UNIVERSITA' DEGLI STUDI DI BERGAMO

Facoltà di Ingegneria

Dipartimento di Ingegneria Industriale

DOTTORATO DI RICERCA

IN

TECNOLOGIE PER L'ENERGIA E L'AMBIENTE

XXV ciclo

Anno 2010-2014

Optimization Techniques Applied to Mechanical Design

DoctoralThesis:

Ing.Bonini Claudio

Supervisor:

Prof.Francesco Bassi

Abstract

The present work deals with the application of optimization techniques to mechanical design. After an initial section devoted to a theoretical review of optimization algorithms in common use, the thesis consists of two parts.

The first part is about the use of the adjoint method in the framework of aerodynamic shape optimization. After a theoretical review, the discrete adjoint method has been implemented in a research code based on the Discontinuous Galerkin (DG) method. This activity represents a new direction of development within the research group operating at University of Bergamo on DG methods for Computational Fluid Dynamics (CFD). Starting from the simple quasi-1D Euler equations, the implementation of the discrete adjoint method has been validated by comparing the values of computed adjoint variables with results of analytical solutions available in the literature. The method has then been applied to a shape optimization problem, using a gradient based algorithm with an inexact line search approach. The values of derivatives in the course of optimization were found to be in close agreement with those obtained by means of the finite difference approach.

The second part of the thesis deals with the application of optimization techniques to an industrial problem. This activity has been carried out at the R&D Centre of TenarisDalmine S.p.A., one of the largest seamless steel pipe producers in the world. This work focuses on the optimization of the thermal cycle of the mandrel of a longitudinal mandrel mill, with the objective of reducing the peak temperature of the mandrel during the rolling phase. The mandrel is a very expensive component, whose life cycle has a significant impact on the final cost of the product, especially for mandrel mills devoted to the production of restricted ranges of pipe sizes. The activity for this part of the thesis required the preliminary set up of a number of computational tools for the analysis of the physical aspects involved in the problem. Such tools have then been integrated in a comprehensive optimization approach driven by the optimization tools available in the Optimization Toolbox of the commercial software Matlab. The results of optimization are encouraging, showing the possibility of a considerable increase of the mandrel life cycle and highlighting the advantages of using optimization techniques in the design process.

Keywords: optimization techniques, computational fluid dynamics, shape optimization, adjoint method, Euler equations, discontinuous Galerkin, longitudinal mandrel mill, thermal cycle optimization.

Acknowledgments

A sincere acknowledgment to the research group operating at University of Bergamo on DG methods for Computational Fluid Dynamics, headed by Prof.Francesco Bassi, for the provided support. In alphabetic order, many thanks to Drs.Lorenzo Botti, Alessandro Colombo and Nicoletta Franchina.

A sincere acknowledgment to Professors Francesco Bassi, Elvio Cossali and Marco Savini for giving me the opportunity to conclude the PhD after my moving to the R&D Centre of TenarisDalmine S.p.A. .

A sincere acknowledgment to Dr.Giuseppe Benzoni for the provided advise at the time of the PhD choice and to Ing.Pierangelo Conti for the provided support.

A sincere acknowledgment to all the R&D Centre of TenarisDalmine S.p.A., focused on the improvement of rolling processes, for the provided help.

Alla mia famiglia…

Alla cosa più bella che la vita mi ha regalato...ad Anna…

Contents

List of Figures

List of Tables

1 Introduction

In the last 30 years the *engineering design process* has deeply changed, following the evolution of the involved tools. This evolution passed through different important steps. First of all, the development of *computer aided design* (*CAD*) techniques has allowed engineers to quickly manage components geometry and to handle very complex assemblies, something impossible even to think using the drawing board. Furthermore, the evolution of *numerical simulations* in each field made it possible to design each component without the need to execute expensive experimental tests. Currently tests are still used but with the aim to verify that the final designed configuration effectively matches the predefined target and to investigate unknown phenomena. This evolution supported engineers in increasing the performances of the designed components, reducing time and cost of the entire process. A further improvement of the design process is the use of the *optimization techniques.* These, starting from a determined configuration, are able to provide engineers with the system responses and to drive the design process automatically towards an *optimal* solution. This further step has allowed to increase the performances of very complex and multidisciplinary systems. The previously briefly described methods are still matter of research by many working groups looking for improved efficiency.

The present work deals with the *application* of optimization techniques in the mechanical design process, focusing on the used *optimization algorithms*.

1.1 Terminology used in Optimization

What is meant by "*optimization*"?

The online Cambridge Dictionary [**1**] provides the following definition:

"the act of making something as good as possible".

The online Reference Dictionary [**2**] provides another useful definition:

"a mathematical technique for finding a maximum or minimum value of a function of several variables subjected to a set of constraints".

The previous definitions are useful to introduce the topic of the optimization. Obviously the first one is more general but explains in absolute sense the effective goal of an optimization process. The second definition, instead, is much more practical and near to the real aspects involved in an optimization process.

The present section introduces the topic of optimization by explaining the meaning of the words appearing in the two definitions. In this sense, keeping in mind the first definition, the second one will be taken as a reference:

- *variables*: every problem has a set of *input* variables that provides one or more outputs. Thus, the term variables refers to the set of input variables of the optimization problem which can be changed, in order to obtain different configurations and, so, different outputs. Each set of input variables can be represented by a dot in the *design space* representing the domain of an optimization problem. Each single dot is a *sample* of the problem. For instance, in a problem of aerodynamic shape optimization, the variables are the set of *geometrical parameters* that controls the shape of an airfoil;
- *function*: this function is referring to the *objective function* of an optimization problem. In the engineering framework, it is usually an engineering quantity to be improved (and hence to be optimized). The number of objective functions of an optimization problem can be one or more, creating, in the second case, a *multi-objective* optimization problem. The process

which starting from the input variables achieves the set of objected functions may be of any kind: a numerical simulation, an experimental test or others. Using the same example of aerodynamic shape optimization, the objective function may be the *drag* of an airfoil;

- *maximum or minimum value*: the scope of an optimization problem is always the maximization or the minimization of the objective function under investigation. In the aerodynamic shape optimization example, the *drag* of an airfoil is the quantity to be generally minimized;
- *constraints*: constraints are value limits of the input variables. Not only bound limits can be present but also more sophisticated limits can exist. In the aerodynamic shape optimization example, the *dimension* of an airfoil is in general subjected to some limits, such as maximum dimension allowed for the aircraft or minimum volume required for the fuel storage;
- *mathematical technique*: it is referred to the technique and the relative *optimization algorithm* by which an optimizer varies the set of input variables in order to decrease (or increase) the objective function, reaching a better configuration. The technique itself represents the heart of the optimization framework. The most used techniques will be explained in detail in chapter 2 and the application of those to engineering problems is object of the present work.

This is only a brief introduction to the topic of optimization, whose aim is to provide a basic idea as a starting point for a deeper study of the involved aspects.

1.2 Different Phases of an Optimization Process

The topic of optimization can be partitioned in the following three macro areas:

- *Design of Experiments*;
- *Optimization Algorithms*;
- *Robust Design Analysis*.

The present work focuses on the second area, relative to optimization algorithms. This paragraph anyway presents a general overview of the three areas with the aim to better understand how optimization algorithms are inserted in an optimization process.

1.2.1 Design of Experiments and Response Surface Modelling

The *design of experiment* (*DOE*) is not an optimization technique itself, but it is a technique used to identify which samples to choose in the design space, in order to get the maximum number of information using the minimum amount of resources. For a given number of samples, there are different ways of choosing an optimal sample arrangement for collecting different information.

The *DOE* is generally followed by a *response surface modelling* (*RSM*), a technique used to interpolate or approximate the information obtained by *DOE*. Different *RSM* techniques exist, with different interpolation or approximation criteria. The advantage represented by using *RSM* techniques is the fact that optimization techniques can be used directly on the obtained interpolation model. This optimization is very quick since it is based on the analytical evaluation of the interpolating or the approximating function.

The use of a *DOE+RSM* technique is cheaper than any optimization algorithm, since a lower number of samples is generally required. The drawback is represented by the fact that it's not possible to know how the approximation process is accurate.

1.2.2 Optimization Algorithms

As explained in the paragraph 1.1, optimization algorithms are mathematical algorithms used to change the set of input variables, in order to decrease (or increase) the objective function, towards a better configuration. The present work focuses on this aspect of the optimization topic: chapter 2 is devoted to a detailed description of the theory of optimization algorithms. In order to provide a general overview of optimization algorithms, in the present paragraph the following basic classification is presented:

- *deterministic optimization*: it refers to algorithms where a rigid mathematical scheduling is followed and no random elements are present;
- *gradient-based optimization*: it is a subset of deterministic optimization and it refers to algorithms that require the computation of the gradient of the objective function in the neighborhood of a sample;
- *stochastic optimization*: it refers to algorithms where randomness is present in the search procedure;
- *evolutionary optimization*: it is a subset of stochastic optimization and it refers to algorithms where the search procedure is carried out mimicking the evolution theory of Darwin;
- *genetic optimization*: it is a subset of genetic optimization and it refers to algorithms where the input variables are discretized in binary strings;
- *unconstrained optimization*: it refers to algorithms where the input variables are unconstrained;
- *constrained optimization*: it refers to algorithms where the input variables are constrained;
- *single-objective optimization*: it refers to algorithms where there is a single objective function;
- *multi-objective optimization*: it refers to algorithms where there are more than one objective function;
- *local optimization*: it refers to algorithms that can stop their search at local minima;
- *global optimization*: it refers to algorithms that can overcome local minima;
- *convex optimization*: it is a subset of gradient-based optimization and it refers to convex optimization problems;
- *discrete optimization*: it refers to algorithms which are able to include non-continue variables; for instance variables with only integer values.

1.2.3 Robust Design Analysis

The *robust design analysis* (*RDA*) aims at evaluating how a small change of input variables reflects on the objective function. This analysis is motivated by the fact that an optimal solution could degrade its performance very quickly as soon as some incontrollable parameters (for example noise factors) enter the game. Two different *RDA* approaches are possible: *multi-objective robust design optimization* and *reliability analysis*.

Multi-objective robust design optimization (*MORDO*) consists in further sampling with a given probability distribution the noise factors in the neighborhood of a sample. In this way it's possible to evaluate the standard deviation of the objective function of interest.

Reliability analysis (*RA*) consists in sampling with a given probability distribution the noise factors in the neighborhood of a sample too. However, this method aims to evaluate that the probability of the problem performance drops below a given minimum performance value, called *failure probability*.

Figure 1-1 shows the elements involved in an optimization process.

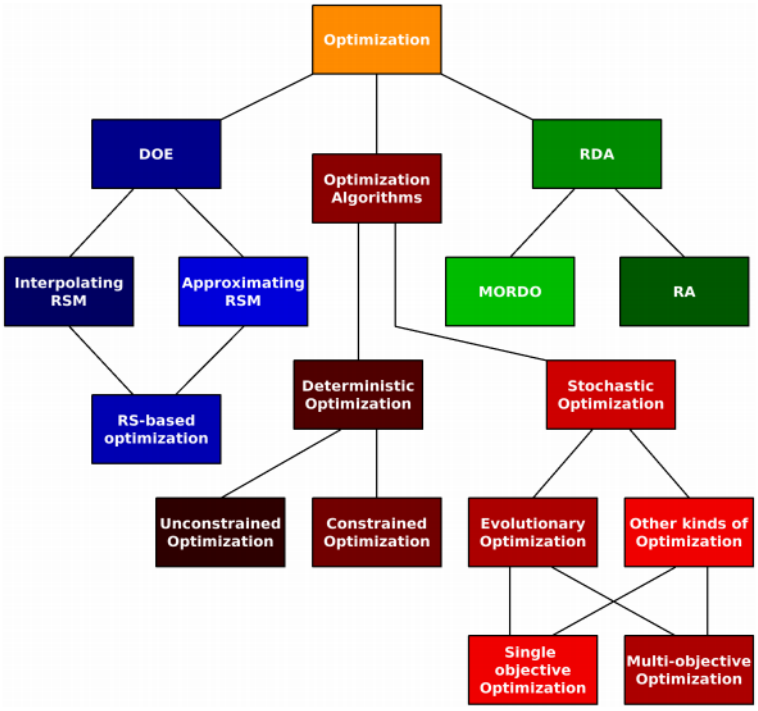

Figure 1-1: Elements Involved in an Optimization Process

1.3 Layout of the Thesis

The present Thesis is structured as follows.

Chapter 1 is an introduction to the topic of optimization.

Chapter 2 is devoted to a first theoretical review of commonly used optimization algorithms. Section 2.1 contains a theoretical survey concerning algorithms used to perform deterministic constrained and unconstrained optimization; while section 2.2 is devoted to stochastic optimization algorithms.

Chapters 3 and 4 deal with the application of optimization techniques to engineering problems.

Chapter 3 is devoted to the use of the *adjoint method* in the framework of aerodynamic shape optimization. After a theoretical review, the discrete adjoint method has been implemented in a research code based on the Discontinuous Galerkin (DG) method. This activity represents a new direction of development within the research group operating at University of Bergamo on DG methods for Computational Fluid Dynamics (CFD).

Chapter 4 deals with the application of optimization techniques to an industrial problem. This activity has been carried out at the R&D Centre of TenarisDalmine S.p.A., one of the largest seamless steel pipe producers in the world. This work focused on the *optimization of the thermal cycle* of the mandrel of a longitudinal mandrel mill, with the objective of reducing the peak temperature of mandrel during the rolling phase. The optimization has been implemented by the use of the Optimization Toolbox of the commercial software Matlab.

Chapter 5 is devoted to conclusions of the present work.

2 Optimization Theory

The present chapter is devoted to a general description of *optimization algorithms*: a theoretical review is provided with the aim to understand the choice of algorithms used in the present work.

The reader should be aware that such description is far from being a complete review: it only mentions the main algorithms and their basic concept.

About the references used in the review, in [**3**] a general overview of *deterministic* and *stochastic* optimization methods is present, while [**4**], [**5**] and [**6**] include a detailed description of algorithms concerning *unconstrained* and *constrained deterministic* cases.

2.1 Deterministic Optimization

Deterministic optimization is the classical branch of optimization algorithms in mathematics: it refers to algorithms where a rigid mathematical programming is followed and no random elements appear. Deterministic optimization algorithms are commonly based in the computation of the gradient and, in some cases, also the Hessian of the objective function of interest. Obviously, deterministic optimization has both advantages and drawbacks. An important advantage is that the convergence to a solution is much *faster* when compared to stochastic optimization algorithms; furthermore, being based on a rigorous mathematical formulation and not involving stochastic elements, the result of a deterministic optimization process is unequivocable and replicable. On the other hand, deterministic optimization algorithms reach steady points of the objective function; thus, the optimal solution found could be a *local optimum* and not the global optimum. Moreover, deterministic algorithms are intrinsically *single objective*.

This chapter deals with a description of the two main aspects of deterministic optimization, namely *unconstrained* and *constrained* optimization.

2.1.1 Introduction to Deterministic Optimization

In this paragraph is present an introduction of the deterministic optimization topic, as explained in [**5**].

2.1.1.1 Basic Notions

The present paragraph contains an overview of basic concepts useful in the optimization framework, as explained in [**6**].

First order derivatives of a real function

Each assigned vector $\boldsymbol{d} \in \mathcal{R}^n$ not null defines a *direction* in $\mathcal{R}^n$, that makes possible to define the following quantities:

- *directional derivative*: considering a generic function $f : \mathcal{R}^n \to \mathcal{R}$; f admits a directional derivative $Df(\boldsymbol{x}, \boldsymbol{d})$ in a point $\boldsymbol{x} \in \mathcal{R}^n$ along the direction $\boldsymbol{d} \in \mathcal{R}^n$ if exists the following finite quantity:

$$\lim_{t \to 0^+} \frac{f(\boldsymbol{x} + t\boldsymbol{d}) - f(\boldsymbol{x})}{\mathrm{t}} := Df(\boldsymbol{x}, \boldsymbol{d}) \qquad (2\text{-}1)$$

- *partial derivative*: considering a generic function $f : \mathcal{R}^n \to \mathcal{R}$; f admits a partial derivative $\partial f(\boldsymbol{x}) / \partial x_j$ in a point $\boldsymbol{x} \in \mathcal{R}^n$ in respect of the variable x_j if exists the following finite quantity:

$$\lim_{t\to 0} \frac{f(x_1,...,x_j+t,...,x_n)-f(x_1,...,x_j,...,x_n)}{\mathrm{t}} := \frac{\partial f(\boldsymbol{x})}{\partial x_j} \qquad (2\text{-}2)$$

- *gradient*: considering a generic function $f:\mathcal{R}^n \to \mathcal{R}$ and a point $x \in \mathcal{R}^n$. If exist the partial derivatives of f in $\boldsymbol{x}$ the gradient of f in $\boldsymbol{x}$ is the following vector:

$$\nabla f(\boldsymbol{x}) := \begin{pmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ ... \\ \frac{\partial f(\boldsymbol{x})}{\partial x_n} \end{pmatrix} \qquad (2\text{-}3)$$

- *differentiability*: considering a generic function $f:\mathcal{R}^n \to \mathcal{R}$; f is differentiable (by Frèchet, or in strong sense) in a point $\boldsymbol{x} \in \mathcal{R}^n$ if exists $g(\boldsymbol{x}) \in \mathcal{R}^n$ such that for each $\boldsymbol{d} \in \mathcal{R}^n$:

$$\lim_{\|\boldsymbol{d}\|\to 0} \frac{f(\boldsymbol{x}+\boldsymbol{d})-f(\boldsymbol{x})-g(\boldsymbol{x})^T\boldsymbol{d}}{\|\boldsymbol{d}\|} = 0 \qquad (2\text{-}4)$$

  The operator $g(\boldsymbol{x}):\mathcal{R}^n \to \mathcal{R}$ is called *derivative (of Frèchet)* of f in $\boldsymbol{x}$.

  Notice that the existence of $\nabla f(\boldsymbol{x})$ not implies, in general, the differentiability defined.

  It is demonstrated, nevertheless, that if $\nabla f(\boldsymbol{x})$ exists and it is *continuous* respect to $\boldsymbol{x}$, then f is differentiable in $\boldsymbol{x}$ and the Frèchet derivative of f in $\boldsymbol{x}$ coincides with $\nabla f(\boldsymbol{x})^T$.

  It is demonstrated that if f is differentiable in $\boldsymbol{x}$, the directional derivative of f along every direction $\boldsymbol{d} \in \mathcal{R}^n$ is the following:

$$\lim_{t\to 0^+} \frac{f(\boldsymbol{x}+t\boldsymbol{d})-f(\boldsymbol{x})}{\mathrm{t}} := \nabla f(\boldsymbol{x})^T\boldsymbol{d} \qquad (2\text{-}5)$$

Differentiability of a vector of functions

Considering $\boldsymbol{g}:\mathcal{R}^n \to \mathcal{R}^m$ a vector with *m* components of real functions. It is possible to define the following quantities:

- *Jacobian matrix*: considering a vector of functions $\boldsymbol{g}:\mathcal{R}^n \to \mathcal{R}^m$ and a point $\boldsymbol{x} \in \mathcal{R}^n$. If exist the partial derivatives $\partial g_i(\boldsymbol{x})/\partial x_j$, with i=1, …, m and j=1, …, n in $\boldsymbol{x}$. the Jacobian Matrix of $\boldsymbol{g}$ in $\boldsymbol{x}$ is the following:

$$\boldsymbol{J}(\boldsymbol{x}) := \begin{pmatrix} \frac{\partial g_1(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial g_1(\boldsymbol{x})}{\partial x_m} \\ \cdots & \cdots & \cdots \\ \frac{\partial g_m(\boldsymbol{x})}{\partial x_n} & \cdots & \frac{\partial g_m(\boldsymbol{x})}{\partial x_n} \end{pmatrix} \qquad (2\text{-}6)$$

- *derivative of a vector of functions*: considering a vector of functions $\boldsymbol{g}: \mathcal{R}^n \to \mathcal{R}^m$ and a point $\boldsymbol{x} \in \mathcal{R}^n$; $\boldsymbol{g}$ is differentiable (by Frèchet, or in strong sense) in a point $\boldsymbol{x} \in \mathcal{R}^n$ if exists a matrix $\boldsymbol{G}(\boldsymbol{x})$ such that for each $\boldsymbol{d} \in \mathcal{R}^n$:

$$\lim_{\|\boldsymbol{d}\| \to 0} \frac{\boldsymbol{g}(\boldsymbol{x}+\boldsymbol{d}) - \boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{G}(\boldsymbol{x})\boldsymbol{d}}{\|\boldsymbol{d}\|} = 0 \qquad (2\text{-}7)$$

  The operator $\boldsymbol{G}(\boldsymbol{x}): \mathcal{R}^n \to \mathcal{R}^m$ is called *derivative (of Frèchet)* of g in $\boldsymbol{x}$.

  It is demonstrated that if $\boldsymbol{J}(\boldsymbol{x})$ exists and it is *continuous* in respect of $\boldsymbol{x}$, then $\boldsymbol{g}$ is differentiable in $\boldsymbol{x}$ and the Frèchet derivative of $\boldsymbol{g}$ in $\boldsymbol{x}$ coincides with $\boldsymbol{J}(\boldsymbol{x})$.

Second order derivatives of a real function

Considering a real function $f: \mathcal{R}^n \to \mathcal{R}$. It is possible to define the following quantities:

- *Hessian matrix*: considering a real function $f: \mathcal{R}^n \to \mathcal{R}$ and a point $\boldsymbol{x} \in \mathcal{R}^n$. If exist the second partial derivatives $\partial^2 f(\boldsymbol{x}) / \partial x_i \partial x_j$, with i=1, …, n and j=1, …, n in $\boldsymbol{x}$. the Hessian Matrix of f in $\boldsymbol{x}$ is the following:

$$\boldsymbol{H}(\boldsymbol{x}) = \nabla^2 f(\boldsymbol{x}) := \begin{pmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^{\,2}} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n^{\,2}} \end{pmatrix} \qquad (2\text{-}8)$$

- *differentiability of second order:* considering a generic function $f: \mathcal{R}^n \to \mathcal{R}$ and a point $\boldsymbol{x} \in \mathcal{R}^n$. f is *twice* differentiable (by Frèchet, or in strong sense) in a point $\boldsymbol{x} \in \mathcal{R}^n$ if the first derivative, $\nabla f(\boldsymbol{x})^T$, is differentiable in $\boldsymbol{x}$. The first derivative of $\nabla f(\boldsymbol{x})^T$ is called *second derivative (of Frèchet)* of f.

  Notice that the existence of $\boldsymbol{H}(\boldsymbol{x})$ not implies, in general, the differentiability defined.

  It is demonstrated, nevertheless, that if $\boldsymbol{H}(\boldsymbol{x})$ exists and it is *continuous* in respect of $\boldsymbol{x}$, then f is twice differentiable in $\boldsymbol{x}$ and the second order Frèchet derivative of f in $\boldsymbol{x}$ coincides with $\nabla f^2(\boldsymbol{x})$.

