânica – Tecnologia da Deformação Plástica Vol. II Aplicações Industriais. Escolar Editora.

V Final Remarks

Chapter 13

Conclusions and Future Works

The main conclusions of the present work are underlined. Some future works in the domain of the present work are presented.

13.1 General Conclusions

The present work was proposed considering that the interest of the stamping industry in numerical simulation of sheet metal forming, including inverse engineering approaches, is increasing. As it was mentioned before, the present work has, as the main objective, the development of numerical methodologies for parameter identification and shape optimization in metal forming simulations.

The quality of the Finite Element Analysis (FEM) results relies on the input data, such as, the material constitutive models. In order to achieve the best material parameters for the material constitutive models, parameter identification inverse problems are considered. Regarding this, in the present work, first an introduction to the kinematics mathematical formulation for continuum mechanics and constitutive modelling was performed in order to understand the behaviour of the materials. Also in this topic, it was described how the inverse problems can be classified and as a transversal theme the optimization algorithms were classified and explained in detail. A practical example of parameter identification was performed in chapter 9 where new optimization strategies that efficiently can lead to accurate material parameters were proposed. These strategies were proposed in order to take advantage of the strength of each selected algorithm and improve the overall robustness and efficiency of classical optimization methodologies based on single stages. Strategies such as, cascade, parallel and hybrid were considered. It was verified that, generically, the developed strategies lead to better values, in terms of objective function value, than the single stage optimization procedures. This fact confirms that these strategies take advantage of the strength of each selected algorithm and improve the overall robustness and efficiency of classical optimization methodologies based on single stages.

The parameters determination should always be performed confronting mathematical and experimental results. The experimental data should be accurate and the number of mechanical tests needed depends on the number of parameters that defines the constitutive model. In this work it was presented the experimental tests more used in the characterization of metallic sheets. Being these tests used to characterize mechanically the AA6082.

The evaluation of the objective function is one of the fundamental points that should be considered carefully. In the present work, a comparative study between the use of the single point and the FE analysis in the parameters' determination problems was performed. For the studied constitutive models, the single-point analysis was considered more efficient in terms of objective function value/computational cost relation. Also, for the constitutive models studied, it was considered that both strategies lead to good results in terms of objective function value. In the problems that the geometrical phenomena do not exist, the single-point is more appropriate considering the good relation between the CPU time and the objective function values. However, the FE analysis seems to be a more conservative approach.

Other main topic in this work starts from the idea that sheet metal forming is a complex deformation process controlled by parameters such as blank shape, tools' geometry, sheet thickness' values, blank holding force, friction, *etc.*. Considering this, in the present work, the inverse problems of blank and tool design were studied. To perform a correct study in this field It was necessary to describe in detail the metal forming processes, the Finite Element Method and the coupling between the FEM software and the optimization software needed to perform the shape optimization processes.

In this context a methodology to find the optimized initial blank shape for a carter forming processes was proposed. Also in this field, a study that allows to understand the influence of the parametric geometry definition in the optimization process was presented. Four studies were performed with differences in the number and location of control vertices that DEFINE the NURBS surface. The four studies had achieved good values in terms of objective function value. It was considered that the optimization results are influenced by the parametric definition of the blank. It was for the study with 16 optimization variables (25 control vertices) that the best value of objective function was achieved.

Concerning the study of the initial blank shape design and the study of the tool design, it was necessary to present in detail the parametric formulations used to define the curves and surfaces of these problems. Bézier, B-spline and NURBS curves (and surfaces) were described regarding the main scope of the presented work.

In the tool shape optimization thematic, the two stage forging of a cylindrical billet problem was proposed. Two different approaches, which try to find a desirable final shape of a specimen, after a forging process, were compared. The first approach optimizes the initial shape of the specimen and the second one optimizes the preform tool shape. With the results obtained it was possible to conclude that the increase of the curve complexity from a Bézier curve to a NURBS curve didn´t lead to better results. When comparing the two different approaches it is possible to conclude that the initial shape optimization of the billet generically leads to better results in terms of objective function value and in terms of computational cost.