Mean value Theorem and Taylor's formula:

In case of differentiable functions is possible give the following results:

- *theorem of mean*: considering a differentiable function $f:\mathcal{R}^n \to \mathcal{R}$; then for each $\boldsymbol{h} \in \mathcal{R}^n$:

$$f(\boldsymbol{x}+\boldsymbol{h}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{z})^T \boldsymbol{h} \qquad (2\text{-}9)$$

  where $\boldsymbol{z} \in \mathcal{R}^n$ is an appropriate point (dependent by $\boldsymbol{x}$ and $\boldsymbol{h}$) such that $\boldsymbol{z} = \boldsymbol{x} + \varsigma \boldsymbol{h}$ with $\varsigma \in (0,1)$.

- *Taylor theorem*: considering a twice differentiable function $f:\mathcal{R}^n \to \mathcal{R}$; then for each $\boldsymbol{h} \in \mathcal{R}^n$:

$$f(\boldsymbol{x}+\boldsymbol{h}) = f(\boldsymbol{x}) + \boldsymbol{h}^T \nabla f(\boldsymbol{x}) + \frac{1}{2}\boldsymbol{h}^T \nabla f^2(\boldsymbol{w})\boldsymbol{h} \qquad (2\text{-}10)$$

  where $\boldsymbol{w} \in \mathcal{R}^n$ is an appropriate point (dependent by $\boldsymbol{x}$ and $\boldsymbol{h}$) such that $\boldsymbol{w} = \boldsymbol{x} + \xi \boldsymbol{h}$ with $\xi \in (0,1)$.

Forms and quadratic functions

Given a square and symmetric matrix *A* of dimension (*n* x *n*), the *quadratic form* associated to the matrix *A* is the following function:

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j \qquad (2\text{-}11)$$

A *quadratic function* is a function of the following type:

$$q(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x} \qquad (2\text{-}12)$$

The quadratic form $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$ and the matrix $\boldsymbol{A}$ associated to the quadratic form are defined as:

- *positive definite*, if $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} > 0 \ \ \forall \boldsymbol{x} \in \mathcal{R}^n, \ \boldsymbol{x} \neq \boldsymbol{0}$;
- *positive semi-definite*, if $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \geq 0 \ \ \forall \boldsymbol{x} \in \mathcal{R}^n$;
- *indefinite*, if for some $\boldsymbol{x} \in \mathcal{R}^n$ results $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} > 0$ and for some $\boldsymbol{x} \in \mathcal{R}^n$ results $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} < 0$;
- *negative (semi-)definite*, if $-\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$ results *positive (semi-)definite*.

Convex Optimization

In the optimization framework there is interest in respect of the so called *convex* problems for reasons that will be explained in the present section. In this sense is important to give these definitions:

- *convex set*: a generic *set* $C \subseteq \mathcal{R}^n$ is *convex* if taken two points $\boldsymbol{y}, \boldsymbol{z} \in C$ results also that $[\boldsymbol{y}, \boldsymbol{z}] \subseteq C$, having denoted with $[\boldsymbol{y}, \boldsymbol{z}]$ the segment that joins the two points, segment given by $\boldsymbol{x}$ point obtained by:

$$\boldsymbol{x} = (1-\beta)\boldsymbol{y} + \beta\,\boldsymbol{z}, \quad \beta \in [\,0,1\,] \tag{2-13}$$

  It is verified that the intersection between a finite number of convex sets is a convex set too.

- *convex function*: a function $v(\boldsymbol{x})$ is *convex* on a convex set C if taken two points $\boldsymbol{y}, \boldsymbol{z} \in C$ results that:

$$v((1-\beta)\boldsymbol{y} + \beta\,\boldsymbol{z}) \le (1-\beta)v(\boldsymbol{y}) + \beta\,v(\boldsymbol{z}), \quad \beta \in [\,0,1\,] \tag{2-14}$$

- *strictly convex function*: a function $v(\boldsymbol{x})$ is *strictly* convex on a convex set C if taken two points $\boldsymbol{y}, \boldsymbol{z} \in C, \boldsymbol{y} \neq \boldsymbol{z}$ results that:

$$v((1-\beta)\boldsymbol{y} + \beta\,\boldsymbol{z}) < (1-\beta)v(\boldsymbol{y}) + \beta\,v(\boldsymbol{z}), \quad \beta \in [\,0,1\,] \tag{2-15}$$

  Geometrically a function is strictly convex if its graphic is always under every its secant.

- *concave function*: a function $v(\boldsymbol{x})$ is *concave* on a convex set C if the function $-v(\boldsymbol{x})$ is convex on C.

Definitions (2-14) and (2-15) allow to deduce important properties concerning first and second derivatives:

- *convex function*: a function $v(\boldsymbol{x})$ is *convex* on a convex set C if, and only if, for each $\boldsymbol{y}, \boldsymbol{z} \in C$ results that:

$$v(\boldsymbol{z}) \ge v(\boldsymbol{y}) + \nabla v(\boldsymbol{y})^T (\boldsymbol{z} - \boldsymbol{y}) \tag{2-16}$$

- *strictly convex function*: a function $v(\boldsymbol{x})$ is strictly convex on a convex set C if, and only if, for each $\boldsymbol{y}, \boldsymbol{z} \in C, \boldsymbol{y} \neq \boldsymbol{z}$ results that:

$$v(\boldsymbol{z}) > v(\boldsymbol{y}) + \nabla v(\boldsymbol{y})^T (\boldsymbol{z} - \boldsymbol{y}) \tag{2-17}$$

  Geometrically a function is strictly convex if its graphic is always up every its tangent.

- *convex function*: a function $v(\boldsymbol{x})$ is *convex* on a convex set C if, and only if, for each $\boldsymbol{x} \in C$ results that:

$$\frac{1}{2}\boldsymbol{y}^T\nabla^2 v(\boldsymbol{x})\boldsymbol{y} \geq 0, \quad \forall\, \boldsymbol{y} \in \mathcal{R}^n \qquad (2\text{-}18)$$

- *strictly convex function*: a function $v(\boldsymbol{x})$ is *strictly* convex on a convex set C if, and only if, for each $\boldsymbol{x} \in C$ results that:

$$\frac{1}{2}\boldsymbol{y}^T\nabla^2 v(\boldsymbol{x})\boldsymbol{y} > 0, \quad \forall\, \boldsymbol{y} \in \mathcal{R}^n \qquad (2\text{-}19)$$

- *quadratic convex function*: a function $q(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}$ is *convex* on a convex set C if, and only if, the matrix $\boldsymbol{Q}$ is positive semi-defined.
- *quadratic strictly convex function*: a function $q(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}$ is *strictly* convex on a convex set C if, and only if, the matrix $\boldsymbol{Q}$ is positive defined.

2.1.1.2 Problem Definition

Problem Description

Considering the problem of determine the value of a vector of *decision variables* $\boldsymbol{x} \in \mathcal{R}^n$ that minimize an *objective function* $f : \mathcal{R}^n \to \mathcal{R}$, when $\boldsymbol{x}$ is required to belong to a *feasible set* $\mathcal{F} \subseteq \mathcal{R}^n$; that is we consider the following problem:

$$\min_{\boldsymbol{x} \in \mathcal{F}} f(\boldsymbol{x}) \qquad (2\text{-}20)$$

Two cases are the main of interest:

- if the feasible set $\mathcal{F}$ is $\mathcal{R}^n$ the problem (2-20) is *unconstrained*:

$$\min_{\boldsymbol{x} \in \mathcal{R}^n} f(\boldsymbol{x}) \qquad (2\text{-}21)$$

- if the feasible set is described by *inequality* and/or *equality constraints* of the decision variables:

$$\mathcal{F} = \left\{ \boldsymbol{x} \in R^n : g_i(\boldsymbol{x}) \leq 0, i = 1,\ldots..,p; h_j(x) = 0, j = 1,\ldots..,m \right\} \qquad (2\text{-}22)$$

then the problem (2-20) becomes:

$$\min_{\boldsymbol{x}\in\mathcal{F}} f(\boldsymbol{x}) \quad \boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0} \quad \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} \qquad (2\text{-}23)$$

where:

- $\boldsymbol{h}: \mathcal{R}^n \to \mathcal{R}^m$ is the set of equality constraints;
- $\boldsymbol{g}: \mathcal{R}^n \to \mathcal{R}^p$ is the set of inequality constraints.

The problem (2-23) is a *Nonlinear Programming Problem* (*NLP*) when at least one of the problem functions f, g_i ,h_j is nonlinear in its argument $\boldsymbol{x}$.

In the following it will be assumed that the problem functions f, g_i ,h_j are at least *continuously differentiable* in R^n.

When f is a *convex* function and $\mathcal{F}$ is a *convex* set the problem (2-20) is a *convex NLP* problem.

Solution Existence

A point $\boldsymbol{x}^* \in \mathcal{F}$ is a *global solution* of the problem (2-20) if $f(\boldsymbol{x}^*) \le f(\boldsymbol{x})$, for all $\boldsymbol{x} \in \mathcal{F}$; it is a *strict* global solution of the problem if $f(\boldsymbol{x}^*) < f(\boldsymbol{x})$, for all $\boldsymbol{x} \in \mathcal{F}$, $\boldsymbol{x} \neq \boldsymbol{x}^*$. It is important distinguish these two cases:

- constrained problem: a main existence result is that a global solution exists if $\mathcal{F}$ is compact (Weirstrass Theorem);
- unconstrained problem: an existence result is that a global solution exists if is compact, for some finite α, the following *level set:*

$$\mathcal{L}_\alpha = \left\{ \boldsymbol{x} \in R^n : f(\boldsymbol{x}) \le \alpha \right\} \qquad (2\text{-}24)$$

A point $\boldsymbol{x}^* \in \mathcal{F}$ is *local solution* of the problem if there exists on open neighborhood $\mathcal{B}_{x^*}$ of $\boldsymbol{x}^*$ such that $f(\boldsymbol{x}^*) \le f(\boldsymbol{x})$, for all $\boldsymbol{x} \in \mathcal{F} \cap \mathcal{B}_{x^*}, \boldsymbol{x} \neq \boldsymbol{x}^*$; it is a *strict* local solution of the problem (2-20) if $f(\boldsymbol{x}^*) < f(\boldsymbol{x})$, for all $\boldsymbol{x} \in \mathcal{F} \cap \mathcal{B}_{x^*}, \boldsymbol{x} \neq \boldsymbol{x}^*$.

Solution Existence – Convex Optimization

Following definitions given in 2.1.1 is possible give the following definition concerning the existence solution in case of *convex* optimization problems:

- *(strictly) convex problem*: problem (2-20) is *convex* if the feasible set $\mathcal{F}$ is a convex set and the objective function f if convex on $\mathcal{F}$. If the objective function f if strictly convex on $\mathcal{F}$ the problem (2-20) is strictly convex.

Convexity introduces some properties that simplify the analysis and the solution of a convex problem:

- a convex optimization problem or has not solution or has only *global* solutions; it cannot have exclusively local solutions;
- in a strictly convex optimization problem the global solution, if exists, is *unique*;
- constrained optimization problem (2-23) is convex if the objective function f is a *convex* function in $\mathcal{R}^n$, inequality constraints are given by *convex* functions $g_i(\boldsymbol{x})$ in $\mathcal{R}^n$ and equality constraints are given by *affine* functions $a_j{}^T x - b_j$.

2.1.1.3 Optimality Conditions

In the present paragraph are illustrated the conditions concerning *local* optimum solution of problem (2-20).

Unconstrained problem

Local solution must satisfy the following *necessary optimality conditions* (*NOC*):

- *NOC-order I*: let $\boldsymbol{x}^*$ be a local solution of problem (2-21), then:

$$\nabla f(\boldsymbol{x}^*) = \boldsymbol{0} \qquad \text{(2-25)}$$

- *NOC-order II*: let $\boldsymbol{x}^*$ be a local solution of problem (2-21); if f is *twice continuously differentiable*, then:

$$\boldsymbol{y}^T \nabla^2 f(\boldsymbol{x}^*) \boldsymbol{y} \geq 0 \;\; \forall \, \boldsymbol{y} \in R^n \qquad \text{(2-26)}$$

  where y represents a generic *direction*.

If a point $\boldsymbol{x}^* \in \mathcal{F}$ satisfy the following *sufficient optimality conditions* (*SOC*), then it is a *local* solution of the problem (2-21):

$$\boldsymbol{y}^T \nabla^2 f(\boldsymbol{x}^*) \boldsymbol{y} > 0 \;\; \forall \, \boldsymbol{y} \in R^n, \boldsymbol{y} \neq \boldsymbol{0} \qquad \text{(2-27)}$$

The condition (2-27) implies that $\nabla^2 f(\boldsymbol{x})$ is *positive definite*; thus $\boldsymbol{x}^*$ is a *strict* local solution of problem (2-21).

Constrained problem

For constrained problem (2-22), most of *NOC* used assume that at a local solution the constraints satisfy some qualification condition to prevent the occurrence of degenerate cases. These conditions are the so called *constraints qualifications*.

Let $\hat{\boldsymbol{x}} \in \mathcal{F}$. An inequality constraint g_i is *active* at $\hat{\boldsymbol{x}}$ if $g_i(\hat{\boldsymbol{x}}) = 0$. The index set of inequality constraints active at $\hat{\boldsymbol{x}}$ is the following:

$$\mathcal{I}_a(\hat{\boldsymbol{x}}) = \left\{ i \in \{1, ..., p\} : g_i(\hat{\boldsymbol{x}}) = 0 \right\} \quad \text{(2-28)}$$

Obviously, an equality constraint h_j is active in $\hat{\boldsymbol{x}}$.

The simplest constraint qualification condition is the so called *linear independence* constraint qualification (*LICQ*). At $\hat{\boldsymbol{x}}$. *LICQ* is the following:

$$\nabla \mathcal{I}_a(\hat{\boldsymbol{x}}) \text{ and } \nabla \boldsymbol{h}(\hat{\boldsymbol{x}}) \rightarrow \text{linearly independent} \quad \text{(2-29)}$$

Under *LICQ*, the *NOC* of problem (2-22) are stated making use of the *Lagrangian function*:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{x}) \quad \text{(2-30)}$$

Where $\boldsymbol{\lambda} \in R^p$, $\boldsymbol{\mu} \in R^m$ are called *Lagrange Multipliers,* or *dual variables*.

Thus, local solution must satisfy the following *NOC*:

- *NOC-order I*, called *Karash-Kuhn-Tucker* (*KKT*) *NOC*: assume that $\boldsymbol{x}^*$ is a local solution of problem (2-22) and *LICQ* holds at x^*, then multipliers $\boldsymbol{\lambda}^* \geq 0, \boldsymbol{\mu}^*$ exist such that:

$$\nabla_x L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = 0, \ \boldsymbol{\lambda}^{*T} \boldsymbol{g}(\boldsymbol{x}^*) = \boldsymbol{0} \quad \text{(2-31)}$$

- *NOC-order II*: if *f, g, h* are *twice continuously differentiable*, then:

$$\boldsymbol{y}^T \nabla^2_x L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \boldsymbol{y} \geq 0, \ \forall \boldsymbol{y} \in \mathcal{N}(\boldsymbol{x}^*) \quad \text{(2-32)}$$

  where:

$$\mathcal{N}(\boldsymbol{x}^*) = \left\{ \boldsymbol{y} \in R^n : \nabla g_{Ia}(\boldsymbol{x}^*)^T \boldsymbol{y} = 0, \nabla \boldsymbol{h}(\boldsymbol{x}^*)^T \boldsymbol{y} = 0 \right\} \quad \text{(2-33)}$$

If a point $\boldsymbol{x}^* \in \mathcal{F}$ satisfy the following *SOC*, then it is a *local* solution of the problem (2-22): assuming that $\boldsymbol{x}^* \in \mathcal{F}$ and $\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$ satisfy the *KKT* condition. Assume further that:

$$\boldsymbol{y}^T \nabla^2_x L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \boldsymbol{y} > 0, \ \forall \boldsymbol{y} \in \mathcal{N}^+(\boldsymbol{x}^*), \boldsymbol{y} \neq \boldsymbol{0} \quad \text{(2-34)}$$

where:

$$\mathcal{N}^+(\boldsymbol{x}^*) = \left\{ \boldsymbol{y} \in R^n : \nabla \boldsymbol{h}(\boldsymbol{x}^*)^T \boldsymbol{y} = 0, \nabla g_i(\boldsymbol{x}^*)^T \boldsymbol{y} = 0, i \in \mathcal{I}_a(\boldsymbol{x}^*) \text{ with } \boldsymbol{\lambda}^*_i > 0 \right\} \quad (2\text{-}35)$$

Then $\boldsymbol{x}^*$ is a strict local solution of problem (2-22).

2.1.1.4 Performance of Algorithms

Convergence

Let $\Omega \in \mathcal{F}$ be the subset of points that satisfy the *NOC-order I* of problem (2-20).

Let $\left\{ \boldsymbol{x}^k \right\}, k = 0, 1, \ldots$ be the sequence of points produced by an algorithm. Then, the algorithm is *globally convergent* if a limit point $\boldsymbol{x}^*$ of $\left\{ \boldsymbol{x}^k \right\}$ exist such that $\boldsymbol{x}^* \in \Omega$ for any starting point $\boldsymbol{x}_0 \in \mathcal{R}^n$; it is *locally convergent* if the existence of the limit point $\boldsymbol{x}^* \in \Omega$ can be established only if the starting point $\boldsymbol{x}_0$ belong to some neighborhood of Ω.

The notion of convergence stated is the weakest that ensures that a point $\boldsymbol{x}^k$ arbitrarily close to Ω can be obtained for k large enough; in the unconstrained case this implies the following condition:

$$\lim_{k \to \infty} \inf \left\| \nabla f(\boldsymbol{x}^k) \right\| = 0 \quad (2\text{-}36)$$

Stronger convergence properties con also be established. For instance, any sequence of $\left\{ \boldsymbol{x}^k \right\}$ posses a limit point and any limit point of $\left\{ \boldsymbol{x}^k \right\}$ belongs to Ω; in the unconstrained case this implies the following condition:

$$\lim_{k \to \infty} \left\| \nabla f(\boldsymbol{x}^k) \right\| = 0 \quad (2\text{-}37)$$

Rate of Convergence

The most widely employed notion of rate of convergence is the *Q-rate of convergence*, that considers the ratio between two further iterates. Then, it is possible to define the following cases concerning the algorithms of common use:

- *Q-linear* rate of convergence:

$$\frac{\left\| \boldsymbol{x}^{k+1} - \boldsymbol{x}^* \right\|}{\left\| \boldsymbol{x}^k - \boldsymbol{x}^* \right\|} \leq r \quad (2\text{-}38)$$

  for *k* sufficient large and with $r \in [0, 1]$.

- *Q-superlinear* rate of convergence:

$$\lim_{k\to\infty} \frac{\left\| \boldsymbol{x}^{k+1} - \boldsymbol{x}^{*} \right\|}{\left\| \boldsymbol{x}^{k} - \boldsymbol{x}^{*} \right\|} \leq 0 \qquad (2\text{-}39)$$

- *Q-quadratic* rate of convergence:

$$\frac{\left\| \boldsymbol{x}^{k+1} - \boldsymbol{x}^{*} \right\|}{\left\| \boldsymbol{x}^{k} - \boldsymbol{x}^{*} \right\|^{2}} \leq R \qquad (2\text{-}40)$$

  where R is a positive constant, not necessary less than 1.

2.1.2 Introduction to Unconstrained Optimization

In the present paragraph are described algorithms for solving *unconstrained* nonlinear problems like (2-21).

It will be assume for simplicity that the problem function f is twice continuously differentiable in $\mathcal{R}^n$ even if not in all cases this condition is required and only once continuously differentiability is needed.

It will be also assumed that the existence condition (2-24) of global solution holds.

The algorithms treated in this paragraph generate a sequence $\left\{ \boldsymbol{x}^{k} \right\}$, starting from $\boldsymbol{x}_0$, by the following iteration:

$$\boldsymbol{x}^{\mathrm{k}+1} = \boldsymbol{x}^{k} + \alpha^{k}\, \boldsymbol{d}^{k} \qquad (2\text{-}41)$$

where:

- $\boldsymbol{d}^{k}$ is a *search direction*;
- α^{k} is a *step size* along $\boldsymbol{d}^{k}$.

The search direction affects on the local behavior of an algorithm and its rate of convergence, whereas global convergence often depends on the choice of the step size.

Exist two different approaches for the choice of the search direction and the step size:

- *line-search* approach;
- *trust region* approach.

2.1.2.1 Line-search Approach

Line-search algorithms keep *fixed* the search direction d^{k} and determine the step size α^{k} along that direction in order to ensure that the algorithm (2-41) results to be *globally convergent*. The main criteria generally used for that evaluation are the following:

- *exact search*: a first possibility is to set $\alpha^{k} = \alpha^{*}$ equal to the value that minimizes the function f along the direction $\boldsymbol{d}^{k}$:

$$\alpha^* = \arg\min_{\alpha} f\left(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k\right) \quad (2\text{-}42)$$

However an exact line search is usually quite computationally costly and its use is not diffused.

- *approximated search*: in case that $\nabla f(\boldsymbol{x})$ can be computed and assuming that $\boldsymbol{d}^k$ is a *descend direction* for f at $\boldsymbol{x}^k$:

$$\nabla f\left(\boldsymbol{x}^k\right)^T \boldsymbol{d}^k < 0 \quad (2\text{-}43)$$

it can be illustrated the simplest approximated line-search method, called the *Armijio's Method*, that find a step size α^k that satisfies a sufficient decrease of the objective function in the following steps:

  - data:

$$\delta \in (0,1), \gamma \in (0,1/2), c \in (0,1), \quad (2\text{-}44)$$

  - step1: chose an *initial* step Δ^k and set $\Delta^k = \alpha^k$;
  - step2: verification of the sufficient decrease condition: if

$$f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k) \le f(\boldsymbol{x}^k) + \gamma \alpha \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k \quad (2\text{-}45)$$

    then $\alpha^k = \alpha$ and stop;

  - step3: otherwise set $\alpha = \delta\alpha$ and go to step 2.

Other approximated methods exist but are not presented in this context.

2.1.2.2 Trust-region Approach

In trust region the iteration (2-41) becomes the following:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{s}^k \quad (2\text{-}46)$$

Where the step $\boldsymbol{s}^k$ is obtained by minimizing a quadratic model q^k of the objective function, not on whole space $\mathcal{R}^n$ but on a suitable *trust region* where the model is supposed to be reliable. The trust region is usually defined as a *Euclidean-norm* of the step. At each iteration *k* the step $\boldsymbol{s}^k$ is obtained by solving

$$\min_{\boldsymbol{s}\in R^n} q(\boldsymbol{s}) \rightarrow \min_{\boldsymbol{s}\in R^n} \frac{1}{2}\boldsymbol{s}^T \nabla^2 f(\boldsymbol{x}^k)\boldsymbol{s} + \nabla f(\boldsymbol{x}^k)^T \boldsymbol{s} + f(\boldsymbol{x}^k)$$
$$\| \boldsymbol{s} \|^2 \le (a^k)^2 \qquad (2\text{-}47)$$

where a^k is the trust region radius, iteratively updated. The idea is that when the approximation model q^k is a good approximation of the objective function, the radius a^k should be large in order to exploit the full step of the iteration method used (for example, the Newton's Method). The updating rule of a^k depends on the ratio between the *current reduction* $f(\boldsymbol{x}^k) - f(\boldsymbol{x}^{k+1})$ and the *expected reduction* $f(\boldsymbol{x}^k) - q^k(\boldsymbol{s}^k)$. A possible scheme of a trust region algorithm is the following:

- data:

$$0 < \gamma_1 \le \gamma_2 < 1, \quad 0 < \delta_1 < 1 \le \delta_2 \qquad (2\text{-}48)$$

- step1: choose $\boldsymbol{x}_0 \in \mathcal{R}^n$ and a radius $a_0 > 0$, set k=0.
- step2: find the step size:

$$\min_{\boldsymbol{s}\in R^n} q(\boldsymbol{s}) \rightarrow \min_{\boldsymbol{s}\in R^n} \frac{1}{2}\boldsymbol{s}^T \nabla^2 f(\boldsymbol{x}^k)\boldsymbol{s} + \nabla f(\boldsymbol{x}^k)^T \boldsymbol{s} + f(\boldsymbol{x}^k)$$
$$\| \boldsymbol{s} \|^2 \le (a^k)^2 \qquad (2\text{-}49)$$

- step3: if $f(\boldsymbol{x}^k) = q^k(\boldsymbol{s}^k)$ stop.
- step4: compute the ratio:

$$\rho^k = \frac{f(\boldsymbol{x}^k) - f(\boldsymbol{x}^k + \boldsymbol{s}^k)}{f(\boldsymbol{x}^k) - q^k(\boldsymbol{s}^k)} \qquad (2\text{-}50)$$

if $\rho^k \ge \gamma_1$ set $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{s}^k$, otherwise set $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k$.

- step 5: update the radius:

$$a^{k+1} = \begin{cases} \delta_1 a^k & \text{if } \rho^k < \gamma_1 \\ a^k & \text{if } \rho^k \in [\gamma_1, \gamma_2] \\ \delta_2 a^k & \text{if } \rho^k > \gamma_2 \end{cases} \qquad (2\text{-}51)$$

- set $k=k+1$ and go to step 2.

2.1.3 Methods for Unconstrained Optimization

In the following paragraph are described the main concepts concerning methods useful to solve unconstrained optimization problem, as explained in [**5**] and [**6**].

2.1.3.1 Gradient Method

The *gradient* method (called also *steepest descend* method) is based on the use of a search direction equal to the *anti-gradient* direction of the objective function f at point $\boldsymbol{x}$:

$$\boldsymbol{d}^k = -\nabla f\left(\boldsymbol{x}^k\right) \qquad (2\text{-}52)$$

Requiring only information about first order derivatives the method is very attractive because of its limit computational cost and storage requirements. Notice that from (2-52) results the following condition:

$$\nabla f\left(\boldsymbol{x}^k\right)^T \boldsymbol{d}^k = -\left\| \nabla f\left(\boldsymbol{x}^k\right)\right\|^2 \qquad (2\text{-}53)$$

Hence, if $\nabla f\left(\boldsymbol{x}^k\right) \neq \boldsymbol{0}$ the anti-gradient direction is *always a descend* direction.

Thus, the iteration *k* of the gradient method is the following:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha^k \nabla f\left(\boldsymbol{x}^k\right) \qquad (2\text{-}54)$$

By an appropriate choice of the step size α^k, following a line-search criteria explained in 2.1.2.1 is possible obtain the global convergence of method.

However, only a *linear* convergence rate can be reached.

2.1.3.2 Conjugate Gradient Method

The conjugate gradient method was originally introduced in order to solve minimization problems of strictly convex quadratic functions with the aim of accelerating the gradient method.

Given the following convex quadratic function:

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{Q}\,\boldsymbol{x} + a^T \boldsymbol{x} \qquad (2\text{-}55)$$

the principle on which the method is based is that the minimization of (2-55) on $\mathcal{R}^n$, with $\boldsymbol{Q}$ symmetric positive definite, can be split in *n* minimization over $\mathcal{R}$, along *n* directions $\boldsymbol{d}^0,\ldots,\boldsymbol{d}^{n-1}$ *conjugate* with respect the Hessian $\boldsymbol{Q}$, direction such that:

$$\boldsymbol{d}^j\ \boldsymbol{Q}\ \boldsymbol{d}^i = 0 \quad \text{for } i \neq j \qquad (2\text{-}56)$$

Along each direction an *exact* line search is performed.

The method described correspond to the so called *conjugate directions algorithm* (*CDA*): the algorithm finds the global minimizer of a strictly convex quadratic functions in at most *n* finite iterations.

In respect of the *CDA*, in which the conjugate directions are given, in the *conjugate gradient algorithm* (*CGA*) the *n* conjugate directions are generated iteratively according to the following rule:

$$\boldsymbol{d}^k = \begin{cases} -\nabla f(\boldsymbol{x}^k) & k = 0 \\ -\nabla f(\boldsymbol{x}^k) + \beta^{k-1}\boldsymbol{d}^{k-1} & k \geq 1 \end{cases} \qquad \text{(2-57)}$$

The scalar β^k is chosen in order to enforce the conjugacy among the directions; here is presented the *Fletcher-Reeves Formula*:

$$\beta_{FR}{}^k = \frac{\left\| \nabla f\left(\boldsymbol{x}^{k+1}\right)\right\|^2}{\left\| \nabla f\left(\boldsymbol{x}^{k}\right)\right\|^2} \qquad \text{(2-58)}$$

When f is not a quadratic function the choice of parameter β is different and a inexact line-search is performed to determine the step size α^k.

2.1.3.3 Newton's Method

The Newton's method is one of the most powerful algorithms for the solution of unconstrained optimization problems. It relies on the following quadratic approximation of the objective function in the neighborhood of the current iterate $\boldsymbol{x}^k$:

$$q^k(\boldsymbol{s}) = \frac{1}{2}\boldsymbol{s}^T\,\nabla^2 f(\boldsymbol{x}^k)\,\boldsymbol{s} + \nabla f(\boldsymbol{x}^k)^T\,\boldsymbol{s} + f(\boldsymbol{x}^k) \qquad \text{(2-59)}$$

The method requires the computation of $\nabla f(\boldsymbol{x})$ and $\nabla^2 f(\boldsymbol{x})$ at each iteration *k*.

Newton's direction $\boldsymbol{d}^k$ is obtained as a steady point of q^k, that is a solution of the following system:

$$\nabla^k q(\boldsymbol{s}) = 0 \;\rightarrow\; \nabla^2 f(\boldsymbol{x}^k)\,\boldsymbol{d}^k = -\nabla f(\boldsymbol{x}^k) \qquad \text{(2-60)}$$

provided that $\nabla^2 f\left(\boldsymbol{x}^k\right)$ is non-singular, Newton's direction is given by:

$$\boldsymbol{d}^k = -\left[\nabla^2 f(\boldsymbol{x}^k)\right]^{-1}\nabla f(\boldsymbol{x}^k) \qquad \text{(2-61)}$$

and the basic algorithmic scheme is defined by the following iteration:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k} - \left[\nabla^2 f(\boldsymbol{x}^{k})\right]^{-1} \nabla f(\boldsymbol{x}^{k}) \qquad (2\text{-}62)$$

If the starting point $\boldsymbol{x}^0$ is close enough to the solution $\boldsymbol{x}^*$, then the sequence generated converges *superlinearly* to the solution.

However, Newton's method presents some drawbacks: $\nabla^2 f\left(\boldsymbol{x}^{k}\right)$ may be singular, and hence Newton's direction cannot be defined; furthermore the starting point can be such that the iteration generated does not converge and even convergence to a maximum point can be occur. Therefore, Newton's method requires some modification that enforces the global convergence to a solution. In this sense, the application of line-search approach or trust region approach ensures this requirement.

2.1.3.4 Quasi-Newton Method

Quasi-Newton methods were introduced in order to obtain efficient methods that do not require the evaluation of second order derivatives.

They are obtained by setting the direction $\boldsymbol{d}^k$ as the solution of the following system:

$$\boldsymbol{B}^{k}\boldsymbol{d}^{k} = -\nabla f(\boldsymbol{x}^{k}) \qquad (2\text{-}63)$$

with $\boldsymbol{B}^k$ is a *n* x *n* symmetric and positive definite matrix which is adjusted iteratively in such a way that the direction $\boldsymbol{d}^k$ tends to approximate the Newton direction. Formula (2-63) is referred to a *direct* quasi-Newton formula, in turn the *inverse* quasi Newton formula is the following:

$$\boldsymbol{d}^{k} = -\boldsymbol{H}^{k}\nabla f(\boldsymbol{x}^{k}) \qquad (2\text{-}64)$$

where $\boldsymbol{H} = \boldsymbol{B}^{-1}$.

The idea at the basis of quasi-Newton method is to obtain the curvature information not from the Hessian but only from the values of the function and the gradient. The matrix $\boldsymbol{B}^{k+1}\left(\boldsymbol{H}^{k+1}\right)$ is obtained as a correction of $\boldsymbol{B}^{k}\left(\boldsymbol{H}^{k}\right)$, namely $\boldsymbol{B}^{k+1} = \boldsymbol{B}^{k} + \Delta\boldsymbol{B}^{k}\left(\boldsymbol{H}^{k+1} = \boldsymbol{H}^{k} + \Delta\boldsymbol{H}^{k}\right)$. The correction $\Delta\boldsymbol{B}^{k}\left(\Delta\boldsymbol{H}^{k}\right)$ is chosen such that:

$$\left(\boldsymbol{B}^{k} + \Delta\boldsymbol{B}^{k}\right)\boldsymbol{\delta}^{k} = \boldsymbol{\gamma}^{k}, \quad \boldsymbol{\delta}^{k} = \left(\boldsymbol{H}^{k} + \Delta\boldsymbol{H}^{k}\right)\boldsymbol{\gamma}^{k} \qquad (2\text{-}65)$$

where

$$\boldsymbol{\delta}^{k} = \boldsymbol{x}^{k+1} - \boldsymbol{x}^{k}, \quad \boldsymbol{\gamma}^{k} = \nabla f(\boldsymbol{x}^{k+1}) - \nabla f(\boldsymbol{x}^{k}) \qquad (2\text{-}66)$$

Exist different rules to update the matrix $\boldsymbol{H}^k$, here is presented the *Broyden-Fletcher-Goldfarb-Shanno* formula (*BFGS*) which is combined with a line-search approach:

$$\Delta \boldsymbol{H}^k = \frac{\boldsymbol{\delta}^k (\boldsymbol{\delta}^k)^T}{(\boldsymbol{\delta}^k)^T \gamma^k} - \frac{\boldsymbol{H}^k \boldsymbol{\gamma}^k (\boldsymbol{H}^k \boldsymbol{\gamma}^k)^T}{(\boldsymbol{\gamma}^k)^T \boldsymbol{H}^k \boldsymbol{\gamma}^k} + (\boldsymbol{\gamma}^k)^T \boldsymbol{H}^k (\boldsymbol{\gamma}^k) \frac{\boldsymbol{\delta}^k}{(\boldsymbol{\delta}^k)^T \boldsymbol{\gamma}^k} - \frac{\boldsymbol{H}^k \boldsymbol{\gamma}^k}{(\boldsymbol{\gamma}^k)^T \boldsymbol{H}^k \boldsymbol{\gamma}^k} \qquad (2\text{-}67)$$

As deals with convergences properties of quasi-Newton methods, satisfactory results exist for the convex case, while in the non- convex case only partial results exist.

2.1.3.5 Derivative-free Methods

Derivative free methods neither compute nor approximate derivatives of f. They are useful when ∇f is either unavailable, or unreliable (for example due to noise) or computationally too expensive.

Exist different kinds of methods. In this section will be described only the main features of the *direct search* methods, in order to provide a basic idea of the concept followed by their algorithms.

Direct search methods try to investigate the behavior of the objective function in the neighborhood of the current iterate by means of samples of the objective function along a set of directions by which span $\mathcal{R}^n$:

$$\mathcal{D}^k = \left\{ \boldsymbol{d}^1, ..., \boldsymbol{d}^r \right\} \text{ with } r \geq n+1 \qquad (2\text{-}68)$$

A possible choice of $\mathcal{D}^k$ is the following:

$$\mathcal{D}^k = \left\{ \boldsymbol{e}_j, j = 1, ..., n \right\} \qquad (2\text{-}69)$$

where $\boldsymbol{e}_j$ is the *j-th* column of the *Identity* matrix.

In case of *Pattern Search* algorithm at every iteration is built a pattern containing the set of candidates for the next iteration:

$$\mathcal{P}^k = \left\{ \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha^k \boldsymbol{d}^j, \text{with } \boldsymbol{d}^j \in \mathcal{D}^k \right\} \qquad (2\text{-}70)$$

Hence, the step $\boldsymbol{s}^k = \alpha^k \boldsymbol{d}^j$ is chosen so as to satisfy a simple reduction of the objective function:

$$f(\boldsymbol{x}^k + \alpha^k \boldsymbol{d}^j) = \min_{i \in \mathcal{D}^k} f(\boldsymbol{x}^k + \alpha^k \boldsymbol{d}^i) < f(\boldsymbol{x}^k) \qquad (2\text{-}71)$$

2.1.3.6 Simplex Method

The *Simplex Method* was firstly introduced by *Spendley* in 1962. A simplex is the *k*-dimensional analogue to a triangle or, in other words, a geometrical figure enclosed within *k+1* vertices in a *k*-dimensional space. The simplex is said *regular* if the length of the edges connecting the vertices is the same.

The Spendley simplex method starts from a set of *k+1* samples, locating a regular simplex in the design space. The vertice associate to the worst sample, with the highest value of objective function,

is then reflected in respect of the centroid of the other *k* vertices. If after this operation that vertice is again the worst, then is reflected the vertice with the second value of the objective function. After a certain number of function evaluations and reflections, the simplex is contracted in respect of the oldest vertices.

A series of modified and much more efficient methods have been proposed from 1962. Here is presented the *Nelder and Mead* simplex: they allow irregular simplexes and use different mechanism for moving the simplex. Denoting $\boldsymbol{x}^{k+1}$ the point to be reflected and $\boldsymbol{x}^{0}$ the centroid of other vertices:

- *Reflection*: reflection of the worst sample, then evaluation of the objective function at the reflection point:

$$\boldsymbol{x}^{r} = \boldsymbol{x}^{0} + \alpha\left(\boldsymbol{x}^{0} - \boldsymbol{x}^{k+1}\right) \tag{2-72}$$

- if after reflection the sample is still the worst, the simplex is contracted moving $\boldsymbol{x}^{k+1}$ to:

$$\boldsymbol{x}^{c} = \boldsymbol{x}^{n+1} + \rho\left(\boldsymbol{x}^{0} - \boldsymbol{x}^{k+1}\right) \tag{2-73}$$

- if after reflection the sample is the best so far, the reflected sample is pushed further along the direction $\boldsymbol{x}^{r} - \boldsymbol{x}^{k+1}$:

$$\boldsymbol{x}^{c} = \boldsymbol{x}^{0} + \gamma\left(\boldsymbol{x}^{0} - \boldsymbol{x}^{k+1}\right) \tag{2-74}$$

- if a certain point $\boldsymbol{x}^{1}$ is sufficiently old, the simplex is shrinked:

$$\boldsymbol{x}^{i} = \boldsymbol{x}^{1} + \sigma\left(\boldsymbol{x}^{i} - \boldsymbol{x}^{1}\right), \quad 1 = 2,\ldots,k+1 \tag{2-75}$$

$\alpha, \rho, \gamma, \sigma$ are respectively the reflection, contraction, expansion and shrinking coefficients. Typical values are the following:

$$\alpha = 1, \rho = \frac{1}{2}, \gamma = 2, \sigma = \frac{1}{2} \tag{2-76}$$

2.1.4 Introduction to Constrained Optimization

In the following paragraph are described the main concepts concerning methods useful to solve constrained optimization problems, as explained in [**4**], [**5**] and [**6**].

First of all, in order to provide a general overview of possible cases involved, it is presented a general classification of constrained optimization problems; each of which requires the application of different kind of algorithms:

- *linear programming*: the objective function and all the constraints are *linear* functions;
- *nonlinear programming*: both the objective function and some of the constraints are *nonlinear* or generally *smooth* functions;
- *linearly constrained optimization*: all the constraints are *linear* functions;
- *bound constrained optimization*: the only constraints of the problem impose lower and upper bounds on some design parameter;
- *convex programming*: the problem is a *convex* optimization problem.

The mathematic laying behind constrained optimization is very complex. In the following section it will provide just some basic idea on it and a discussion of the main common methods and algorithms concerning the fields of *linear* and *nonlinear programming*, branches to which belong two cases analyzed in the present work.

2.1.5 Methods for Constrained Optimization: Linear Programming

In this section are described the main concepts concerning methods useful to solve *linear programming* constrained optimization problems.

2.1.5.1 Duality

A *linear* programming optimization problem is a *convex* optimization problem too. The *NOC* explained in 2.1.1.3 allow deduce many important results concerning the theory of *linear programming*, results known like *Duality Theory*, treated in [**6**].

Referring for simplicity to the following linear *primal* problem:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{R}^n} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ & \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned} \tag{2-77}$$

where:

- $\boldsymbol{A}$ is a p x n matrix;
- $\boldsymbol{c}$ is a vector of dimension n;
- $\boldsymbol{b}$ is a vector of dimension p;

Considering the following linear *dual* problem:

$$\begin{aligned} \max_{\boldsymbol{u}\in\mathcal{R}^p} \quad & \boldsymbol{b}^T\boldsymbol{u} \\ & \boldsymbol{A}^T\boldsymbol{u} \leq \boldsymbol{c} \\ & \boldsymbol{u} \geq \boldsymbol{0} \end{aligned} \tag{2-78}$$

Due to the convex nature of the problems, for both the primal and dual problem the *KKT* conditions (2-31) are necessary and sufficient of *global* optimum.

It is demonstrated that the *KKT* conditions (2-31) are the same for both primal and dual problem: this leads to the result that the optimal solution of primal and dual problem is the same.

This is an important result that will be exploit in the first application treated in the present work.

2.1.5.2 Interior Point

This paragraph contains a description of the *Interior Point* method, as explained in [7]. Here is presented the method applied to a linear programming optimization problem but the method can be used to other cases, too.

Considering the following linear programming problem:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{R}^n} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ & \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned} \qquad (2\text{-}79)$$

where:

- $\boldsymbol{A}$ is a m x n matrix;
- $\boldsymbol{c}$ is a vector of dimension n;
- $\boldsymbol{b}$ is a vector of dimension m;

The Lagrangian function associated to the problem is the following:

$$L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{s}) = \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{y}^T(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}) - \boldsymbol{s}^T\boldsymbol{x} \qquad (2\text{-}80)$$

where $\boldsymbol{x}\in\mathcal{R}^n$ is the vector of parameters involved and $\boldsymbol{y}\in R^p, \boldsymbol{s}\in R^n$ are called *Lagrange Multipliers.*

The *KKT* conditions (2-31) of (2-79) are the following:

$$\begin{aligned} & \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c} \\ & \boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{0} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \boldsymbol{x}^T\boldsymbol{s} = \boldsymbol{0} \end{aligned} \qquad (2\text{-}81)$$

With some arrangement the *KKT* condition becomes the following:

$$\begin{aligned} & \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c} \\ & \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{0} \\ & x_i s_i = 0, \quad i = 1,\ldots,n \quad (\boldsymbol{XSe} = \boldsymbol{0}) \\ & (\boldsymbol{x},\boldsymbol{s}) \geq \boldsymbol{0} \end{aligned} \qquad (2\text{-}82)$$

where:

- $\boldsymbol{S} = diag(s_1,\ldots,s_n)$;
- $\boldsymbol{X} = diag(x_1,\ldots,x_n)$;
- $\boldsymbol{e} = (1,\ldots,1)^T$;

In reality, in the Interior Point method the *complementarity* relation $\boldsymbol{XSe}=\boldsymbol{0}$ is modified introducing a *perturbation* or *duality gap* parameter μ:

$$\boldsymbol{XSe}=\mu\boldsymbol{e} \qquad (2\text{-}83)$$

Furthermore is introduced a *centering parameter* $0<\sigma<1$:

$$\boldsymbol{XSe}=\sigma\mu\boldsymbol{e} \qquad (2\text{-}84)$$

Thus, the Interior Point system is resolved by the Newton's method with a line-search parameter α, chosen in order to satisfy the feasibility (from here the name on interior method).

In conclusion, Newton's method is applied to the following *perturbed* system:

$$\begin{aligned} \boldsymbol{A}^T\boldsymbol{y}+\boldsymbol{s}&=\boldsymbol{c}\\ \boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}&=0\\ \boldsymbol{XSe}&=\sigma\mu\boldsymbol{e}\\ (\boldsymbol{x},\boldsymbol{s})&\geq\boldsymbol{0} \end{aligned} \qquad (2\text{-}85)$$

Notice that when $\mu\to 0$ system (2-85) coincides with (2-82). It is demonstrate that when $k\to\infty$ the duality gap $\mu\to 0$.

2.1.6 Methods for Constrained Optimization: Non Linear Programming

In this section are described the main concepts concerning methods useful to solve *Non linear programming* constrained optimization problems (*NLP* problems).

As reported in [**5**], two main approaches have been developed:

- *unconstrained minimization methods*: the first approach is based on the transformation of the constrained problem in a sequence of *unconstrained* problems, or even in a single unconstrained problem. The main unconstrained minimization methods are the following:
  - quadratic penalty method;
  - logarithmic barrier method;
  - augmented Lagrangian method;
  - exact penalty method;
- *simpler constrained methods*: the second approach is based on the transformation of the constrained problem in a sequence of *simpler* constrained problems. The main simpler constrained methods are the following:
  - sequential quadratic programming;
  - elimination method;
  - active set method.

2.1.6.1 Quadratic Penalty Method

Considering the constrained optimization problem (2-23), and let $p(\boldsymbol{x}), p:\mathcal{R}^n\to\mathcal{R}$ be a continuous function such that $p(\boldsymbol{x})=0$ for all $\boldsymbol{x}\in\mathcal{F}$, $p(\boldsymbol{x})>0$ for all $\boldsymbol{x}\notin\mathcal{F}$ with $\mathcal{F}$ is the

feasible set. Then, it is possible to associate the constrained problem (2-23) to the following *unconstrained* problem:

$$\min_{\boldsymbol{x}\in R^n}\left[f(\boldsymbol{x})+\frac{1}{\varepsilon}p(\boldsymbol{x})\right] \qquad (2\text{-}86)$$

where $\varepsilon>0$. The function

$$P(\boldsymbol{x},\varepsilon)=f(\boldsymbol{x})+\frac{1}{\varepsilon}p(\boldsymbol{x}) \qquad (2\text{-}87)$$

parameterized by the *penalty parameter* ε is called a penalty function of problem (2-23) and it is obtained by adding to the objective function of the original problem a term that penalizes the constraint violation. Since the minimizer is usually exterior to the feasible set $\mathcal{F}$, the function to minimize is called *exterior penalty function*. The constraint violation is more severe as the penalty parameter ε is smaller. Given a sequence of positive numbers $\varepsilon>0, \left\{\varepsilon^{\mathrm{k}}\right\}, k=0,1,\ldots$ such that $\varepsilon^{k+1}<\varepsilon^{k}, \lim_{\mathrm{k}\to\infty}\varepsilon^{\mathrm{k}}=0$ the exterior penalty method developed for solving (2-23) is the following:

- data: $\left\{\varepsilon^{\mathrm{k}}\right\}$ such that $\varepsilon^{k+1}<\varepsilon^{k}, \lim_{\mathrm{k}\to\infty}\varepsilon^{\mathrm{k}}=0$.
- step1: choose $\boldsymbol{x}^{s}\in\mathcal{R}^{n}$ and set k=0.
- step2: starting from $\boldsymbol{x}^{s}$ minimize the following unconstrained problem:

$$\boldsymbol{x}^{k}=\arg\min_{x\in R^n}P(\boldsymbol{x},\varepsilon^{k}) \qquad (2\text{-}88)$$

- step3: if $\boldsymbol{x}^{k}$ is a KKT point, then stop.
- step4: set $\boldsymbol{x}^{\mathrm{s}}=\boldsymbol{x}^{k}, k=k+1$ and go to step 2.