Generically, in this work new optimization methodologies in the field of parameters identification problems, initial shape design and tool shape design are presented.

13.2 Future works

This work is a compilation of different research areas, all in the computational mechanics and engineering optimization field. Considering the work presented, some future guidelines of research are suggested. Some of the work presented are preliminary studies allowing having the possibility to be continued.

Concerning the new strategies presented for the parameter identification problems it is possible to suggest as a future work the use of new optimization algorithms with different potentialities, trying also to improve the overall robustness and efficiency of classical optimization methodologies based on single stages. Additionally, a study considering the presented strategies can be extended to other different constitutive models, in order to understand how these behaves for other constitutive models.

In the present work, the use of single-point or metal forming FE analysis for the evaluation of the objective function in the identification process is another research line that could be improved. Also in this case, this study can be extended to other constitutive models in order to generalize the conclusions performed. An interesting study that can be accomplished is the use of ARAMIS experimental results in the evaluation of the objective function.

A blank shape design and a study of the influence of the geometry definition are presented in chapter 11. The study presented can be continued considering a study where other parametric definitions, such as Bézier surfaces, B-spline surfaces, among others, are used in the geometric definition of the upper-surface of the blank. This study will allow to understand how different parametric surfaces influence the optimization procedure.

In the chapter 12, a tool shape optimization problem is proposed. The mechanical problem considered is the two stage forging of a cylindrical billet, a merely academic example. However, that allows to validate the optimization methodologies that could be applied in real optimization processes. This study can be improved in few ways. One possible improvement is the use of the validated RSURFU to more complex mechanical engineering problems. In the meanwhile, extensions to this work are being started within the research group the author is involved with. Also an interesting improvement is the extension of the considered RSURFU formulations to 3D cases.