It is demonstrated that limit point of the sequence $\left\{\varepsilon^{\mathrm{k}}\right\}$ are local solution of the constrained problem (2-23).

The quadratic penalty method suffers from the disadvantages that, as ε becomes small, the Hessian matrix $\nabla^2 P(\boldsymbol{x},\varepsilon)$ becomes ill-conditioned, so the method becomes difficult and slow.

2.1.6.2 Logarithmic Barrier Method

The logarithmic barrier method can be employed for the solution of *NLP* constrained optimization problems with only *inequality* constraints:

$$\begin{aligned}&\min_{\boldsymbol{x}\in\mathcal{R}^n} f(\boldsymbol{x})\\ &\boldsymbol{g}(\boldsymbol{x})\le\boldsymbol{0}\end{aligned} \qquad (2\text{-}89)$$

Furthermore it is assumed that the *strictly* feasible region

$$\mathcal{F}_0 = \left\{ \boldsymbol{x} \in R^n : g_i(\boldsymbol{x}) < 0, i = 1,....., p \right\} \quad \text{(2-90)}$$

is nonempty.

The logarithmic barrier method is built by adding a *barrier term* to the objective function of the original problem: let be $v(\boldsymbol{x})$ a continuous function defined in $\mathcal{F}_0$, such that $v(\boldsymbol{x}) \to \infty$ as $\boldsymbol{x} \in \mathcal{F}_0$ approaches to the boundary $\mathcal{F}$. Then, it is possible to associate the constrained problem (2-89) to the following *unconstrained* problem:

$$\min_{\boldsymbol{x} \in \mathcal{F}^0} \left[f(\boldsymbol{x}) + \varepsilon\, v(\boldsymbol{x}) \right] \quad \text{(2-91)}$$

where $\varepsilon > 0$. The function

$$V(\boldsymbol{x}, \varepsilon) = f(\boldsymbol{x}) + \varepsilon\, v(\boldsymbol{x}) \quad \text{(2-92)}$$

parameterized by ε is called a *barrier function* of problem (2-89). The barrier term prevents that a descent path for V starting in the interior of $\mathcal{F}$ crosses the boundary. The algorithms generated is called *interior point algorithm* because the sequence that are produced are interior to the strictly feasible region. The barrier is as sharper as the barrier parameter ε is larger.

Given a sequence of positive numbers $\varepsilon > 0$, $\left\{ \varepsilon^{\mathrm{k}} \right\}, k = 0, 1, \ldots$ such that $\varepsilon^{k+1} < \varepsilon^k$, $\lim_{\mathrm{k} \to \infty} \varepsilon^{\mathrm{k}} = 0$ the barrier function method developed for solving (2-89) is the following:

- data: $\left\{ \varepsilon^{\mathrm{k}} \right\}$ such that $\varepsilon^{k+1} < \varepsilon^k$, $\lim_{\mathrm{k} \to \infty} \varepsilon^{\mathrm{k}} = 0$.
- step1: choose $\boldsymbol{x}^s \in \mathcal{F}^0$ and set k=0.
- step2: starting from $\boldsymbol{x}^s$ minimize the following unconstrained problem:

$$\boldsymbol{x}^k = \arg \min_{\boldsymbol{x} \in \mathcal{F}^0} V(\boldsymbol{x}, \varepsilon^k) \quad \text{(2-93)}$$

- step3: if $\boldsymbol{x}^k$ is a KKT point, then stop.
- step4: set $\boldsymbol{x}^{\mathrm{s}} = \boldsymbol{x}^k, k = k + 1$ and go to step 2.

It is demonstrated that limit point of the sequence $\left\{ \varepsilon^{\mathrm{k}} \right\}$ are local solution of the constrained problem (2-89).

The most important barrier function used is the *logarithmic* barrier function and the corresponding method is called the logarithmic barrier method:

$$V(\boldsymbol{x}, \varepsilon) = f(\boldsymbol{x}) - \varepsilon \sum_{i=1}^{p} \log\left(-g_i(\boldsymbol{x})\right) \quad \text{(2-94)}$$

Like for the quadratic penalty method, also the logarithmic barrier method suffers from the disadvantages that, as ε becomes small, the Hessian matrix $\nabla^2 P(\boldsymbol{x}, \varepsilon)$ becomes ill-conditioned, so the method becomes difficult and slow.

2.1.6.3 Augmented Lagrangian Method

The augmented Lagrangian method can be employed for the solution of *NLP* constrained optimization problems (2-23).

Considering the following *NLP* constrained optimization problems with only *equality* constraints:

$$\begin{aligned} &\min_{\boldsymbol{x} \in \mathcal{R}^n} f(\boldsymbol{x}) \\ &\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} \end{aligned} \qquad (2\text{-}95)$$

The Lagrangian function of problem (2-95) is the following:

$$L(\boldsymbol{x}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\mu}\, \boldsymbol{h}(\boldsymbol{x}) \qquad (2\text{-}96)$$

By adding to L the quadratic penalty term for the constraint violation, the *augmented* Lagrangian function of problem (2-95) is the following:

$$L_a(\boldsymbol{x}, \boldsymbol{\mu}, \varepsilon) = L(\boldsymbol{x}, \boldsymbol{\mu}) + \frac{1}{\varepsilon} \| \boldsymbol{h}(\boldsymbol{x}) \|^2 \qquad (2\text{-}97)$$

It is demonstrated that to a local solution of problem (2-95) corresponds a local solution of the following problem:

$$\min_{\boldsymbol{x} \in \mathcal{R}^n} L_a(\boldsymbol{x}, \boldsymbol{\mu}^*, \varepsilon) \qquad (2\text{-}98)$$

where the parameter $\varepsilon \in (0, \varepsilon^*]$. Thus, the augmented Langrangian function can be employed in the quadratic penalty method without requiring that ε^k decreases to zero, mitigating the ill-conditioning when ε tends to zero. However, since $\boldsymbol{\mu}$ is not known in advance, an iterative procedure for its evaluation must be included in the method. The simplest update formula is the following:

$$\boldsymbol{\mu}^{k+1} = \boldsymbol{\mu}^k + \frac{2}{\varepsilon^k} \boldsymbol{h}(\boldsymbol{x}^k) \qquad (2\text{-}99)$$

Given a sequence of positive numbers $\varepsilon > 0$, $\{\varepsilon^k\}, k = 0, 1, \ldots$ the augmented Lagrangian method developed for solving (2-95) is the following:

- step1: choose $\boldsymbol{x}^0, \boldsymbol{\mu}^0$ and $\boldsymbol{x}^s \in \mathcal{R}^n$ and set k=0.
- step2: starting from $\boldsymbol{x}^s$ minimize the following unconstrained problem:

$$\boldsymbol{x}^k = \arg\min_{\boldsymbol{x}\in\mathcal{R}^n} L_a(\boldsymbol{x}, \boldsymbol{\mu}^k, \varepsilon^k) \qquad (2\text{-}100)$$

- step3: if $(\boldsymbol{x}^k, \boldsymbol{\mu}^k)$ is a KKT pair, then stop.
- step4: choose a penalty parameter $\varepsilon^{k+1} = (0, \varepsilon^k]$.
- step5: update the multiplier estimate $\boldsymbol{\mu}^{k+1}$.
- step6: set $\boldsymbol{x}^s = \boldsymbol{x}^k, k = k+1$ and go to step 2.

2.1.6.4 Exact Penalty Method

Exact penalty methods attempt to solve *NLP* problems by means of a *single* minimization of an *unconstrained function* rather than by means of a sequence of unconstrained minimizations. In this section is present a general description in order to explain the basic idea present behind the method.

Like for previous methods, these methods are based on the construction of a function depending on a penalty parameter $\varepsilon > 0$. It is possible to subdivide the exact penalty methods in two classes:

- methods based on *exact penalty functions*: the variables of the unconstrained problem are the same of the original constrained problem;
- methods based on *exact augmented Lagrangian functions*: in addition to the variables of the original constrained problem are introduced the Lagrange multipliers.

The function used possesses different *exactness* properties, depending on which kind of correspondence can be established between the solution of the original constrained problem and the unconstrained problem.

For example, continuously differentiable exact penalty function can be obtained from the augmented Lagrangian function described in 2.1.6.3 by substituting the multiplier vectors $\boldsymbol{\lambda}, \boldsymbol{\mu}$ by continuously differentiable multiplier functions $\boldsymbol{\lambda}(\boldsymbol{x}), \boldsymbol{\mu}(\boldsymbol{x})$ with the property that $\boldsymbol{\lambda}(\boldsymbol{x}^*) = \boldsymbol{\lambda}^*, \boldsymbol{\mu}(\boldsymbol{x}^*) = \boldsymbol{\mu}^*$ whenever the triplet $\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$ satisfies the *KKT* condition.

For the equality constrained problem (2-95) a multiplier function is the following:

$$\boldsymbol{\mu}(\boldsymbol{x}) = -\left[\nabla \boldsymbol{h}(\boldsymbol{x})^T \nabla \boldsymbol{h}(\boldsymbol{x})\right]^{-1} \nabla \boldsymbol{h}(\boldsymbol{x})^T \nabla f(\boldsymbol{x}) \qquad (2\text{-}101)$$

The associate *Fletcher's exact penalty function* is the following:

$$U_F(\boldsymbol{x}, \varepsilon) = f(\boldsymbol{x}) + \boldsymbol{\mu}(\boldsymbol{x})^T \boldsymbol{h}(\boldsymbol{x}) + \frac{1}{\varepsilon} \| \boldsymbol{h}(\boldsymbol{x}) \|^2 \qquad (2\text{-}102)$$

Other exact penalty functions exist for different cases; with respective advantages and drawbacks. Their detailed debate is not reported in the present work.

2.1.6.5 Sequential Quadratic Programming Method

The *sequential quadratic programming* approach (*SQP*), is a generalization to constrained optimization of Newton's method for unconstrained optimization.

For simplicity, considering the following equality constrained problem:

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathcal{R}^n} \; f(\boldsymbol{x}) \\ &\qquad \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} \end{aligned} \qquad \text{(2-103)}$$

The Lagrangian function of problem (2-103) is the following:

$$L(\boldsymbol{x},\boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\mu}^T \, \boldsymbol{h}(\boldsymbol{x}) \qquad \text{(2-104)}$$

As shown in (2-31), the *KKT* condition leads to a minimization of the Lagrangian function:

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathcal{R}^n} \; L(\boldsymbol{x},\boldsymbol{\mu}^*) \\ &\qquad \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} \end{aligned} \qquad \text{(2-105)}$$

where $\boldsymbol{\mu}^*$ is the *KKT* multiplier associated to the solution $\boldsymbol{x}^*$ of (2-107).

Considering the following quadratic approximation of the Lagrangian function:

$$\min_{\boldsymbol{s}\in R^n} L(\boldsymbol{x}^k,\boldsymbol{\mu}^k) \rightarrow \min_{s\in R^n} \frac{1}{2}\boldsymbol{s}^T \, \nabla_x{}^2 L(\boldsymbol{x}^k,\boldsymbol{\mu}^k)\,\boldsymbol{s} + \nabla_x L(\boldsymbol{x}^k,\boldsymbol{\mu}^k)^T \, \boldsymbol{s} + L(\boldsymbol{x}^k,\boldsymbol{\mu}^k) \qquad \text{(2-106)}$$

Then, in case of *linear* equality constraints, the associate *quadratic programming* problem of (2-105) is the following:

$$\begin{aligned} \min_{\boldsymbol{s}\in R^n} \quad & \frac{1}{2}\boldsymbol{s}^T \, \nabla_x{}^2 L(\boldsymbol{x}^k,\boldsymbol{\mu}^k)\,\boldsymbol{s} + \nabla_x f(\boldsymbol{x}^k)^T \, \boldsymbol{s} \\ & \nabla_x \boldsymbol{h}(\boldsymbol{x}^k)^T \, \boldsymbol{s} + \boldsymbol{h}(\boldsymbol{x}^k) = \boldsymbol{0} \end{aligned} \qquad \text{(2-107)}$$

The Newton iteration for the solution of (2-108) leads to the following system of equations :

$$\begin{pmatrix} \nabla^2 L(\boldsymbol{x}^k,\boldsymbol{\mu}^k) & \nabla \boldsymbol{h}(\boldsymbol{x}^k) \\ \nabla \boldsymbol{h}(\boldsymbol{x}^k)^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{s} \\ \boldsymbol{\eta} \end{pmatrix} = -\begin{pmatrix} \nabla f(\boldsymbol{x}^k) \\ \boldsymbol{h}(\boldsymbol{x}^k) \end{pmatrix} \qquad \text{(2-108)}$$

where $\boldsymbol{\eta}$ is a multiplier vector for the linear equality constraints of problem (2-107).

Employing an iteration based on the solution of problem (2-107) is obtained the sequential quadratic programming approach for the case of linear equality constraints. As in the Newton's method for unconstrained optimization problems, the evaluation of the Hessian matrix $\nabla_x{}^2 L\left(\boldsymbol{x}^k,\boldsymbol{\mu}^k\right)$ can be avoided with a Quasi-Newton approximation.

Different cases, not presented here, lead to other different system of equations.

2.1.6.6 Elimination Method

The *elimination method* is generally used for solving constrained optimization problems in which the objective function is a *quadratic* function. The basic idea on which the method is build is described in the present section.

Considering the following quadratic programming problem with k variables and l equality constraints:

$$\min_{\boldsymbol{x}\in R^n} \quad f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\,\boldsymbol{Q}\,\boldsymbol{x} + \boldsymbol{a}^T\,\boldsymbol{x} \qquad \boldsymbol{A}^T\,\boldsymbol{x} = \boldsymbol{b} \qquad \text{(2-109)}$$

where

- $\boldsymbol{Q}$ is positive or semi-positive definite k x k matrix;
- $\boldsymbol{x}, \boldsymbol{a}$ are k x 1 vectors;
- $\boldsymbol{b}$ is l x 1 vector;
- $\boldsymbol{A}$ is k x l matrix.

With *direct elimination method* are used the constraints to eliminate the variables, creating the following partition:

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix},\ \boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_{11}\,\boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21}\,\boldsymbol{Q}_{22} \end{pmatrix},\ \boldsymbol{a} = \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{pmatrix},\ \boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{pmatrix} \qquad \text{(2-110)}$$

where

- $\boldsymbol{x}_1, \boldsymbol{a}_1$ are l x 1 vectors;
- $\boldsymbol{x}_2, \boldsymbol{a}_2$ are (k - l) x 1 vectors;
- $\boldsymbol{A}_1, \boldsymbol{Q}_{11}$ are l x l matrix;
- $\boldsymbol{A}_2, \boldsymbol{Q}_{21}$ are (k - l) x l matrix;
- $\boldsymbol{Q}_{12}$ is l x (k - l) matrix;
- $\boldsymbol{Q}_{22}$ is (k - l) x (k - l) matrix.

Hence, the constraint can be written in the following form:

$$\boldsymbol{x}_1 = \boldsymbol{A}_1^{-T}\left(\boldsymbol{b} - \boldsymbol{A}_2^{-T}\boldsymbol{x}_2\right) \qquad \text{(2-111)}$$

Substituting (2-111) in $f(\boldsymbol{x})$ is obtained the following *unconstrained* optimization problem of a quadratic function:

$$\min_{\boldsymbol{x}_2\in R^n} \quad \psi(\boldsymbol{x}_2) \qquad \text{(2-112)}$$

Solving the problem is obtained the optimal $\boldsymbol{x}_2^*$, by substituting in (2-111) is obtained the optimal $\boldsymbol{x}_1^*$ and the Lagrangian multipliers are obtained by solving the following system of equations:

$$\nabla f(\boldsymbol{x}^*) = \boldsymbol{A}\boldsymbol{\lambda}^* \qquad (2\text{-}113)$$

Different cases concerning constraints involved, not presented here, lead to other different system of equations.

2.1.6.7 Active Set Method

Active set methods are methods for treating *inequality* constraints.

The most common is the *direct* active set method. The main phases on which the method is based are the following:

- The constraints included in the active set $\mathcal{A}$ are treated as equality constraints and the method iteratively adjusts this set;
- At iteration *k* a feasible point $\boldsymbol{x}^k$ satisfying the current active set $\mathcal{A}^k$ is known. The solution to the equality constraint problem in which only the active constraints occur is sought;
- Calling $\boldsymbol{\delta}^k$ the correction to $\boldsymbol{x}^k$:
  - in case $\boldsymbol{x}^k + \boldsymbol{\delta}^k$ is feasible respect the constraints not in $\mathcal{A}^k$, the next iteration is $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{\delta}^k$;
  - otherwise a line-search is performed along $\boldsymbol{\delta}^k$ to find the best feasible point;
  - if the search terminates at a point where an inactive constraints becomes active, $\boldsymbol{x}^{k+1} = x\boldsymbol{x}^k + \alpha^k \boldsymbol{\delta}^k; 0 < \alpha^k < 1$ is updated and the constraint is added to the active set;
  - if the solution to the equality constraints problem yields $\boldsymbol{\delta}^k = \boldsymbol{0}$, Lagrange multipliers must be computed to check whether an active inequality constraints has become inactive $(\alpha_i < 0)$; if this happens the constraint which has become inactive is removed from the active set;
  - if the solution to the equality constraints problem yields $\boldsymbol{\delta}^k = \boldsymbol{0}$ and no constraints to be removed from the active set is found, the optimization terminates and $\boldsymbol{x}^* = \boldsymbol{x}^k$ is the optimal solution.

2.2 Stochastic Optimization

The present paragraph is devoted to a description of the theory of *stochastic* optimization algorithms, as explained in detail in [**3**].

2.2.1 Introduction to Stochastic Optimization

Stochastic optimization framework is referred to optimization methods where *randomness* is present inside the criteria used to carry out the search.

In respect of deterministic optimization methods, stochastic optimization methods have the following features:

- are lower mathematically complicated;
- include randomness processes;
- are more slower towards the reaching of the optimal solution;

- are more *robust* being able to a better spanning of the design space and allowing to reach the *global* optimum solution and to overcome local minima;
- are useful to solve *multi-objective* optimization problems.

Important concepts commonly used in stochastic optimization methods are the following:

- *population*: stochastic optimization processes usually start from a set of samples in the design space. According to different rules this set evolves through several iterations. The set of samples is called *population*;
- *individual*: an individual is each sample of a population. The number of individuals composing the population is called *size* and is kept constant during the iterations. As a rule of thumb, the size should be at least 16 and possibly more than twice the input variables times the number of objectives.

2.2.2 Multi-Objective Optimization

An important feature of stochastic optimization methods deals with the possibility to solve *multi-objective* optimization problems. Also deterministic optimization algorithms are used to solve multi-objective optimization problems, considering a single objective function obtained by the average of the different objective functions involved in the problem. On the other hand, stochastic optimization methods are able to solve *true* multi-objective optimization problems, keeping the involved objective functions separated.

In this context must be defined the concept of *Pareto Optimality*. Considering a multi-objective optimization problem with l objective functions and let $\boldsymbol{f}(\boldsymbol{x})=(f_1(\boldsymbol{x}),...,f_l(\boldsymbol{x}))^T$ be the vector of values of the objective functions at the point $\boldsymbol{x}=(\boldsymbol{x}_1,...,\boldsymbol{x}_k)^T$ in the design space. Due to the conflict between objectives, there is not a single solution $\boldsymbol{x}^*$ that would be optimal for all objectives simultaneously: any objective vector can be better than others and solutions exist where none of the components of objective vector can be improved without deteriorating at least one of the other components. In this sense it is important to introduce the following definition: a point in the design space $\boldsymbol{x}^*$ is *Pareto Optimal* if the vector of the objective functions $\boldsymbol{f}(\boldsymbol{x}^*)$ is *non-dominated*. A vector $\boldsymbol{f}(\boldsymbol{x}_1)$ is said to dominate $\boldsymbol{f}(\boldsymbol{x}_2)$ if and only if $f_i(\boldsymbol{x}_1)\le f_i(\boldsymbol{x}_2)\forall\, i$, and at least a j exists for which $f_j(\boldsymbol{x}_1)< f_j(\boldsymbol{x}_2)$. The *Pareto Frontier* is the plot of the objective functions in the solution space whose vectors $\{\boldsymbol{f}(\boldsymbol{x})\}$ are non-dominated. The corresponding values of the input variables in the design space $\{\boldsymbol{x}\}$ form the set of the optimum solutions. Hence, the result of a multi-objective optimization problem is the set of the designs whose objective functions are non-dominated by any other design tested.

2.2.3 Methods for Stochastic Optimization

The main Stochastic optimization methods currently used in practical cases are the following:

- *simulated annealing*: this method simulates the *annealing* heat treatment process of steel;
- *particle swarm optimization*: this method emulates the social behavior of *birds flocking*;
- *game theory-based optimization*: this method simulates a *game* between different players, in which each player tries to reach his own objective;
- *evolutionary algorithm*: this method emulates the *evolution of the species* following the Darwin's theory;
- *genetic algorithm*: this method emulates the *evolution of the species* similarly to *EA*, of which they are considered a subcategory, due to the fact that they use different criteria for the emulation of the evolution of species.

2.2.3.1 Simulated Annealing

Simulated Annealing (*SA*) is an optimization technique that reproduces the annealing process exploited in metallurgy: the process consists in an initial heating of a material and a further controlled cooling in order to increase the size of crystals, reducing defects.

Annealing process starts from high temperature, at which the mobility of the atoms of material and their internal energy are high; further, by mean of a slow cooling, during which in first assumption the thermal equilibrium is kept, the material reaches a lower internal energy state than the initial one.

An *SA* optimization process starts from a *randomly* chosen point of the design space, at which is evaluated the objective function of interest $f(\boldsymbol{x}^1)$. Further a *decrease law* of temperature is imposed so that the temperature decreases to zero during the iterations: many laws exist, here is presented the following:

$$T^k = T^1 \left(1 - \frac{k-1}{k_{\max} - 1} \right)^p \qquad \text{(2-114)}$$

where:

- T^1 is the initial temperature;
- k is the current iteration;
- k_{MAX} is the maximum number of iteration, which is used as a stopping criteria of the process;
- $p \geq 1$ is the annealing coefficient.