Appendix

Table A.1 - RSURFU code for the Bézier RSURFU.

```
      subroutine rsurfu(h,p,tgt,dnds,x1,time,u,ciname,slname,msname,
     1                  noel,node,lclose)
c
      include 'aba_param.inc'
c
      character*80 ciname,slname,msname
      double precision  tgt(3,2),dnds(3,2),x1(3,2),u(6,2),h,P(3)
      double precision  A(3),yr,CV(4,3),step,m,distf
      double precision  bxm,bym,dist,pos
      double precision DSDT1m,DSDT2m,DDSDDT1m,DDSDDT2m
      double precision tang1m,tang2m,n(3)
      double precision  normDSDT,cross(3),normcross,k,v,T(3),AAl(3)


c set abig to: 1.d27 for vax under vms, 1.d30 for non-unix ibm,
c              1.d40 all other machines.
c
      parameter( zmax = 25.d0, zero  = 0.d0,
     +           one  =  1.d0, two    = 2.d0,  abig   = 1.d27,
     +           small = 1.d-8 )
c
c initialize variables to be returned by rsurfu
c

      h = -abig
      do 50 k1=1, 2
        do 25 k2=1, 3
           tgt(k2,k1)  = zero
           dnds(k2,k1) = zero
           p(k2)       = zero
 25   continue
 50   continue

c
c coordinates on deforming body
c

      A(1) = x1(1,1)
      A(2) = x1(2,1)
      A(3) = x1(3,1)

c
c simplification needed for convergence
c

      if (x1(1,1)<0.)then
        A(1) = 0.
      end if
```

```
c
c tool displacement
c

      yr = u(2,2)

c
c import the control vertices
c
      open(16,file='C:\data.txt')

      read (16,*) CV(1,1)
      read (16,*) CV(1,2)
      read (16,*) CV(1,3)
      read (16,*) CV(2,1)
      read (16,*) CV(2,2)
      read (16,*) CV(2,3)
      read (16,*) CV(3,1)
      read (16,*) CV(3,2)
      read (16,*) CV(3,3)
      read (16,*) CV(4,1)
      read (16,*) CV(4,2)
      read (16,*) CV(4,3)

      close(16)

c
c find closest point P(3)- position of the point A´ on the surface
c                          of the rigid body
c


      step=1./99.
      m=0
      distf=100000.


      do while (m<99)

      bxm=((1-m*step)**3.)*CV(1,1)+(3.*m*step*(1-m*step)**2.)*CV(2,1)
     &+3.*(m*step)**2.*(1-m*step)*CV(3,1)+(m*step)**3.*CV(4,1)
      bym=((1-m*step)**3.)*CV(1,2)+(3.*m*step*(1-m*step)**2.)*CV(2,2)
     &+3.*(m*step)**2.*(1.-m*step)*CV(3,2)+(m*step)**3.*CV(4,2)

      dist=sqrt((A(1)-bxm)**2+(A(2)-(bym+yr))**2)

      if (dist<distf)then
        distf=dist
        pos=m
      else
        distf=dist
      end if

      m=m+1

      end do

c         Determination of the coordinates of point A´ and its first and
c         second derivatives
```

```
      bxm=((1-pos*step)**3.)*CV(1,1)+(3.*pos*step*(1-pos*step)**2.)
     &*CV(2,1)
     &+3.*(pos*step)**2.*(1-pos*step)*CV(3,1)+(pos*step)**3.*CV(4,1)
      bym=((1-pos*step)**3.)*CV(1,2)+(3.*pos*step*(1-pos*step)**2.)
     &*CV(2,2)
     &+3.*(pos*step)**2.*(1.-pos*step)*CV(3,2)+(pos*step)**3.*CV(4,2)