Also a law relative to the *variation of parameters* involved is imposed. This is done based on the value of the current temperature, in order to keep the variation high at the start of process and low at the end, making a sort of final refinement of the solution. A possible law for the control of the variation of parameters imposed is the following:

$$x_i^{k+1} = x_i^k + \left[\left(x_i^{\max} - x_i^k \right) r_i^k - \left(x_i^k - x_i^{\min} \right) s_i^k \right] \frac{\mathrm{T}^k}{\mathrm{T}^1} \qquad \text{(2-115)}$$

where r_i and s_i are random numbers chosen in the range $[0,1]$.

In *SA* the objective is thought like the internal energy of the steel during the annealing process ad whose process aims to minimize. At each iteration, if the new objective function is better than the previous, the new configuration is accepted; otherwise, although its internal energy is higher, the new configuration has a certain probability of being accepted. A possible criteria for the solution selection is the following:

$$f(\boldsymbol{x}^{k+1}) \leq f(\boldsymbol{x}^k) \cdot \left(1 + t^k \frac{\mathrm{T}^k}{\mathrm{T}^1} \right) \qquad \text{(2-116)}$$

where t is a random number chosen in the range $[0,1]$.

Many variation of the algorithm exist, here has been presented only a base version of method in order to illustrate its logic.

In conclusion, the effectiveness of *SA* is due to the fact that when the temperature is high new samples are accepted even though they are not improving the performance of the system, allowing to overcome local minima.

2.2.3.2 Particle Swarm

Particle swarm optimization algorithms (*PSO*) emulate the social behavior of birds flocking during the search of food, that has made following the leader of the flock, the bird that has found where the food is.

In a *PSO* process each individual is a bird in the design space; at each iteration every bird shifts with a certain velocity in a direction that is function of the global location found so far by the swarm and the personal best location found so far by the bird. Method for avoiding collisions and for introducing craziness in the flock can be added, in order to overcome local minima.

Here is presented a basic idea of a *PSO* algorithm. The position x_i of individual *i* at iteration *k* is changed according to its own experience and that of its neighbours:

$$x_i^{\,k} = x_i^{\,k-1} + v_i^{\,k} \qquad \textbf{(2-117)}$$

where v_i is the velocity of individual *i*.

The velocity reflects the social exchanged information:

$$v_i^{\,k} = W\, v_i^{\,k-1} + C_1\, r_1 \left(\bar{x}_i - x_i^{\,k-1} \right) + C_2\, r_2 \left(\tilde{x}_i - x_i^{\,k-1} \right) \qquad \textbf{(2-118)}$$

where:

- $\bar{x}_i$ is the personal best location;
- $\tilde{x}_i$ is the global best location;
- C_1 is the *cognitive learning* factor, representing the attraction of the individual towards its own success;
- C_2 is the *social learning* factor, representing the attraction of the individual towards the success of its neighbours;
- W is the *inertia* factor;
- r_1 and r_2 are random numbers chosen in the range $[0,1]$.

2.2.3.3 Game Theory

Game theory (*GT*) is an optimization technique employed exclusively for multi-objective optimization problems. Each single objective function is assigned to a player which tries to minimize it. The input variables are subdivided between the players. At each turn of the game the player has at his disposal a number of iterations to be carried out in the design subspace on the input

variables assigned to him, trying to minimize his objective function. At the end an equilibrium is met between the objectives since the strategy of each player is affected by the other players.

A base *GT* algorithm is presented. Considering a minimization problem with two objective functions $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$. The input variables $\boldsymbol{x}_1$ are assigned to the first player and $\boldsymbol{x}_2$ are assigned to the second player.

In a *simultaneous competitive* game the player operate at the same time choosing their strategies, influencing the results obtained by the other player. The procedure is repeated through the turns of the game until the equilibrium is met. In this case, the equilibrium is called *Nash equilibrium*: $(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*)$ is a Nash equilibrium point if and only if:

$$\begin{cases} f_1(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*) = \inf_{x_1} f_1(\boldsymbol{x}_1, \boldsymbol{x}_2^*) \\ f_2(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*) = \inf_{x_2} f_2(\boldsymbol{x}_1^*, \boldsymbol{x}_2) \end{cases} \Leftrightarrow \begin{cases} \left.\dfrac{\partial f_1(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\partial \boldsymbol{x}_1}\right|_{\boldsymbol{x}1^*, \boldsymbol{x}2^*} = 0, \\ \left.\dfrac{\partial f_2(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\partial \boldsymbol{x}_2}\right|_{\boldsymbol{x}1^*, \boldsymbol{x}2^*} = 0 \end{cases} \quad (2\text{-}119)$$

that is, if each player, given the optimum solution found by the opponent, could not found any better arrangement for the input variables he controls.

2.2.3.4 Evolutionary Algorithms

Evolutionary algorithms (*EA*) are optimization algorithms based on the Darwin's theory relative to the evolution of species, by which the individuals of a population evolve creating better further populations. Each individual consists of a series of *real values* associated to the values of design variables involved. The main steps of an *EA* are the following:

- *initialization*: the initial population is created and evaluated;
- *mutation*: a *mutant individual* is created for each individual in the population;
- *cross-over*: the mutant individual is combined with its parent creating a *trial individual*;
- *evaluation*: the objective function of the trial individual is evaluated;
- *selection*: the best between the trial individual and its parent is selected to survive in the next generation.

Apart from the initialization, all steps are repeated until stopping criteria are met. *EA* are mainly based on the mutation operator.

Different approaches to *EA* exist: here is presented the approx of *differential evolution* (*DE*).

Calling x_i^k a real valued vector of the input variables of individual *i* and generation *k*, with *m* is the size of the population, in *DE* a mutant individual is represented by the following vector:

$$v_i^{k+1} = x_i^k + K\left(x_a^k - x_i^k\right) + F\left(x_b^k - x_c^k\right) \quad (2\text{-}120)$$

where:

- $a,b,c \in \{1,...,i-1,i+1,...,m\}$ are individuals *randomly* chosen, each one different from others;
- F is the *mutation constant*;
- K is the *scaling factor*.

Notice that each individual has the same probability to be chosen for the creation of the mutant individual.

Hence, the trial individual ${u_i}^{k+1}$ is created from the mutant individual and its parent:

$$ {u_{i,j}}^{k+1} = \begin{cases} {v_{i,j}}^{k+1} \text{ if } {r_{i,j}}^{k+1} \le \mathrm{C} \text{ or } j = {s_{i,j}}^{k+1} \\ \\ {x_{i,j}}^{k} \text{ if } {r_{i,j}}^{k+1} > \mathrm{C} \text{ or } j \ne {s_{i,j}}^{k+1} \end{cases} \quad i = 1,...,m; j = 1,...,n \qquad \text{(2-121)} $$

where:

- ${u_{i,j}}^{k+1}$ is the component j of the trial individual i at generation $k+1$;
- ${r_{i,j}}^{k+1}$ is a uniformly distributed random number in the range $[0,1]$;
- $C \in [0,1]$ is the *cross-over constant*;
- ${s_{i,j}}^{k+1}$ is the component j of vector ${s_i}^{k+1}$ which is a random permutation of the vector $[1,...,n]^T$;

Thus, a trial individual consists of some components of the mutant individual and at least one component of the parent vector.

The objective function of the trial individual is evaluated and compared with its parent. The better individual is selected to enter the next generation following the present criteria:

$$ {x_i}^{k+1} = \begin{cases} {u_i}^{k+1} \text{ if } f({u_i}^{k+1}) < f({x_i}^{k}) \\ {x_i}^{k} \text{ if } f({u_i}^{k+1}) \ge f({x_i}^{k}) \end{cases} \qquad \text{(2-122)} $$

Whenever the best individual does not change from a generation to the further, it could be displaced towards a better location in the design space through a steepest descend step.

2.2.3.5 Genetic Algorithms

Genetic algorithms (*GA*) are an evolutionary optimization technique principally based on *cross-over* operations and in which the input variables are encoded in binary strings.

In *GA* each individual consists of a string of *binary* data enconding the value of its input variables. The input variables are called *genes*, the set of genes unambiguously determining the individual in called *chromosome* or *DNA*, and the single bit of the string is called *allele*. We refer to the coding of the variables as *genotypes*; and to the variables themselves as *phenotypes*. Like in *EA*, the iteration is called *generation*, the individuals of a generations are chosen as *parents* for generating the new individuals, *offspring* or *children*, which will form the next generation.

In *GA* the design space has to be discretized, possibly in such a way that the number of value the variables can attain is an integer of power of 2, letting possible a binary representation of the input

variable. For instance, considering an optimization problem with three input variables $\boldsymbol{x} = [x_1, x_2, x_3]^T$ with each variable distributed in the range $[0,1]$. Let the range of x_1 be discretized in 2^2=4 nodes, x_2 in 2^3=8 nodes, x_3 in 2^4=16 nodes. The discretized design space allows $2^2 \cdot 2^3 \cdot 2^4 = 512$ possible solutions. Binary representations of the variables are now conceivable:

x_1		x_2		x_3			
binary	real	binary	real	binary	real	binary	real
00	0.000	000	0.000	0000	0.000	1000	0.533
01	0.333	001	0.143	0001	0.067	1001	0.600
10	0.667	010	0.286	0010	0.133	1010	0.667
11	1.000	011	0.429	0011	0.200	1011	0.733
		100	0.571	0100	0.267	1100	0.800
		101	0.714	0101	0.333	1101	0.867
		110	0.857	0110	0.400	1110	0.933
		111	1.000	0111	0.467	1111	1.000

Table 2-1: GA - Input Variables - Chromosome

Thus, the chromosome of the individual is a string of nine bits (or nine allele). For instance, an individual whose chromosome is 101100101 has genotypes 10, 110, 0101 and phenotypes x_1=0.667, x_2=0.857 and x_3=0.333.

The main steps of a *GA* are the following:

- initialization: initialization of a population of *m* individuals and evaluation of the objective function of each individual of the population;
- generation: at a generic generation *k* the following steps are repeated for the creation of a couple of offspring up to when *m* children have been generated:
  - selection of a pair of parents;
  - cross-over: application of the cross-over operator p_c , giving the birth of two children. The cross-over probability is in general quite high: $p_c = 0.9$;
  - mutation: application of the mutation operator to the two offspring with probability p_m . The mutation probability is in general low: $p_m = 0.01$ since it is applied to every allele and not to the whole individual;
- after the creation of a series of couple of offspring the new population is created, replacing completely the previous.

The selection of the parents is random but the probability to be chosen is not the same for all individuals: in fact, generally it depends on the value of the objective function of the single individual.

Different cross-over operators are applicable; the most common are the following:

- *one point* cross-over;
- *two point* cross-over;
- *uniformly* cross-over.

Figure 2-1 shows the mechanism of the three operators:

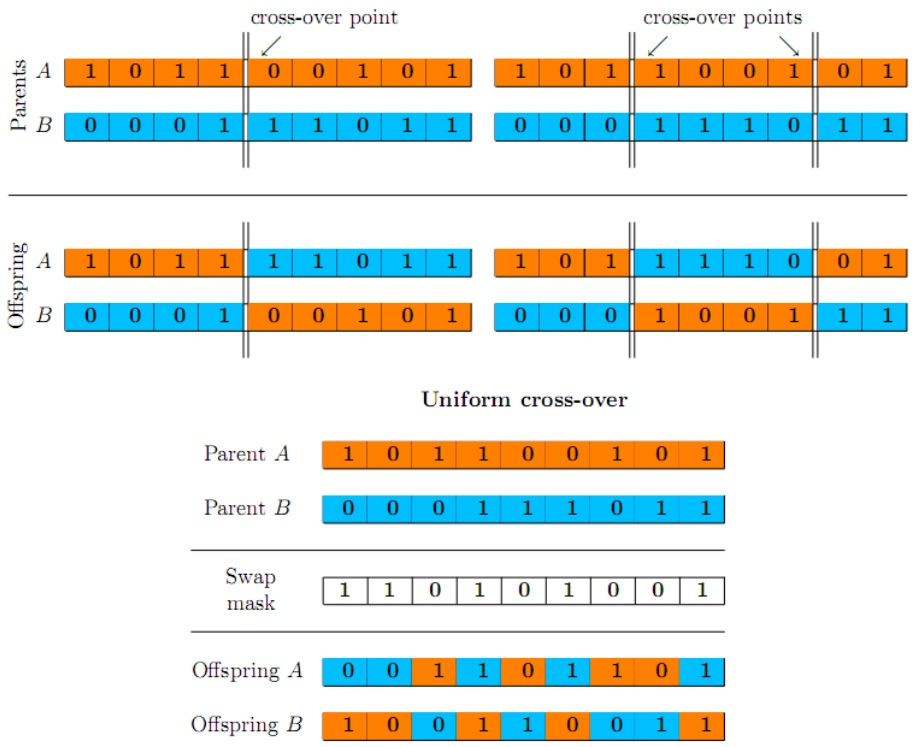

Figure 2-1: GA - Cross-Over Operators

Mutation operator acts simply swapping the allele on which it is applied. Figure 2-2 shows the effect of mechanism:

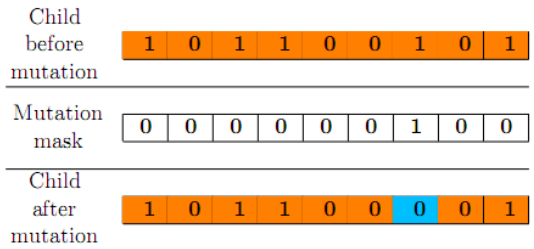

Figure 5.6: Mutation operator.

Figure 2-2: GA - Mutation Operator

Different operators exist, but they are not described in the present work.

3 Shape Optimization with Adjoint Method

The first part of the present work deals with the application of the *adjoint method* in the framework of aerodynamic shape optimization.

3.1 Overview of the Work

This activity represents a new direction of development within the research group operating at University of Bergamo on DG methods for Computational Fluid Dynamics (CFD).

The present work concerns the application of the *discrete* adjoint method, an efficient approach used to solve both shape optimization and grid-adaptivity problems. The method, useful to solve local constrained optimization problems, has been applied in the present context to solve a *shape* optimization problem.

The present chapter is structured as follows: 3.2 includes an overview of tools and methods generally employed in the design process and contains a description of optimization techniques currently used in the framework of aerodynamic shape optimization; 3.3 deals with a theoretical review of the discrete adjoint method; 3.4 describes the validation of the implementation of the discrete adjoint method; 3.5 concerns the application of the method to a shape optimization problem, while 3.6 includes conclusions and considerations about the activity.

3.2 Introduction

3.2.1 Aeronautical Design Process – Tools and Methods

The past 30 years have seen a revolution in the engineering design process: computational simulation has come to play an increasingly dominant role. As explained in [**8**], most important *CAD* methods have replaced the drawing boards as main tool for definition and control of design configuration. Now, software systems such as *CATIA* and *UNIGRAPHICS* provide solid modeling capabilities that enable designers to prepare and manage complex layouts, without the need to build prototypes.

Similarly, in the field of aeronautical design, structural analysis is now almost entirely carried out by computational methods. Commercially available software systems such as *NASTRAN* and *ANSYS* have been progressively developed and augmented with new features. Currently, engineers place complete confidence in their results and structural testing is limited to the role of verification that the design truly meets its specified requirements.

Computational simulation of fluid flow has not yet reached the same level of maturity. While commercial software for the simulation of fluid flow is offered by numerous vendors, aircraft companies continue to make investments in the in-house developments of their own methods. At the same time there are major ongoing efforts to develop the science of *CFD* in government research agencies, all of which are a source of industrially used computer programs. This reflects the fact that fluid is more complex and harder to predict than the behavior of structures.

Furthermore, in order to exploit computational simulation in the overall design process, currently are made many efforts to integrate *optimization techniques* in the design process. At the moment are numerous the commercial software, like *MODEFRONTIER* or *ISIGHT,* that allow to interface different tools, applying optimization techniques on design process. The number of simulations necessary to explore possible configurations is very high and the CPU time required for the entire process too.

For these reasons, currently there is a high interest in the scientific community towards an increasing of the efficiency of the optimization algorithms. In the recent period has became popular the *adjoint method,* a deterministic technique useful to solve efficiently local constrained

optimization problems. This method exploits the mathematics of the governing equations of the problem of interest, requiring a dedicate implementation, and allow the evaluation of gradient of interest independently from the number of parameters involved. The present work is focused on that method.

3.2.2 Aerodynamic Shape Optimization – Methods Overview

The present section deals with a general description of the use of optimization techniques in the framework of aerodynamic shape optimization. The reader should be aware that such description is far from being a complete review: it only mentions that has had a relevant impact on the present work.

Traditionally the process to select design variations has been carried out by trial and error, relying on the experience and intuition of the designer. In order to take full advantage of the possibility of examining a large design space, the numerical simulations have been combined with automatic search and optimization procedures. As explained in [**8**], this has lead to automatic design methods which will fully realize the potential improvements in aerodynamic efficiency.

3.2.2.1 Optimization by Genetic Algorithm

An approach that has become increasingly popular is to carry out a search over a large number of variations via a *genetic algorithm*. As explained in [**8**], this technique is generally used in a first phase of an optimization process and may allow the discovery of (sometimes unexpected) optimum design choices in very complex multi-objective problems. On the other hand, the technique becomes extremely expensive when each evaluation of the cost function requires intensive computation, as is the case in aerodynamic problems. Hence, the use of a genetic algorithm is possible only with a reduction of the number of design variables, which requires the use of *geometry parameterization techniques*.

3.2.2.2 Gradient-based Optimization

A different approach is represented by a *gradient-based* optimization technique. Initially developed in the field of structural engineering optimization problems, this technique employs the gradient of the objective function to drive the design in a direction of improvements and allows to reach a local minimum point. As explained in [**9**], due to the local nature of the method, it is usually employed in a final refinement phase of the optimization process.

A survey of approaches concerning the framework of gradient-based optimization is present in [**10**].

3.2.2.3 Gradient-based by Finite Difference

Initially the gradient has been evaluated by *finite differences* technique, using the flow solver like a *black-box*. As explained in [**9**], black-box gradient-based optimization methods appeared to be effective, nevertheless they also appeared to be extremely expensive, impractical for computing-intensive cases such as flow around a 3D wing. The computation of gradient was and is again currently the cause of the large cost: using finite difference technique it requires at least one additional evaluation for each design variable. In the case of N design variables, $N+1$ flow computations are required to compute the gradient. A rule of thumb says that an optimization problem converges in N steps, which means that the overall cost of the optimization problem scales with N^2. This explains why such an approach is impractical for 3D wings, which may requires hundreds or thousands of design variables. Also in this case, the use of parameterization techniques has been very important but didn't resolve completely the problem.

3.2.2.4 Gradient-based by Adjoint Method

Since the computation of gradient represented the main obstacle in using gradient-based optimization methods, researchers have tried to find ways of computing the gradient efficiently by the adjoint method.

As explained in [**9**], two main different approaches relative to the method exist: the *continuous* and the *discrete* adjoint method. In [**11**] is present a detailed description of both the *continuous* and *discrete* adjoint approach.

The Continuous Adjoint Method

Antony Jameson [**12**] proposed a method based on *control theory* to derive the sensitivities of flow equations with respect to shape parameters, for application of the compressible potential flow equations and the Euler equations. The sensitivity analysis employs dual variables which are solved by additional equations. The solution method is similar to the one used for the flow equations. The additional equations are called the *adjoint equations*.

Jameson and co-workers have developed the continuous (analytic) adjoint method, initially for the potential flow equation, then for the Euler equations and finally for the RANS equations. Airfoils as well as wing have been successfully optimized. The aerospace community has been impressed by the applications and several research groups have started working on the method.

In [**13**] and [**14**] is present a detailed description of the continuous adjoint solution for a quasi-1D Euler case.

In [**10**] there is a description of the continuous adjoint method applied to a 2D Euler case.

In [**15**] is present a description of the continuous adjoint method applied to a 3D viscous case.

Drawbacks of continuous adjoint method appeared. As explained in [**9**], the differentiation of the equations followed by discretization can lead to an inconsistency on the gradient obtained; furthermore boundary conditions for the dual variables, which have to be provided in order to solve the equations, are not very easy to define because the dual variables do not have an immediate physical interpretation.

The Discrete Adjoint Method

An alternative of the continuous adjoint method is the *discrete* adjoint method, obtained by performing the sensitivity analysis directly on the discrete code. As explained in [**9**], the method is relative straightforward to understand: only basic algebra is involved and the boundary conditions of the *dual variables* are not an issue because they unfold naturally. Moreover, the *direct differentiation* of the discrete flow equations, if performed exactly, implies a consistent gradient.

In [**14**] is present a description of the discrete adjoint solution for a quasi-1D Euler case.

In [**9**], [**10**], [**16**] and [**17**] there is a description of the use of discrete adjoint method for viscous shape optimization problems.

Drawbacks of the method are the storage requirements and the difficulty to differentiating large implementations. The storage requirements have to be carefully considered because several matrices appear in the derivation, but not of all need to be stored and some may be evaluated on-the-fly. Differentiating large implementations may be a very difficult activity: in order to simplify the derivation of the adjoint code a methodology known as *Automatic Differentiation* (AD) may be applied. AD uses basis linearization rules to manipulate source code and to create the sensitivity code.

Discrete versus Continuous Adjoint

The scientific community has investigated the difference between the continuous and the discrete adjoint method.

In [**14**] is present a comparison between the continuous and the discrete adjoint method for a quasi-1D Euler case.

[**10**] includes a comparison between the continuous and the discrete adjoint method for a 2D Euler case.

[**18**] concerns a comparison between the continuous and the discrete adjoint method for a 2D viscous case.

3.2.2.5 Shape Parameterization Techniques

As introduced in 3.2.2.1, the use of *geometry parameterization techniques* is necessary to reduce of the number of design variables involved in the optimization process. The use of the techniques is not a topic of the present work.

In [**9**] is present a list of parameterization techniques generally used in aerodynamic shape optimization problems.

3.3 Discrete Adjoint Approach in Aerodynamic Design

3.3.1 Sensitivity Analysis - Linearised Objective Function

The goal of an optimization problem is the minimization (or maximization) of a determined *objective function*. In the field of aerodynamic design optimization the objective function is in general a nonlinear function of a set of *discrete flow variables*. For example, the *lift* of an airfoil can be expressed like a function of discrete flow variables, solution of an approximation of flow equations:

$$L = L(\mathbf{U}) \qquad (3\text{-}1)$$

where

- L is the scalar function indicating the lift;
- $\mathbf{U}$ is the vector of flow variables, solution of flow equations.

In design optimization framework the main interest deals with the evaluation of the variation of the objective function due to a *perturbation of the geometry* and, hence, in the flow field. This evaluation is called *sensitivity analysis*. Focusing on the sensitivity term related to the flow perturbation, the reader will understand later the reason of this, the evaluation of the quantity is the following:

$$\boldsymbol{g}^T \boldsymbol{u} = \frac{\partial L}{\partial \mathbf{U}} \boldsymbol{u} \qquad (3\text{-}2)$$

where

- $\boldsymbol{g}^T = \frac{\partial L}{\partial \mathbf{U}}$ is the sensitivity of the objective function in respect of the flow variables;
- $\boldsymbol{u}$ is the perturbation in the flow field satisfying the flow equations.