      DSDT1m=-3.*(1.-pos*step)**2.*CV(1,1)+(3.*(1.-pos*step)**2.-6.
     &*pos*step*(1.-pos*step))*CV(2,1)
     &+(6.*pos*step*(1.-pos*step)+(-3.*(pos*step)**2.))*CV(3,1)+3.*
     &(pos*step)**2.*CV(4,1)
      DSDT2m=-3.*(1.-pos*step)**2.*CV(1,2)+(3.*(1.-pos*step)**2.
     &-6.*pos*step*(1.-pos*step))*CV(2,2)
     &+(6.*pos*step*(1.-pos*step)+(-3.*(pos*step)**2.))*CV(3,2)+3.
     &*(pos*step)**2.*CV(4,2)
      DDSDDT1m=6.*(1-pos*step)*CV(1,1)-6.*(2.-3.*pos*step)*CV(2,1)
     &+6.*(1-3.*pos*step)*CV(3,1)+6.*pos*step*CV(4,1)
      DDSDDT2m=6.*(1.-pos*step)*CV(1,2)-6.*(2.-3.*pos*step)*CV(2,2)
     &+6.*(1.-3.*pos*step)*CV(3,2)+6.*pos*step*CV(4,2)

      tang1m=-DSDT1m/sqrt(DSDT2m**2.+DSDT1m**2.)
      tang2m=-DSDT2m/sqrt(DSDT2m**2.+DSDT1m**2.)

c         Determination of the normal vector

      n(1)=-tang2m
      n(2)= tang1m
      n(3)=0

      P(1)= bxm
      P(2)= bym +yr
      P(3)= A(3)

c
c Determination of the direction cosines of the two unit tangents to the
c surface t1 and t2 at point A´.
c

      TGT(1,1)  = tang1m
      TGT(2,1)  = tang2m
      TGT(3,2)  =-1.

c
c Determination of the rates of change of the normal surface, n, at A´,
c with respect to distance measuring coordinates S1 and S2 along t1 and
c t2.
c
c
c         Considering the frenet equations to compute de DNDS and
c         considering this a planar curve where torsion doesn´t exist
c         DNDS = -kT
c         k = curvature function
c         v = velocity of the curve
c         T = unit tangent
c
c         calculation of the norm of the DSDT (first derivative of the
c         curve)

          normDSDT = sqrt(DSDT1m**2.+DSDT2m**2+DSDT3m**2.)
```

```
c
c         curvature function k
c
c         cross product between first and second derivatives of the
c         curve
c         DSDT X DDSDDT

          cross(1) =  DSDT2m*DDSDDT3m - DDSDDT2m*DSDT3m
          cross(2) = -DSDT1m*DDSDDT3m + DDSDDT1m*DSDT3m
          cross(3) =  DSDT1m*DDSDDT2m - DDSDDT1m*DSDT2m

c         norm of the cross product between first and second
c         derivatives  of the curve ||DSDT X DDSDDT||
!
          normcross = sqrt (cross(1)**2.+cross(2)**2.+cross(3)**2.)
          k = normcross /((normDSDT)**3.)
c
c         velocity of the curve v = ||DSDT||
c
          v = normDSDT
c
c         unit tangent T
c
          T(1) = -DSDT1m / normDSDT
          T(2) = -DSDT2m / normDSDT
          T(3) = -DSDT3m / normDSDT

      DNDS(1,1) =-( - k * T(1))
      DNDS(2,1) =-( - k * T(2))

c
c Determination of the penetration(H) of the point A on the deforming
c structure into the surface of the rigid body, measured down the
c outward
c normal to the rigid surface.
c

      do i=1,3
        AAl(i)=P(i)-A(i)
      end do

      H=AAl(1)*n(1)+AAl(2)*n(2)+AAl(3)*n(3)

      return
      end

        end
```