Hence, the goal is the evaluation of the quantity $\boldsymbol{g}^T \boldsymbol{u}$.

3.3.2 Theory of Duality – Adjoint Variables

In this section is invoked the theorem of *duality*, mentioned in 2.1.5.1. , with the aim to understand the adjoint approach.

Primal Problem

Suppose that one wish to solve a *primal problem* and evaluate the following quantity:

$$g^T u \tag{3-3}$$

where $\boldsymbol{u}$ satisfies the following *linear system* of equations:

$$A u = f \tag{3-4}$$

where

- $\boldsymbol{A}$ is a generic matrix;
- $\boldsymbol{f}$ is a generic vector.

Dual Problem

The *dual form* of the primal problem is to evaluate the following quantity:

$$\psi^T f \tag{3-5}$$

where $\boldsymbol{\psi}$ is the *adjoint variable* and satisfies the following *linear system* of equations:

$$A^T \psi = g \tag{3-6}$$

It's interesting to note the used of the transpose matrix $\boldsymbol{A^T}$ and the interchange in the roles of vectors $\boldsymbol{f}$ and $\boldsymbol{g}$.

The equivalence of the two forms is the following:

$$\psi^T f = \psi^T A u = \left(A^T \psi\right)^T u = g^T u \tag{3-7}$$

Hence, solving the primal or the dual problem allows to obtain the same result. Furthermore, in case of single $\boldsymbol{f}$ and single $\boldsymbol{g}$ the use of primal or dual form requires the same computational effort, so there is no difference between two methods.

Indeed, suppose that we want to solve the problem for p different values of $\boldsymbol{f}$ and for m different values of $\boldsymbol{g}$. The choice would be to do *either* p different primal calculations or m different dual calculations. When the dimension of the system is very large the computational cost of the vector

dot product becomes negligible in respect of the cost for the solution of the linear system of equations, hence:

- the primal approach is much cheaper when $p << m$;
- the dual or adjoint approach is much cheaper when $m << p$.

It will see in the next paragraph that, in case of aerodynamics shape optimization problems, the dual form is much cheaper than the primal form.

3.3.3 Sensitivity Analysis: Direct Approach

Problem Description

In aerodynamics shape optimization framework the problem to be solved is the following:

$$\min J(\mathbf{U}, \alpha) \tag{3-8}$$

where

- J is a scalar objective function;
- $\mathbf{U}$ is the vector of flow variables, solution of flow equations, at discrete grid points;
- α is a set of design variables which controls the geometry to be designed.

This minimization is subject to the constraint that the discrete flow equations and boundary conditions are all satisfied:

$$\mathbf{N}(\mathbf{U}, \alpha) = 0 \tag{3-9}$$

where $\mathbf{N}$ is the set of discrete flow equations of problem.

Linearization

For a *single design variable* we can linearise the problem about a base solution $\mathbf{U_0}$:

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial \mathbf{U}} \frac{d\mathbf{U}}{d\alpha} + \frac{\partial J}{\partial \alpha} \tag{3-10}$$

where

- α is the single design variable;
- $\frac{dJ}{d\alpha}$ is the sensitivity of the objective function in respect of the design variable;
- $\frac{\partial J}{\partial \mathbf{U}}$ is the *partial* sensitivity of the objective function in respect of the flow variables;
- $\frac{\partial J}{\partial \alpha}$ is the *partial* sensitivity of the objective function in respect of the design variable;

- $\frac{d\mathbf{U}}{d\alpha}$ is the *flow sensitivity*;

Subject to the constraint that the *flow sensitivity* satisfies the linearized flow equations:

$$\frac{\partial \mathbf{N}}{\partial \mathbf{U}}\frac{d\mathbf{U}}{d\alpha}+\frac{\partial \mathbf{N}}{\partial \alpha}=0 \qquad (3\text{-}11)$$

where

- α is the single design variable;
- $\frac{\partial \mathbf{N}}{\partial \mathbf{U}}$ is the *partial* sensitivity of the flow equations in respect of the flow variables;
- $\frac{\partial \mathbf{N}}{\partial \alpha}$ is the *partial* sensitivity of the flow equations in respect of the design variable.

Defining:

$$\boldsymbol{g}^T=\frac{\partial J}{\partial \mathbf{U}};\ \boldsymbol{u}=\frac{d\mathbf{U}}{d\alpha};\ \boldsymbol{A}=\frac{\partial \mathbf{N}}{\partial \mathbf{U}};\ \boldsymbol{f}=-\frac{\partial \mathbf{N}}{\partial \alpha} \qquad (3\text{-}12)$$

problem becomes the following:

$$\frac{dJ}{d\alpha}=\boldsymbol{g}^T\boldsymbol{u}+\frac{\partial J}{\partial \alpha} \qquad (3\text{-}13)$$

subject to the constraint:

$$\boldsymbol{A}\boldsymbol{u}=\boldsymbol{f} \qquad (3\text{-}14)$$

3.3.4 Sensitivity Analysis: Adjoint Approach

The theory of duality applied to the *direct approach* leads to the following equivalence:

DIRECT APPROACH	ADJOINT APPROACH	
Problem	Problem	
$\frac{dJ}{d\alpha} = \boldsymbol{g}^T \boldsymbol{u} + \frac{\partial J}{\partial \alpha}$	$\frac{dJ}{d\alpha} = \boldsymbol{\psi}^T \boldsymbol{f} + \frac{\partial J}{\partial \alpha}$	**(3-15)**
Constraint	Constraint	
$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$	$\boldsymbol{A}^T \boldsymbol{\psi} = \boldsymbol{g}$	

Explicit each term the equivalence becomes the following:

DIRECT APPROACH	ADJOINT APPROACH	
Problem	Problem	
$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial \mathbf{U}} \frac{d\mathbf{U}}{d\alpha} + \frac{\partial J}{\partial \alpha}$	$\frac{dJ}{d\alpha} = \boldsymbol{\psi}^T \left(-\frac{\partial \mathbf{N}}{\partial \alpha} \right) + \frac{\partial J}{\partial \alpha}$	**(3-16)**
Constraint	Constraint	
$\frac{\partial \mathbf{N}}{\partial \mathbf{U}} \frac{d\mathbf{U}}{d\alpha} = -\frac{\partial \mathbf{N}}{\partial \alpha}$	$\left(\frac{\partial \mathbf{N}}{\partial \mathbf{U}} \right)^T \boldsymbol{\psi}^T = \left(\frac{\partial J}{\partial \mathbf{U}} \right)^T$	

It is evident that the linear system of the adjoint method doesn't depend by the design variable α: in case of a large number of shape parameters, and so a large number of design variables, the adjoint approach becomes much cheaper than the direct approach due to the fact that the linear system of equation needs to be evaluated only once, while in the direct approach is necessary to solve each linear system of equations for each design variable. Other mathematical operations are only vector dot products.

3.3.5 Sensitivity Analysis: Lagrangian Viewpoint

An alternative description of the adjoint approach arises using the terminology of *Lagrange Multipliers* associated with constrained minimization. In this framework, adjoint variables are Lagrange multipliers, written as λ, and are introduced in an augmented objective function to enforce the satisfaction of the discrete flow equations:

$$I(\mathbf{U}, \alpha) = J(\mathbf{U}, \alpha) - \boldsymbol{\lambda}^T \cdot \mathbf{N}(\mathbf{U}, \alpha) \qquad \textbf{(3-17)}$$

where

- I is the augmented objective function;
- λ is the vector of Lagrange multipliers.

Linearizing the augmented objective function:

$$dI = \left(\frac{\partial J}{\partial \mathbf{U}} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{N}}{\partial \mathbf{U}} \right) d\mathbf{U} + \left(\frac{\partial J}{\partial \alpha} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{N}}{\partial \alpha} \right) d\alpha \qquad (3\text{-}18)$$

where

- $d\boldsymbol{U}$ is a general perturbation of flow variables;
- $d\alpha$ is a general perturbation of design variable.

If λ is chosen to satisfy the adjoint equation:

$$\frac{\partial J}{\partial \mathbf{U}} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{N}}{\partial \mathbf{U}} = 0 \rightarrow \left(\frac{\partial \mathbf{N}}{\partial \mathbf{U}} \right)^T \boldsymbol{\lambda} = \left(\frac{\partial J}{\partial \mathbf{U}} \right)^T \qquad (3\text{-}19)$$

then

$$dI = \left(\frac{\partial J}{\partial \alpha} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{N}}{\partial \alpha} \right) d\alpha \qquad (3\text{-}20)$$

and thus dI / dα is obtained.

The final equation is the same that obtained from a duality point of view.

3.3.6 Nonlinear Optimization

As explained in the previous paragraphs, the sensitivity analysis computed by the direct or the adjoint approach, allows to evaluate the gradient of the objective function in respect of parameters representing the shape configuration.

After the evaluation of gradient dJ / dα , an iterative procedure can be used in order to reach the local minimum. As explained in chapter 5, different methods exist, like a *steepest descent* by *gradient method* or more fast methods, like *Newton* or *Quasi-Newton* method, each of one has own advantages and drawbacks.

Referring to the present work, an *inexact line search approach* with a *steepest descent* (gradient-based) algorithm has been used:

$$\Delta\alpha = -\Delta \frac{dJ}{d\alpha} \qquad (3\text{-}21)$$

where

- $\Delta\alpha$ is the single step variation of the design variable α;
- Δ is a parameter that controls the step size.

3.4 Discrete Adjoint Validation

This paragraph deals with the validation of the implementation of the discrete adjoint method. As explained, this activity represents a new direction of development within the research group operating at University of Bergamo on DG methods for Computational Fluid Dynamics (CFD). Starting from the simple quasi-1D Euler equations, the discrete adjoint method has been implemented in a research code based on the DG method.

Numerous are the scientific articles concerning the validation of the adjoint solution on the quasi-1D Euler case.

In [**13**] is present a description of the continuous (analytic) adjoint solution for a quasi-1D Euler case.

In [**14**] there is a comparison between the analytic and the discrete adjoint solution for a quasi-1D Euler case, using the *finite volume* method.

[**9**], [**10**], [**16**] and [**17**] concern the use of the discrete adjoint method for viscous shape optimization problems.

The validation of the implementation of the method has been carried out by comparing the values of computed adjoint variables with results of analytical solutions obtained by the author of [**13**].

3.4.1 Quasi-1D Steady Euler Equation

Equations

As explained in [**13**], Quasi-1D Steady Euler Equation for flow in a duct of cross-section *h(x)* is the following:

$$\mathbf{R}(\boldsymbol{u},h)=\frac{d}{dx}\left(\,h\mathbf{F}(\boldsymbol{u})\,\right)-\frac{dh}{dx}\left(\,\mathbf{P}(\boldsymbol{u})\,\right)=0$$

$$\boldsymbol{u}=\begin{pmatrix}\rho\\ \rho E\\ \rho q\end{pmatrix},\ \mathbf{F}(\boldsymbol{u})=\begin{pmatrix}\rho q\\ \rho q H\\ \rho q^2+p\end{pmatrix},\ \mathbf{P}(\boldsymbol{u})=\begin{pmatrix}0\\ 0\\ p\end{pmatrix} \qquad (3\text{-}22)$$

$$H=E+\frac{p}{\rho}=\frac{\gamma}{\gamma-1}\frac{p}{\rho}+\frac{q^2}{2}$$

where

- ***R(u,h)*** is the *residual vector*;
- *h* is the *cross-section* of duct;
- *ρ* is the *density*;
- *q* is the *velocity*,
- *p* is the *static pressure*;
- *E* is the *Total Energy per unit mass*;
- *H* is the *Total Enthalpy per unit mass.*

Note: the case is called *quasi-1D* because the behavior of quantities along the cross-section of duct is constant.

3.4.2 Discontinuous Galerkin Discretization

The governing equations have been discretized by the use of the DG method.

As explained in [**19**] concerning to *2D Euler Equations*, the phases of the DG discretization are the following.

Starting from analytic equations described in 3.4.1, by multiplying by a *test function*, integrating over the domain and performing and integration by parts has been obtained the weak statement of the problem:

$$-\int_{\Omega} \nabla \boldsymbol{v} \left(h\,\mathbf{F}(\mathbf{u}) \right) d\Omega + \int_{\partial\Omega} \boldsymbol{v} \left(h\,\mathbf{F}(\mathbf{u}) \right) \cdot \mathbf{n}\, d\partial\Omega - \int_{\Omega} \boldsymbol{v} \frac{dh}{dx} \left(\mathbf{P}(\mathbf{u}) \right) d\Omega = 0 \qquad \forall \boldsymbol{v} \qquad \text{(3-23)}$$

where

- **v** is the test function;
- Ω is the domain;
- $\partial\Omega$ denote the boundary of Ω.

A discrete form of the weak statement of the problem has been obtained by subdividing the entire domain Ω in a series of nonoverlapping elements $\{E\}$ and by considering functions $\mathbf{u}_h$ and $\mathbf{v}_h$, defined within each element, given by the combination of *n shape functions* ϕ_i :

$$\mathbf{u}_h(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{U}i\, \phi_i(\mathbf{x}), \quad \mathbf{v}_h(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{V}i\, \phi_i(\mathbf{x}) \quad \forall \mathbf{x} \in E \qquad \text{(3-24)}$$

where the expansion coefficients **U**i and **V**i are the *degrees of freedom* of the numerical solution and of the test function for a generic element *E*. Notice that there is no continuity requirement for $\mathbf{u}_h$ and $\mathbf{v}_h$ across element interface.

By splitting the integral over domain Ω in the sum of integrals over the elements *E* and by admitting only the functions $\mathbf{u}_h$ and $\mathbf{v}_h$ has been obtained the *semi-discrete form* of the problem relative to a generic element *E*:

$$-\int_{E} \nabla \boldsymbol{v}_h \left(h\,\mathbf{F}(\mathbf{u}_h) \right) d\Omega + \int_{\partial E} \boldsymbol{v}_h \left(h\,\mathbf{F}(\mathbf{u}_h) \right) \cdot \mathbf{n}\, d\partial\Omega - \int_{E} \boldsymbol{v}_h \frac{dh}{dx} \left(\mathbf{P}(\mathbf{u}_h) \right) d\Omega = 0 \qquad \forall \boldsymbol{v}_h \qquad \text{(3-25)}$$

where ∂E denote the boundary of element *E*.

The semi-discrete form must be satisfied for any element *E* and for any test function $\mathbf{v}_h$. Due to the fact that, within each element, the $\mathbf{v}_h$ are a linear combination of *n* shape function, the semi-discrete form is equivalent to the following system of equations:

$$-\int_{E} \nabla \phi_i \left(h\,\mathbf{F}(\mathbf{u}_h) \right) d\Omega + \int_{\partial E} \phi_i \left(h\,\mathbf{F}(\mathbf{u}_h) \right) \cdot \mathbf{n}\, d\partial\Omega - \int_{E} \phi_i \frac{dh}{dx} \left(\mathbf{P}(\mathbf{u}_h) \right) d\Omega = 0 \qquad 1 \le i \le n \qquad \text{(3-26)}$$

The system of *n* equations obtained is sufficient to determine the *n* degrees of freedom $\mathbf{U}_i$ of the unknown solution $\mathbf{u}_h$.

Due to the discontinuity of approximated solution, flux terms are not uniquely determined at the element interfaces. Following techniques usually used in *finite volume schemes* the *flux function* appearing in the second term is replaced by a *numerical flux function*:

$$\mathbf{F}(\mathbf{u}_h)\cdot\mathbf{n} = \mathbf{h}\left(\mathbf{u}_h^{-},\mathbf{u}_h^{+}\right) \qquad (3\text{-}27)$$

where

- **h** is the numerical flux function;
- $\mathbf{u}_h^{-}$ is the internal interface state;
- $\mathbf{u}_h^{+}$ is the interface state on the neighboring element;
- **n** is normal direction to the interface.

In order to guarantee the formal accuracy of the scheme the numerical flux must be satisfy the *consistency condition* and the *directional consistency condition*:

$$\mathbf{h}(\mathbf{u},\mathbf{u}) = \mathbf{F}(\mathbf{u})\cdot\mathbf{n} \quad , \quad \mathbf{h}(\mathbf{u},\mathbf{v}) = -\mathbf{h}(\mathbf{u},\mathbf{v}) \qquad (3\text{-}28)$$

Several numerical fluxes that satisfy the above criteria exist: in the present work has been used the *Godunov scheme*.

About the geometry treatment, as common practice in the finite element method has been introduced a *reference element* in a non-dimensional space and a *geometric transformation* which maps the reference element onto the real element in the physical space. The mapping from the reference element to the real element is a polynomial function of order less than or equal to *m* defined in the space of the reference element for each independent variable. In order to guarantee the geometric continuity between neighboring elements the mapping has been expressed in terms of polynomial functions and coefficients, function of *nodal coordinates* on physical space:

$$\mathbf{x}_h(\xi) = \sum_{i=0}^{m} \mathbf{a}_i\left(\mathbf{x}_{h\_N}\right)\hat{P}_i^{\,m}(\xi) \quad \forall\xi\in\hat{E} \qquad (3\text{-}29)$$

where

- $\hat{E}$ is the reference element;
- ξ is the independent coordinate on the reference element;
- $\mathbf{x}_{h\_N}$ is the nodal coordinate on physical space;
- $\mathbf{a}_i$ are the coefficients function of nodal coordinates;
- $\hat{P}_i^{\,m}$ are polynomials of degree less than or equal to *m* within the reference element.

Also functions $\mathbf{u}_h$ and ϕ_i are defined as polynomial functions on the reference element:

$$\hat{\mathbf{u}}_h(\xi) = \sum_{i=1}^{n} \mathbf{U}i\,\hat{\phi}_i^{\,k}(\xi) \quad \forall\xi\in\hat{E} \qquad (3\text{-}30)$$

where

- $\hat{\mathbf{u}}_h$ is function $\mathbf{u}_h$ evaluated on the reference element;

- $\hat{\phi}_i^{\,k}$ are polynomials of degree less than or equal to *k* within the reference element.

The function of $\mathbf{u}_h$ and ϕ_i for a generic *real element E* has been formally obtained as:

$$\mathbf{u}_h(\mathbf{x}_h) = \hat{\mathbf{u}}_h[\,\xi(\mathbf{x}_h)\,], \quad \phi^k(\mathbf{x}_h) = \hat{\phi}^{\,k}\,[\,\xi(\mathbf{x}_h)\,] \quad \forall \mathbf{x}_h \in E \qquad \text{(3-31)}$$

where $\xi = \xi(\mathbf{x}_h)$ is the inverse of the mapping.

Effectively, the integrals have been evaluated in the space of the reference element by means of numerical quadrature formulae. As a consequence the quantities have been evaluated directly solving integrals on the reference element and computing the determinant of the inverse Jacobian matrix of the transformation at each quadrature point.

The number of quadrature point used has been chosen in order to integrate exactly on the reference element polynomials of order *2k*.

Hence, by assembling together all the element contribution the system of ordinary differential equations which govern the problem has been written as

$$\mathbf{R}(\mathbf{U}) = \mathbf{0} \qquad \text{(3-32)}$$

where **R** is the *residual vector*.

The solution of the system of equations has been obtained by an *implicit backward Euler scheme*, as explained in [**9**], which can be written as:

$$\left[\frac{\mathbf{M}h}{\Delta t} + \frac{\partial \mathbf{R}(\mathbf{U}^n)}{\partial \mathbf{U}}\right] \Delta\mathbf{U} = -\mathbf{R}(\mathbf{U}^n) \qquad \text{(3-33)}$$

where

- **M** is the *Mass Matrix*;
- Δt is the time increment;
- $\Delta\mathbf{U} = \mathbf{U}^{n+1} - \mathbf{U}^n$ is the variation of solution of current instant $\mathbf{U}^{n+1}$ and previous instant $\mathbf{U}^n$;
- $\frac{\partial \mathbf{R}(\mathbf{U}^n)}{\partial \mathbf{U}}$ is the *exact Jacobian* of residual in respect of the solution, evaluated at previous instant.

Imposing the initial conditions and obtaining the following condition:

$$\mathbf{R}(\mathbf{U}) = \mathbf{0} \qquad \text{(3-34)}$$

the steady value of flow field variables has been determined.

3.4.3 Discrete Adjoint System

As for flow variables, a discretization of the adjoint variable has been carried out, considering the function $\boldsymbol{\psi}_h$ given by the combination of *n shape functions* ϕ_i:

$$\boldsymbol{\psi}_h(\mathbf{x}) = \sum_{i=1}^{n} \boldsymbol{\psi} i \, \phi_i(\mathbf{x}) \qquad \forall \mathbf{x} \in E \tag{3-35}$$

where the expansion coefficients $\boldsymbol{\psi}$i are the *degrees of freedom* of the adjoint numerical solution.

Obviously, considerations about the geometry treatment on the reference element explained for the flow field hold for the adjoint system too.

As explained in 3.3.5, the discrete adjoint system to solve is the following:

$$\left(\frac{\partial \mathbf{R}(\mathbf{U})}{\partial \mathbf{U}}\right)^T \boldsymbol{\psi} = \left(\frac{\partial J(\mathbf{U})}{\partial \mathbf{U}}\right)^T \tag{3-36}$$

As explained in [**20**] and [**9**], in the present work the solution of the discrete adjoint system has been obtained using an iterative strategy, in order to introduce minor changes in the original routines for the solution of the flow filed: the strategy consists on employing the same implicit scheme as the one used for the flow equations, treating the sensitivity equations as non-linear equations. The *incremental formulation* used is the following:

$$\left[\frac{\mathbf{M}h}{\Delta t} + \frac{\partial \mathbf{R}(\mathbf{U})}{\partial \mathbf{U}}\right]^{\mathrm{T}} \Delta\boldsymbol{\psi} = \left[\frac{\partial \mathbf{R}(\mathbf{U})}{\partial \mathbf{U}}\right]^T \boldsymbol{\psi}^n - \left[\frac{\partial J(\mathbf{U})}{\partial \mathbf{U}}\right]^T \tag{3-37}$$

where $\Delta\boldsymbol{\psi} = \boldsymbol{\psi}^{n+1} - \boldsymbol{\psi}^n$ is the variation of solution of current instant $\boldsymbol{\psi}^{n+1}$ and previous instant $\boldsymbol{\psi}^n$;

Defining the *adjoint residual vector*:

$$\mathbf{R}(\boldsymbol{\psi}^n) = \left[\frac{\partial \mathbf{R}(\mathbf{U})}{\partial \mathbf{U}}\right]^T \boldsymbol{\psi}^n - \left[\frac{\partial J(\mathbf{U})}{\partial \mathbf{U}}\right]^T \tag{3-38}$$

the system of equation solved is the following:

$$\left[\frac{\mathbf{M}h}{\Delta t} + \frac{\partial \mathbf{R}(\mathbf{U})}{\partial \mathbf{U}}\right]^{\mathrm{T}} \Delta\boldsymbol{\psi} = \mathbf{R}(\boldsymbol{\psi}^n) \tag{3-39}$$

Notice the equivalence with the system of equations solved for the resolution of the flow field.

We are interested to the evaluation of the *steady* adjoint variables; hence, during the time marching the quantities relative to the flow filed are kept fixed on their steady value $\mathbf{U}^{st}$:

$$\left[\frac{\mathbf{M}h}{\Delta t} + \frac{\partial \mathbf{R}(\mathbf{U}^{st})}{\partial \mathbf{U}}\right]^{\mathrm{T}} \Delta\boldsymbol{\psi} = \left[\frac{\partial \mathbf{R}(\mathbf{U}^{st})}{\partial \mathbf{U}}\right]^T \boldsymbol{\psi}^n - \left[\frac{\partial J(\mathbf{U}^{st})}{\partial \mathbf{U}}\right]^T \tag{3-40}$$

Imposing the initial condition and obtaining the following condition:

$$R(\psi)=0 \qquad (3\text{-}41)$$

the steady value of adjoint variables has been determined.

3.4.4 Adjoint Validation

The validation of the implementation of the method has been carried out by comparing the values of computed adjoint variables with results of analytical solutions obtained by the author of [**13**].

3.4.4.1 Quasi-1D Case Description

Geometry

The geometry of the duct of the present case is the following:

$$h(x)=\begin{cases} 2 & \left(-1\le x\le -\frac{1}{2}\right) \\ 1+\sin^2(\pi x) & \left(-\frac{1}{2}\le x\le \frac{1}{2}\right) \\ 2 & \left(\frac{1}{2}\le x\le 1\right) \end{cases} \qquad (3\text{-}42)$$

Figure 3-1 shows the geometry of duct:

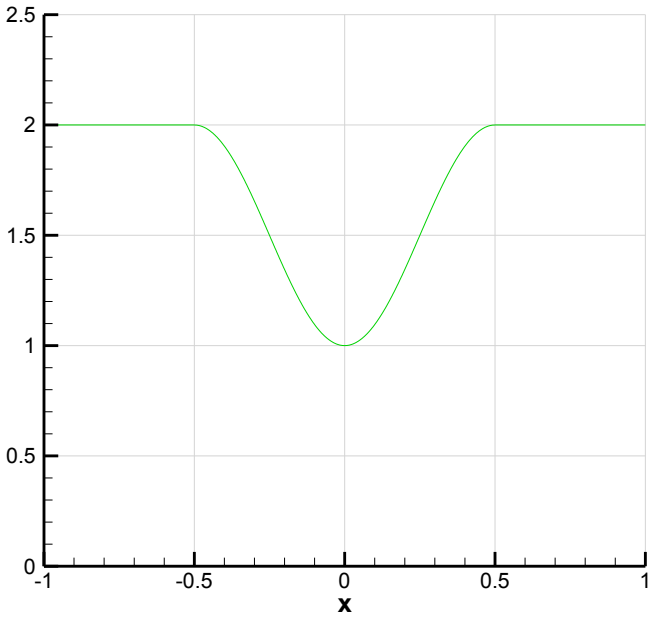

Figure 3-1: Adjoint Validation - Duct Geometry

Boundary conditions

As explained in [**13**], subsonic dimensionless boundary conditions imposed to the equations are the following:

- H_{in} = 4 (inlet Total Enthalpy per unit mass);
- P_{0_in} = 2 (inlet Total Pressure);
- p_{out} = 1.98 (outlet Static Pressure).

Objective function

As reported in [**13**], the adjoint variable has been evaluated respect to the following *J objective function*:

$$J = \int_{-1}^{1} p \, dx \qquad \text{(3-43)}$$

where p is the static pressure.

3.4.4.2 Flow Field Solution

The flow solution has been computed by solving the system of flow field equations with the following parameters:

- number of elements: 500;
- approximation order: *m = k = 1 (isoparametric elements);*
- initial conditions imposed:

$$p^0 = 1, \; T^0 = 1, \; q^0 = 0.5 \qquad \text{(3-44)}$$

Figure 3-2 shows the steady static pressure obtained:

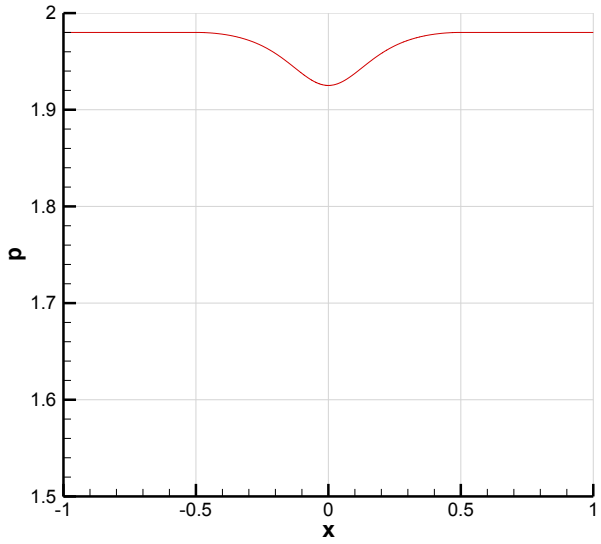

Figure 3-2: Adjoint Validation - Flow Field Solution - Static Pressure

3.4.4.3 Adjoint Solution

The adjoint solution has been computed by solving the system of adjoint equations with the following parameters:

- number of elements: 500;
- approximation order: *m = k = 1 (isoparametric elements);*
- initial conditions imposed:

$$\psi^0 = 0 \qquad (3\text{-}45)$$

Figure 3-3 shows the steady adjoint solution obtained:

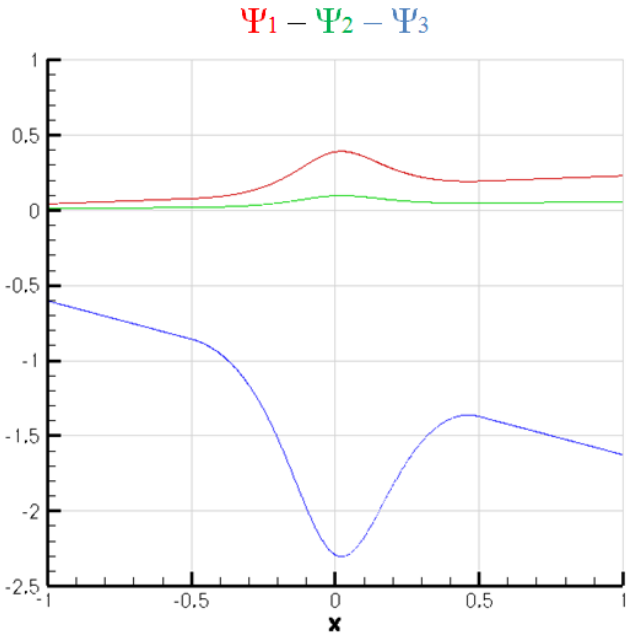

Figure 3-3: Adjoint Validation - Adjoint Variables

The discrete adjoint solution is in close agreement to the analytic adjoint solution.

Each single adjoint variable shows the relation between the sensitivity of a single flow equation in respect of all the flow variables and the sensitivity of the objective function in respect of all the flow variables too. In (3-46) the adjoint system has been reported, without the aspects due to the used discretization:

$$\begin{bmatrix} \frac{\partial R_1}{\partial u_1} & \frac{\partial R_2}{\partial u_1} & \frac{\partial R_3}{\partial u_1} \\ \frac{\partial R_1}{\partial u_2} & \frac{\partial R_2}{\partial u_2} & \frac{\partial R_3}{\partial u_2} \\ \frac{\partial R_1}{\partial u_3} & \frac{\partial R_2}{\partial u_3} & \frac{\partial R_3}{\partial u_3} \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial u_1} \\ \frac{\partial J}{\partial u_2} \\ \frac{\partial J}{\partial u_3} \end{bmatrix} \qquad (3\text{-}46)$$

where

- R_i is the i-th equation of the flow field;
- u_i is the i-th flow variable.

3.5 Shape Optimization Application

The method has then been applied to a shape optimization problem.

3.5.1 Case Description

The shape optimization has been performed on a simple quasi-1D Euler case, exploiting the activity carried out for the validation of the implementation.

Initial Shape

A duct of constant cross-section, equal to the inlet cross section of the validation case, has been chosen as initial shape.

Shape Parameterization

The *parameterization* of geometry has been carried out by a *single design parameter* α, used to modify the duct cross-section:

$$h = h(x,\alpha) \quad \rightarrow \quad h(x) = \begin{cases} 2 & \left(-1 \le x \le -\frac{1}{2} \right) \\ 2 - \alpha\,\Delta & \left(-\frac{1}{2} \le x \le \frac{1}{2} \right) \\ 2 & \left(\frac{1}{2} \le x \le 1 \right) \end{cases} \qquad (3\text{-}47)$$

where

- α is the design parameter;
- Δ is a fixed quantity defined as

$$\Delta = 2 - \left[1 + \sin^2(\pi x) \right] \qquad (3\text{-}48)$$

The described parameterization leads to the following condition:

- initial shape: $\alpha_{start} = 0$ → constant cross section duct;
- final (target) shape: $\alpha_{target} = 1$ → duct equal to the validation case.

Figure 3-4 shows the initial shape (*A*) and the final (target) shape (*Atgh*).

Objective function

The *objective function* followed to modify the shape is the following:

$$J = \frac{1}{2} \int_{-1}^{1} \left(p - p_{tgh} \right)^2 dx \qquad (3\text{-}49)$$

where

- p is the static pressure along the duct, evaluated at the current configuration;

- p_{tgh} is the target static pressure, equal to the static pressure distribution of the validation case.

Boundary conditions

The same boundary conditions imposed on the validation case have been used:

- H_{in} = 4 (inlet Total Enthalpy per unit mass);
- P_{0_in} = 2 (inlet Total Pressure);
- p_{out} = 1.98 (outlet Static Pressure).

Figure 3-5 shows the initial static pressure (p) and the final (target) static pressure (p_{tgh}) distribution.

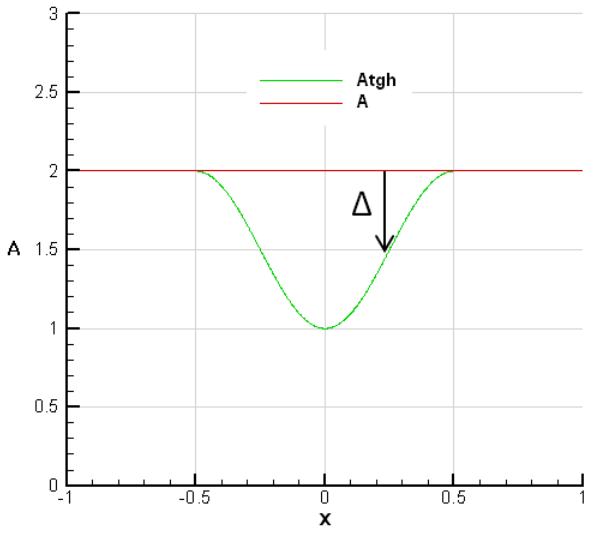

Figure 3-4: Adjoint - Shape Optimization - Shape Parameterization

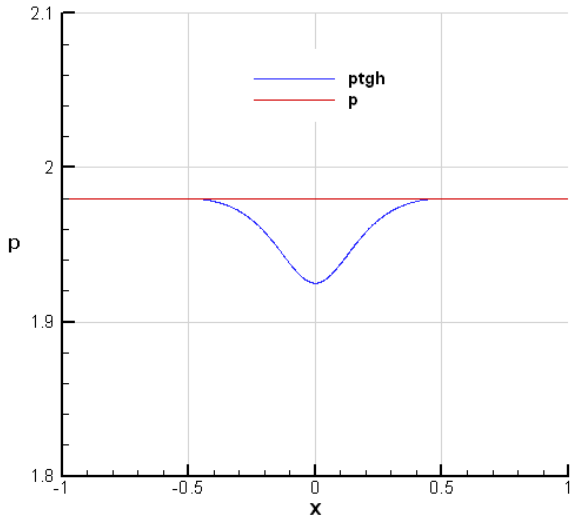

Figure 3-5: Adjoint - Shape Optimization – Initial and Target Static Pressure

3.5.2 Optimization Process

In the present paragraph the main aspects of the optimization process are resumed:

- *optimization algorithms*: as explained in 3.3.6, the variation of the design parameter α has been obtained by a *line search approach* with a *steepest descent* (gradient-based) algorithm:

$$\Delta\alpha = -\Delta\frac{dJ}{d\alpha} \qquad (3\text{-}50)$$

- *gradient*: as explained in 3.3.4, the sensitivity of the objective function in respect of the design parameter has been evaluated by solving the discrete adjoint system:

Sensitivity Evaluation

$$\frac{dJ}{d\alpha} = \boldsymbol{\psi}^T\left(-\frac{\partial \mathbf{R}}{\partial\alpha}\right)+\frac{\partial J}{\partial\alpha}$$

Constraint – Adjoint Problem (3-51)

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^T \boldsymbol{\psi}^T = \left(\frac{\partial J}{\partial \mathbf{U}}\right)^T$$

- *step length*: as explained in 2.1.2.1, the length of the step Δ used at each iteration has been evaluated by an *inexact line search approach*, with the aim to satisfy the *Armijo condition*:

$$f\left(x^k + \Delta d^k\right) \le f\left(x^k\right) + \gamma\,\Delta\,\nabla f\left(x^k\right)' d^k \qquad (3\text{-}52)$$

- $\frac{\partial \mathbf{R}}{\partial\alpha}$: the derivative of residual in respect of the design parameter has been computed by means of the *forward finite difference* formula:

$$\frac{\partial \mathbf{R}}{\partial\alpha} \approx \frac{\mathbf{R}(\alpha+h)-\mathbf{R}(\alpha)}{h} \qquad (3\text{-}53)$$

where h is the increment parameter, equal to 0.000001.

- *validation of derivative*: as suggested in [**17**], the values of derivatives computed by the adjoint approach have compared to those computed by means of the *central finite difference* formula:

$$\frac{dJ}{d\alpha} \approx \frac{J(\alpha+h)-J(\alpha-h)}{2h} \qquad (3\text{-}54)$$

where h is the increment parameter, equal to 0.000001.

3.5.3 Results

Results obtained in the optimization process are the following:

- the target objective function has been reached in less than 10 iterations;
- the values of derivatives in the course of optimization were found to be in close agreement with those obtained by means of the finite difference approach.

Figure 3-6 and Figure 3-7 show respectively the evolution of the shape and of the static pressure during the optimization process.

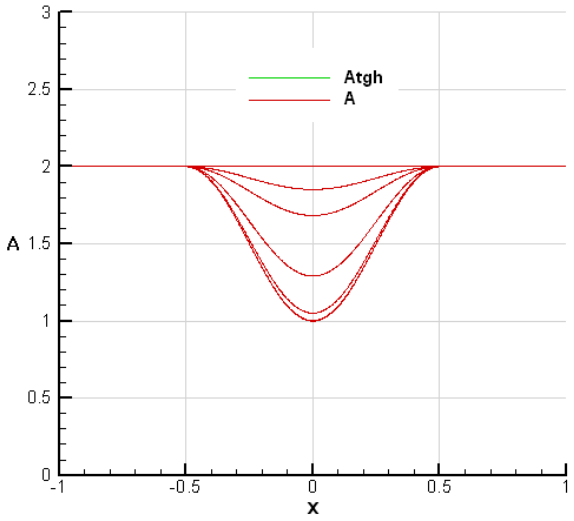

Figure 3-6: Adjoint - Shape Optimization - Shape Evolution

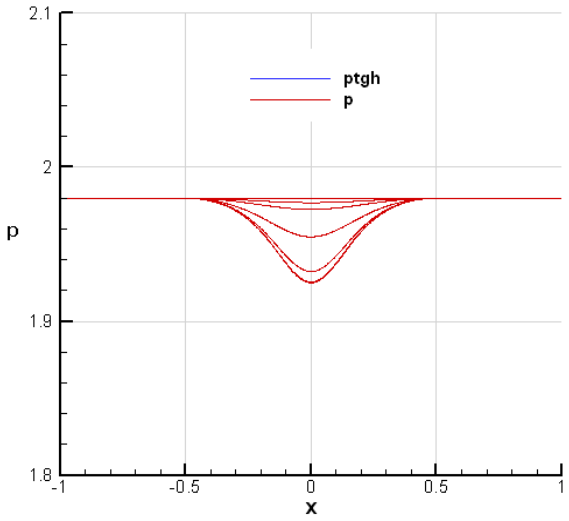

Figure 3-7: Adjoint - Shape Optimization – Static Pressure Evolution

Figure 3-8 shows the evolution of the objective function, the derivatives evaluated by means of the adjoint approach and those obtained by means of the finite difference approach.

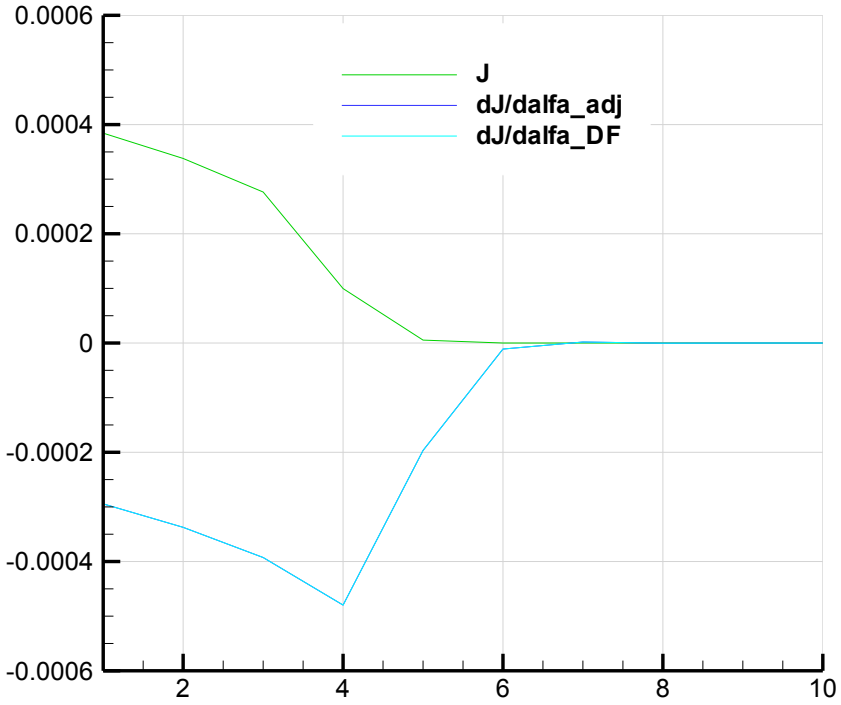

Figure 3-8: Adjoint - Shape Optimization – Validation of Derivative

Table 3-1 shows a comparison of derivatives evaluated by means of the adjoint approach and those obtained by means of the finite difference approach at each iteration.

dJ/dα			
ITERATION	ADJ	DF	(ADJ-DF)/DF
1	-0.000294486202	-0.000294486246	-0.000000149413
2	-0.000337435228	-0.000337435184	0.000000130395
3	-0.000392597790	-0.000392597847	-0.000000145187
4	-0.000479806263	-0.000479806238	0.000000052104
5	-0.000196644963	-0.000196644990	-0.000000137303
6	-0.000011074630	-0.000011074632	-0.000000207682
7	0.000001845356	0.000001845356	0.000000059744
8	-0.000000328360	-0.000000328360	-0.000000111806
9	0.000000057814	0.000000057814	0.000000243263
10	-0.000000010198	-0.000000010198	0.000000760401

Table 3-1: Adjoint - Shape Optimization – Validation of Derivative

Figure 3-9 shows the evolution of the adjoint variables value during the optimization at iterations 1, 3, 5 and 10.

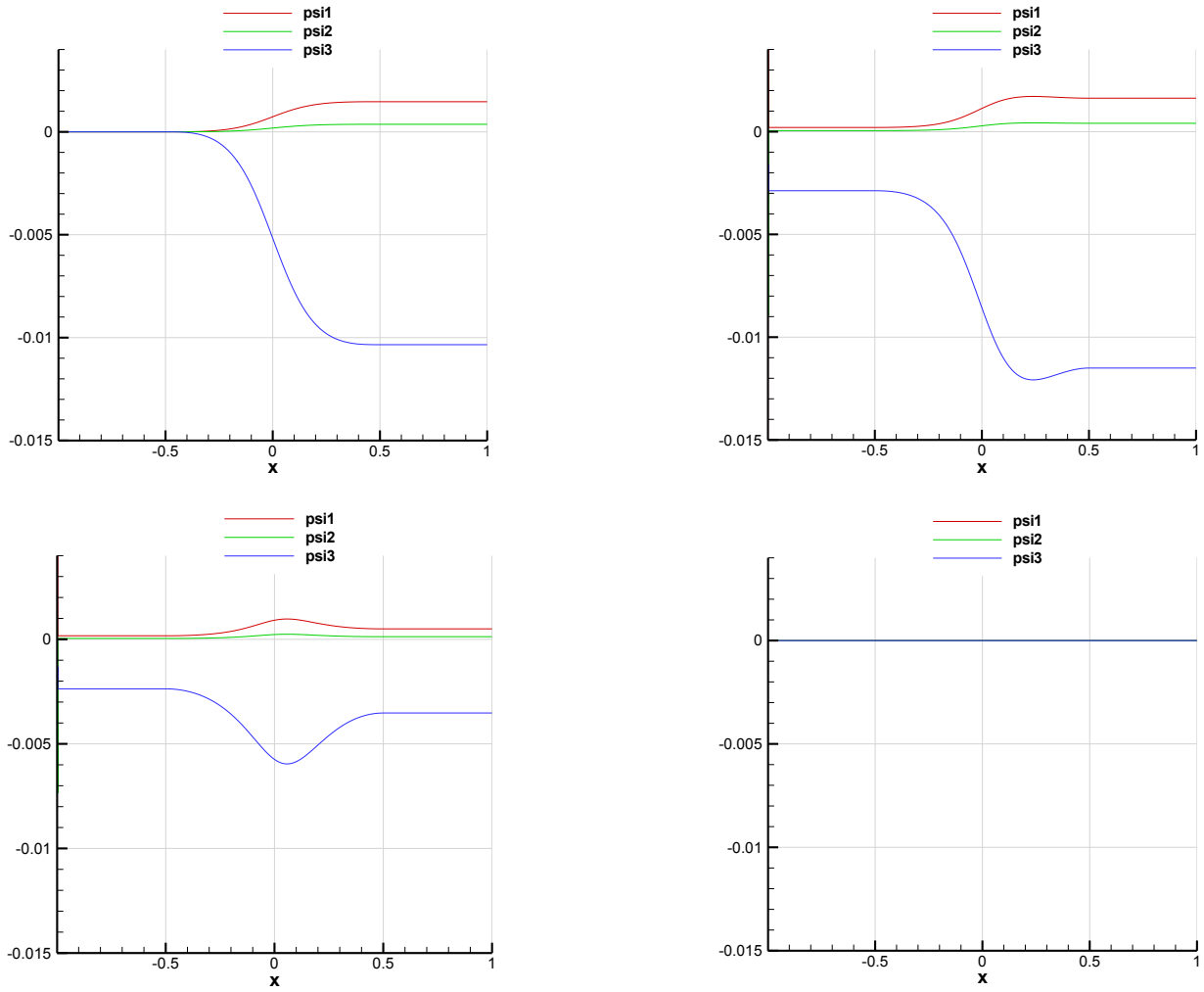

Figure 3-9: Adjoint – Shape Optimization – Adjoint Variable

3.6 Conclusion

A new direction of development within the research group operating at University of Bergamo on DG methods for Computational Fluid Dynamics has been started.