Table A.2 - RSURFU code for the NURBS RSURFU.

```
      subroutine rsurfu(h,p,tgt,dnds,x1,time,u,ciname,slname,msname,
     1                  noel,node,lclose)
c
      include 'aba_param.inc'
c
      character*80 ciname,slname,msname
      double precision  tgt(3,2),dnds(3,2),x1(3,2),u(6,2),h,P(3)
      double precision  A(3),yr,CV(4,3),h1,h2,h3,h4,step,m,distf
      double precision  sum,bxm,bym,dist,pos
      double precision dsum,DSDT1m,DSDT2m,DDSDDT1m,DDSDDT2m
      double precision dR14,dR24,dR34,dR44
      double precision ddsum,ddR14,ddR24,ddR34,ddR44
      double precision tang1m,tang2m,n(3)
      double precision  normDSDT,cross(3),normcross,k,v,T(3),AA1(3)


c set abig to: 1.d27 for vax under vms, 1.d30 for non-unix ibm,
c              1.d40 all other machines.
c
      parameter( zmax = 25.d0, zero  = 0.d0,
     +           one  =  1.d0, two    = 2.d0,  abig   = 1.d27,
     +           small = 1.d-8 )

c
c initialize variables to be returned by rsurfu
c

      h = -abig
      do 50 k1=1, 2
        do 25 k2=1, 3
           tgt(k2,k1)  = zero
           dnds(k2,k1) = zero
           p(k2)       = zero
 25   continue
 50   continue

c
c coordinates on deforming body
c

      A(1) = x1(1,1)
      A(2) = x1(2,1)
      A(3) = x1(3,1)

c
c simplification needed for convergence
c

      if (x1(1,1)<0.)then
        A(1) = 0.
      end if

c
c tool displacement
c
```

```
      yr = u(2,2)

c
c import the control vertices
c
      open(16,file='C:\data.txt')

      read (16,*) CV(1,1)
      read (16,*) CV(1,2)
      read (16,*) CV(1,3)
      read (16,*) CV(2,1)
      read (16,*) CV(2,2)
      read (16,*) CV(2,3)
      read (16,*) CV(3,1)
      read (16,*) CV(3,2)
      read (16,*) CV(3,3)
      read (16,*) CV(4,1)
      read (16,*) CV(4,2)
      read (16,*) CV(4,3)
      read (16,*) h1
      read (16,*) h2
      read (16,*) h3
      read (16,*) h4
      close(16)

c
c find closest point P(3)- position of the point A´ on the surface
c                          of the rigid body
c


      step=1./99.
      m=0
      distf=100000.


      do while (m<99)

      sum=((1-m*step)**3.)*h1+(3.*m*step*(1-m*step)**2.)*h2
     &+3.*(m*step)**2.*(1-m*step)*h3+(m*step)**3.*h4

      bxm=(((1-m*step)**3.)*h1*CV(1,1)+(3.*m*step*(1-m*step)**2.)
     &*h2*CV(2,1)+3.*(m*step)**2.*(1-m*step)*h3*CV(3,1)
     &+(m*step)**3.*h4*CV(4,1))/sum


      bym=(((1-m*step)**3.)*h1*CV(1,2)+
     &(3.*m*step*(1-m*step)**2.)*h2*CV(2,2)
     &+3.*(m*step)**2.*(1.-m*step)*h3*CV(3,2)
     &+(m*step)**3.*CV(4,2)*h4)/sum


      dist=sqrt((A(1)-bxm)**2+(A(2)-(bym+yr))**2)

      if (dist<distf)then
        distf=dist
        pos=m
      else
        distf=dist
      end if

      m=m+1

      end do

c         Determination of the coordinates of point A´ and its first and
c         second derivatives

       m=pos
       sum=((1-m*step)**3.)*h1+(3.*m*step*(1-m*step)**2.)*h2
     &+3.*(m*step)**2.*(1-m*step)*h3+(m*step)**3.*h4

      bxm=(((1-m*step)**3.)*h1*CV(1,1)+(3.*m*step*(1-m*step)**2.)
     &*h2*CV(2,1)+3.*(m*step)**2.*(1-m*step)*h3*CV(3,1)
     &+(m*step)**3.*h4*CV(4,1))/sum


      bym=(((1-m*step)**3.)*h1*CV(1,2)+
     &(3.*m*step*(1-m*step)**2.)*h2*CV(2,2)
     &+3.*(m*step)**2.*(1.-m*step)*h3*CV(3,2)
     &+(m*step)**3.*CV(4,2)*h4)/sum



      dsum=3*(-h1+h2)+6*m*step*(h1-2*h2+h3)+
     &3*(m*step)**2*(-h1+3*h2-3*h3+h4)

      ddsum=6*(h1-2*h2+h3)+6*(m*step)*(-h1+3*h2-3*h3+h4)


      dR14=(h1*(-3*(1-m*step)**2))/sum
     &-((h1*(1-m*step)**3)*dsum)/(sum**2)

      dR24=(h2*(3-12*m*step+9*(m*step)**2))/(sum)
     &-((h2*3*m*step*(1-m*step)**2)*dsum)/(sum**2)

      dR34=(h3*(6*m*step-9*(m*step)**2))/(sum)
     &-((h3*3*(m*step)**2*(1-m*step))*dsum)/(sum**2)


      dR44=(h4*3*(m*step)**2)/(sum)
     &-(h2*(m*step)**3*dsum)/(sum**2)

      ddR14=(h1*(6*(1-m*step)))/(sum)
     &-((2*h1*(-3*(1-m*step)**2))*dsum)/(sum**2)
     &-(ddsum*h1*(1-m*step)**3)/(sum**2)
     &+(2*h1*((1-m*step)**3)*dsum**2)/(sum**3)



      ddR24=(h2*(-12+18*m*step))/(sum)
     &-((2*h2*(3*(1-m*step)**2)-6*(1-m*step))*dsum)/(sum**2)
     &-(h2*(3*m*step*(1-m*step)**2)*ddsum)/(sum**2)
     &+((2*h2*(3*(1-m*step)**2))*(dsum**2))/(sum**3)


      ddR34=(h3*(6-18*m*step)/(sum)
     &-((2*h3*(6*m*step*(1-m*step)-3*(m*step)**2))*dsum)/(sum**2)
     &-((h3*(3*((m*step)**2)*(1-m*step)))*ddsum)/(sum**2)
     &+(2*h3*((3*(m*step)**2)*(1-m*step)))*dsum**2)/(sum**3)



      ddR44=(h4*6*m*step)/(sum)
     &-(2*h4*3*((m*step)**2)*dsum)/(sum**2)
     &-(ddsum*h4*(m*step)**3)/(sum**2)
     &+(2*h4*((m*step)**3)*dsum)/(sum**3)


      DSDT1m=CV(1,1)*dR14+CV(2,1)*dR24+CV(3,1)*dR34+CV(4,1)*dR44

      DSDT2m=CV(1,2)*dR14+CV(2,2)*dR24+CV(3,2)*dR34+CV(4,2)*dR44

      DDSDDT1m=CV(1,1)*ddR14+CV(2,1)*ddR24+CV(3,1)*ddR34+CV(4,1)*ddR44

      DDSDDT2m=CV(1,2)*ddR14+CV(2,2)*ddR24+CV(3,2)*ddR34+CV(4,2)*ddR44

      tang1m=-DSDT1m/sqrt(DSDT2m**2.+DSDT1m**2.)
      tang2m=-DSDT2m/sqrt(DSDT2m**2.+DSDT1m**2.)

c         Determination of the normal vector

      n(1)=-tang2m
      n(2)= tang1m
      n(3)=0

      P(1)= bxm
      P(2)= bym +yr
      P(3)= A(3)

c
c Determination of the direction cosines of the two unit tangents to the
c surface t1 and t2 at point A´.
c

      TGT(1,1)  = tang1m
      TGT(2,1)  = tang2m
      TGT(3,2)  =-1.

c
c Determination of the rates of change of the normal surface, n, at A´,
c with respect to distance measuring coordinates S1 and S2 along t1 and
c t2.
c
c
c         Considering the frenet equations to compute de DNDS and
c         considering this a planar curve where torsion doesn´t exist
c         DNDS = -kT
c         k = curvature function
c         v = velocity of the curve
c         T = unit tangent
c
c         calculation of the norm of the DSDT (first derivative of the
c         curve)

          normDSDT = sqrt(DSDT1m**2.+DSDT2m**2+DSDT3m**2.)
```

```
c
c         curvature function k
c
c         cross product between first and second derivatives of the
c         curve
c         DSDT X DDSDDT

          cross(1) =  DSDT2m*DDSDDT3m - DDSDDT2m*DSDT3m
          cross(2) = -DSDT1m*DDSDDT3m + DDSDDT1m*DSDT3m
          cross(3) =  DSDT1m*DDSDDT2m - DDSDDT1m*DSDT2m

c         norm of the cross product between first and second
c         derivatives  of the curve ||DSDT X DDSDDT||
!
          normcross = sqrt (cross(1)**2.+cross(2)**2.+cross(3)**2.)
          k = normcross /((normDSDT)**3.)
c
c         velocity of the curve v = ||DSDT||
c
          v = normDSDT
c
c         unit tangent T
c
          T(1) = -DSDT1m / normDSDT
          T(2) = -DSDT2m / normDSDT
          T(3) = -DSDT3m / normDSDT

      DNDS(1,1) =-( - k * T(1))
      DNDS(2,1) =-( - k * T(2))

c
c Determination of the penetration(H) of the point A on the deforming
c structure into the surface of the rigid body, measured down the
c outward
c normal to the rigid surface.
c

      do i=1,3
        AAl(i)=P(i)-A(i)
      end do

      H=AAl(1)*n(1)+AAl(2)*n(2)+AAl(3)*n(3)

      return
      end

        end
```