The present work is about the use of the adjoint method in the framework of aerodynamic shape optimization. The adjoint approach allows to solve local constrained optimization problems efficiently: the solution of only one linear system related to the constraint is required, independently from the number of the involved design parameters. The technique is applied to shape optimization and grid adaptivity problems for CFD applications, where the number of the involved parameters is generally high.

Starting from the simple quasi-1D Euler equations, the implementation of the discrete adjoint method has been validated by comparing the values of computed adjoint variables with results of analytical solutions available in the literature.

The method has then been applied to a shape optimization problem with a single design variable, using a gradient based algorithm with an inexact line search approach. The sensitivity of the objective function in respect of the design parameter has been evaluated by solving the adjoint system, while the length of the step at each iteration has been quantified requiring the satisfaction of the Armijo condition. The derivative of residual in respect of the design parameter has been computed by means of the finite difference approach. The target value of the objective function has been reached in less than 10 iterations and the values of derivatives in the course of optimization were found to be in close agreement with those obtained by means of the finite difference approach.

The application of the method to practical shape optimization problems requires further developments: the extension to other objective functions, the introduction of surface parameterization techniques, in order to deal with more complex geometries, the extension of the analysis to 2D-3D cases and to viscous and unsteady cases.

4 Thermal Cycle Optimization of a Mandrel Mill Component

The second part of the thesis deals with the application of optimization techniques to an industrial problem.

4.1 Overview of the Work

This activity focused on the optimization of the thermal cycle of the mandrel of a longitudinal mandrel mill devoted to the production of seamless steel pipes, with the objective of reducing the peak temperature of the mandrel during the rolling phase.

The present chapter is structured as follows: 4.2 includes a general description of the analyzed case, referring to the involved aspects; 4.3 deals with a description of the used tools and of the performed optimization; 4.4 concerns a description of the obtained results and 4.5 includes conclusions of the work.

4.2 Case Description

4.2.1 Production Cycle of Seamless Steel Pipes

The production cycle of seamless steel pipes includes the following phases:

- *Heating*: a filled *billet* has heated in a furnace, with the aim to achieve a sufficient temperature for the further hot processes. The billet temperature at the exit of furnace is of 1250°C;
- *Piercing*: the central cavity is produced by a piercer, through a hot plastic deformation process. A pierced cylindrical pipe, called *shell,* is obtained from a billet. The process takes place through external cylinders, called *rolls*, that trail longitudinally the billet on a central plug, creating the central cavity;
- *Rolling*: a specific geometry is conferred to the pipe. External rolls trail longitudinally the shell, pressing it on a central cylinder, called *mandrel*. The process occurs along subsequent rolling stands, producing a progressive variation of the geometry. The product at the exit of the mandrel mill is called *tube*.
- *Reheating*: the tube is reheated further, in order to achieve a sufficient temperature for the subsequent hot processes;
- *Calibration*: the geometry is calibrated, with the aim to obtain the final geometry of the product, called *pipe*. The process takes place through external rolls, that trail longitudinally the tube, giving it the desired geometry.
- *Cooling*: final cooling of the pipe on a cooling bed.

After these phases other thermal treatments and cool phases follow, in order to obtain the final features of the product. These phases are not relevant for the activity carried out in the present work; hence, their description is not presented.

A scheme of the described production cycle is present in Figure 4-1.

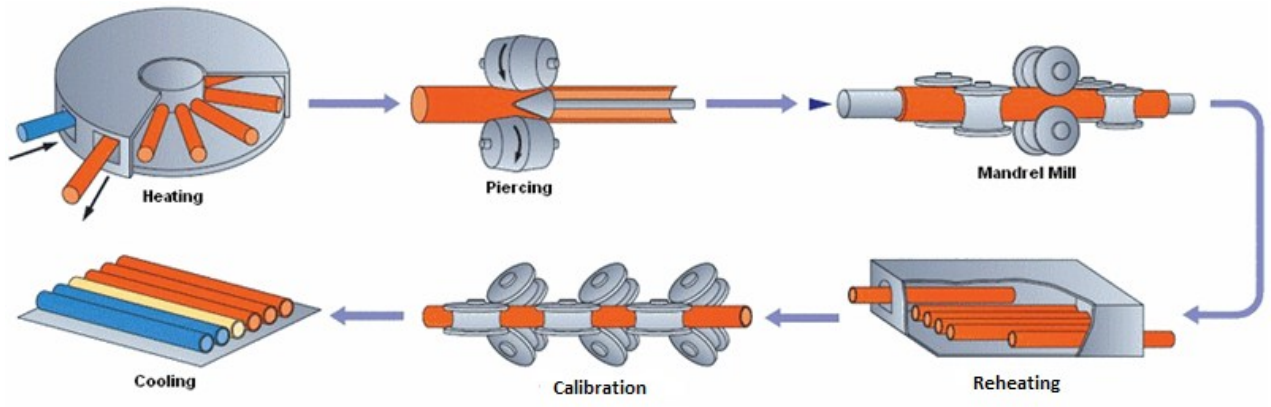

Figure 4-1: Seamless Steel Pipes Production Cycle

In the Thesis, from this point the *shell* will be called *pipe*.

4.2.2 Thermal cycle of the mandrel during the Rolling Phase

4.2.2.1 Thermal Loads

In order to obtain a thickness reduction, during the rolling phase the pipe is trailed and pressed by external rolls on the mandrel. The effective phase when the thickness reduction takes place concerns instants when the pipe is in contact with rolls (from this point called *rolling phase*), while during instants between the stands (from this point called *inter-stand phase*) there is not a thickness reduction. The thermal cycle of the mandrel is due to the following phenomena:

- *effective thickness reduction phase* (*rolling phase*): during this phase, the *mandrel* is subjected to thermal loads due to:
  - *conduction*: in the entire rolling process the temperature of pipe is of 1100°C, while at the beginning of the phase the temperature of the mandrel is of 90°C. This difference of temperature causes a conduction heat flux during the contact phases. Further, the high contact pressure increases the heat exchange thermal coefficient. Treating the *conductive thermal flux* present on the surface of the mandrel like an *equivalent convective flux*, on each stand the conductive heat flux acting on the mandrel is the following:

$$q_{COND} = q_{CONV\_eq} = h_{eq} \cdot (T_P - T_M) \qquad (4\text{-}1)$$

    where
    - q_{COND} is the conduction heat flux between the pipe and the mandrel;
    - q_{CONV_eq} is the equivalent convective heat flux acting on the mandrel;
    - h_{eq} is the equivalent convective heat transfer coefficient;
    - T_P is the pipe temperature;
    - T_M is the mandrel temperature;

  - *friction*: the thickness reduction of the pipe causes a reduction of its cross section. Hence, along the mandrel mill the pipe speed progressively increases, while the mandrel speed is constant: there is always a *sliding* between the pipe and the mandrel, due to the difference of speed. Further, the high contact pressure causes a high *share stress component*. These aspects lead to a high friction thermal flux. In addition, not all of the generated friction heat flux enters in the mandrel but only a

portion of it. Hence, on each stand the friction heat flux acting on the mandrel is the following:

$$q_{FR} = \tau \cdot (v_P - v_M) \cdot \%_{ABS} \quad (4\text{-}2)$$

where

- q_{FR} is the friction heat flux generated by the sliding between the pipe and the mandrel;
- τ is the contact shear stress between the pipe and the mandrel;
- v_P is the pipe speed;
- v_M is the mandrel speed;
- $\%_{ABS}$ is the % fraction of friction heat flux that enters in the mandrel;

- *inter-stand phase*: in this phase the mandrel is subjected to thermal loads due to:
  - *conduction*: despite the absence of external rolls in these phases, however contact zones between the pipe and the mandrel are present, with lower pressures respect to the rolling phase. On each stand, the conduction heat flux present on the mandrel is evaluated like explained for the rolling phase, but the values of the heat exchange coefficients present are different;
  - *radiation*: in the areas where there is not contact, the pipe radiates the mandrel. On each stand, the radiation heat flux acting on the mandrel is the following:

$$q_{RAD} = F \cdot \varepsilon \cdot \sigma \cdot (T_P^{\,4} - T_M^{\,4}) \quad (4\text{-}3)$$

where

- q_{RAD} is the radiation heat flux present on the mandrel;
- F is a geometric parameter;
- ε is the pipe emissivity;
- σ is the Stephan Boltzmann constant.

Due to the high contact pressure present and the difference between the pipe and the mandrel speed, the most critical phase is the rolling phase. Under each stand, during these phases the temperature of the mandrel increases more quickly, reaching the higher values.

4.2.2.2 Rolls disposition on Stands

An important aspect of the mandrel thermal cycle concerns the circumferential disposition of rolls along the mandrel mill. In this sense, it is useful to define these two different sections:

- *bottom gorge section*: it is the circumferential section corresponding to the axis of symmetry of rolls;
- *intermediate section*: it is the circumferential middle section between the bottom gorge section of consecutive rolls of a single stand.

In order to guarantee a uniform section of the pipe along the mandrel mill, the circumferential position of rolls is not the same between a stand and its subsequent. There is an angular shift in such a way as to ensure that a section that has not rolled on the current stand, will see the bottom gorge section of rolls of the subsequent stand.

Hence, each circumferential section of the mandrel sees different thermal cycles. In the present work the section considered for the evaluation of the mandrel thermal cycle are the following:

- bottom gorge section of odd stands;
- bottom gorge section of even stands;
- intermediate section (equal for odd and even stands).

Figure 4-2 shows the position of the bottom gorge and of the intermediate sections.

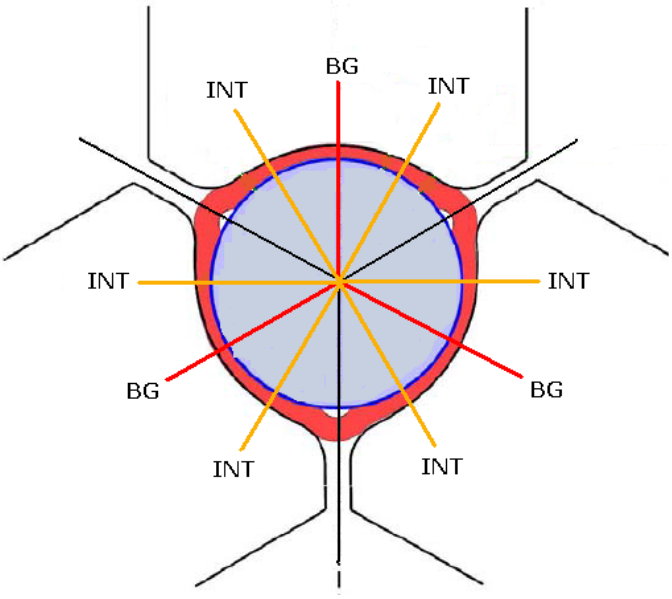

Figure 4-2: Bottom Gorge and Intermediate Section

Figure 4-3 shows the position of the bottom gorge and the intermediate sections of a single roll, in respect of the contact area between the pipe and the mandrel.

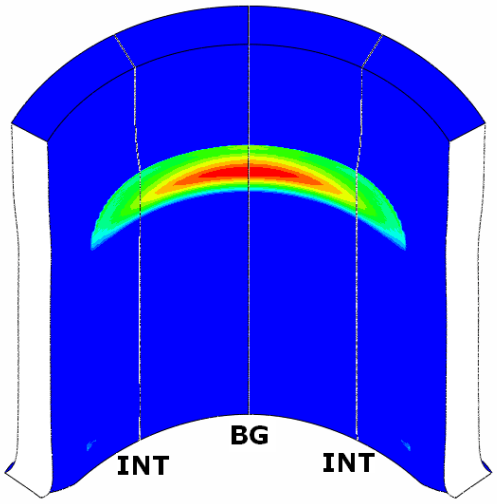

Figure 4-3: BG and INT Section in respect of Contact Area

Images concerning the rolling phase and contact areas involved are present in Figure 4-4.

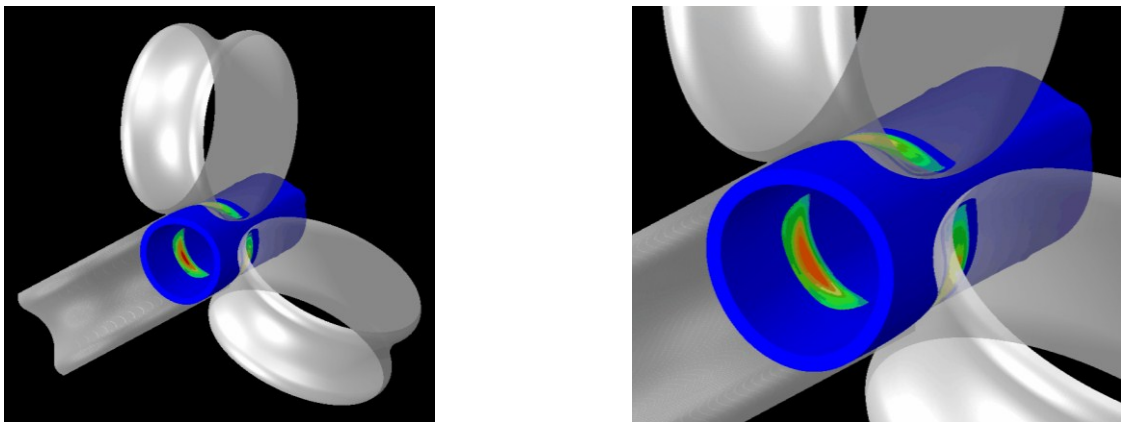

Figure 4-4: Mandrel Mill with 3 Rolls – Single Stand - Rolling Phase - Contact Areas

Images concerning the rolling phase and contact pressure on each roll are shown in Figure 4-5.

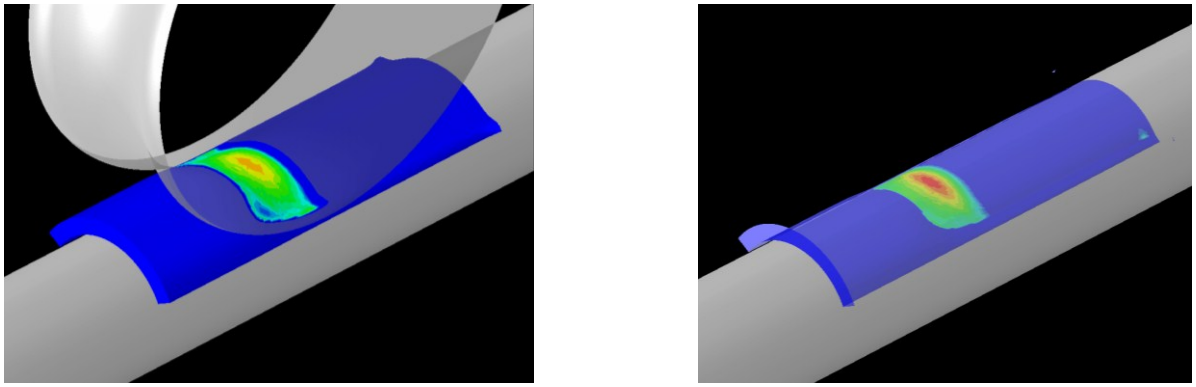

Figure 4-5: Singe Stand - Rolls-Pipe and Pipe-Mandrel Contact Pressure

Images concerning the temperature behavior of the mandrel surface during the rolling phase, in respect of a single circumferential section, are present in Figure 4-6. Notice the peaks of temperature corresponding to the rolling phases.

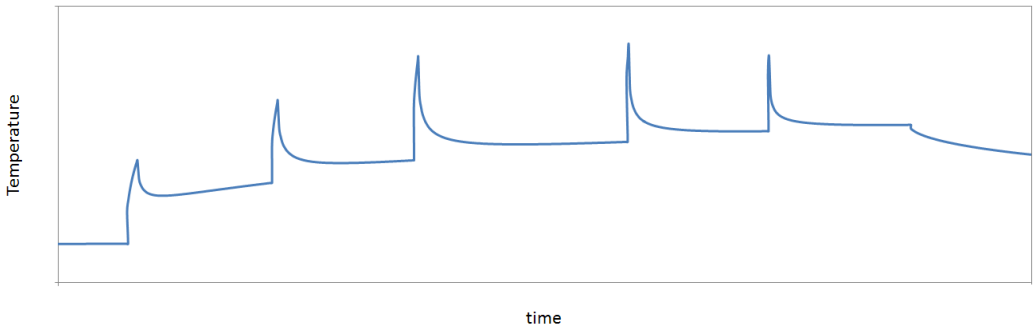

Figure 4-6: Single Circumferential Section - Temperature of the Mandrel Surface Behavior

4.2.3 Effects of the Thermal Cycle on the Mandrel

The mandrel is a very long cylinder composed by AISI H13 steel, with an external coating composed by electrodeposited hard Chromium. The chemical composition of mandrel steel is reported in **Table 4-1**:

AISI H13 STEEL				
C [%]	Si [%]	Cr [%]	Mo [%]	V [%]
0.4	1	5.2	1.4	1

Table 4-1: Chemical Composition of Steel of the mandrel

The thermal cycle causes the following effects on mandrel steel:

- *Hardness reduction*: if the temperature exceeds the discovery temperature, the steel hardness decreases; this leads to a softening of the coating basement;
- *Austenitizing*: if the temperature exceeds the start austenitizing limit, a variation of metal structure of steel occurs; this leads to a damage of the coating basement.

The mandrel life depends strictly on the integrity of the coating of the mandrel: a damage of the external coating causes a waste of the component. For this reason it is important to avoid a softening and a damage of the coating basement and, for the reasons explained, limit the peak of temperature of the mandrel surface.

4.2.4 Approach to the Problem

The peak of temperature of the mandrel is reached during the rolling phase. Due to high gradients of temperature present on the first 3-5 mm deep and due to the high speed of the involved phenomena (speed of temperature variation higher than 1000°C/s) is not possible to measure the real evolution of the mandrel temperature. Hence, it has been decided to deal with the problem numerically. Tools used in the analysis are the following:

- Preset Model for the evaluation of setup of the mandrel mill;
- Finite Element Model for the simulation of the rolling process;
- Finite Difference Model for the simulation of the mandrel thermal evolution;
- Micro structural Model for the prediction of the hardness fail.

4.2.5 Case Object of the Work

The case object of this work refers to a mandrel mill of TenarisDalmine S.p.A. with 6 stands; each of one has 3 external rolls.

Due to reasons concerning the TenarisDalmine's industrial secret policy, all data and results concerning the problem are not mentioned directly but they are presented in dimensionless form, in respect of a reference case, not mentioned.

4.3 Optimization Process

4.3.1 Parameters and Constraints

Parameters involved in the process are the following:

- *number of stands* (N): number of stands present in the mandrel mill;
- *stand elongation* (λ): parameter that describes the elongation of the pipe under each single stand. This quantity is function of rolls position, that determines the thickness variation of

the pipe, having effects on contact area (A), on contact length (l) and on contact pressure distribution (p) between the pipe and the mandrel. The elongation is defined as the ratio of the transversal area of the pipe, between the inlet and the outlet of the stand:

$$\lambda_{stand(i)} = \lambda_i = \frac{A_{trasv\_PIPE-IN\_stand(i)}}{A_{trasv\_PIPE-OUT\_stand(i)}} \quad (4\text{-}4)$$

- *mandrel speed* (v_M): velocity of the mandrel during the rolling phase. This parameter has effect on the relative speed (v_r) between the pipe and the mandrel;
- *pipe outlet speed* (v_P): velocity of pipe at the exit of the last stand of the mandrel mill. This quantity has effect on the relative speed (v_r) between the pipe and the mandrel;
- *rolls diameter* (D): diameter of external rolls. This parameter has effects on contact area and on contact length between the pipe and the mandrel;
- *inter-stand distance* (d): distance between two consecutive stands. This quantity has effects on time (t) during which a mandrel section passes from a stand and its subsequent.

Constraints present in the analysis are the following:

- *bound constraints*: a part the number of stand, fixed in this analysis, each parameter has an own range of values. The reasons of each constraint present are presented:
  - *stand elongation*:
    - *MIN*: each single stand must lead to a minimal reduction of the transversal area of pipe;
    - *MAX*: there is a limit on mechanical loads of each stand;
  - *mandrel speed*:
    - *MIN*: in order to limit friction components, there is a limit on the minimum relative speed between the pipe and the mandrel;
    - *MAX*: there is a limit on the maximum length of the mandrel;
  - *pipe outlet speed*:
    - *MIN*: a limit on the minimum productivity is present;
    - *MAX*: there is a limit on the installed motors power;
  - *inter-stand distance*:
    - *MIN*: due to the dimension of each stand;
    - *MAX*: a limit on the total dimension of mandrel mill is present.
- *other constraints*:
  - *pipe dimension*: in order to obtain a determined dimension of the pipe, the *total elongation* of all the mandrel mill must be kept constant:

$$\lambda_{TOT} = \prod_{i=1}^{N} \lambda_i = K \quad (4\text{-}5)$$

  - *inlet mandrel speed*: due to engineering aspects of the mandrel mill, the mandrel must be always pulled by the pipe. Hence, mandrel speed must be always lower than the pipe speed. The lowest value of pipe speed is present on the first stand, hence:

$$v_M \leq v_P\big|_{stand1} \quad (4\text{-}6)$$

4.3.2 Tools Description

This section concerns a general description of tolls available at R&D Centre of TenarisDalmine S.p.A., used for the problem solution.

4.3.2.1 Preset Model

The Preset Model is a software developed at the R&D Centre of TenarisDalmine S.p.A., in order to identify the correct setup configuration of the mandrel mill for the production of a determined product.

The Preset Model includes several calculation modules that are briefly described in next paragraphs, concerning the analysis carried out in the present work.

Geometrical module

The Preset Model is able to calculate, in correspondence of each working stand, the following data:

- spatial position of the rolls and pass data;
- external / internal material profile at the outlet section;
- circumferential distribution of the wall thickness of the pipe;
- pipe cross sections, pipe outer diameter, pipe perimeter, circumferential contact angles between rolls and mandrel, elongations;
- rolls position by which obtain a determined wall thickness at the outlet of the mandrel mill.

Kinematics module

The Preset Model is able to calculate, in correspondence of each working stand, the following data, according to several plant and technological constrains:

- stands kinematics:
  - pipe length and speed;
  - rolls and motors rotational speed;
- mandrel kinematics:
  - mandrel speed;
  - mandrel front abscissa;
- pipe kinematics:
  - pipe front abscissa;
  - pipe speed.

Dynamics Module

The Dynamics Module has been carried out using the *flat rolling equivalence,* with the aim to simplify the approach to the problem. The Preset Model is able to calculate, in correspondence of each working stand, the following data:

- material-roll relative speeds;
- material–mandrel relative speeds.

Details about each module are presented in the next sections.

Geometrical module

Figure 4-7 shows a simplified logical flow of geometrical calculus in the Preset Model.

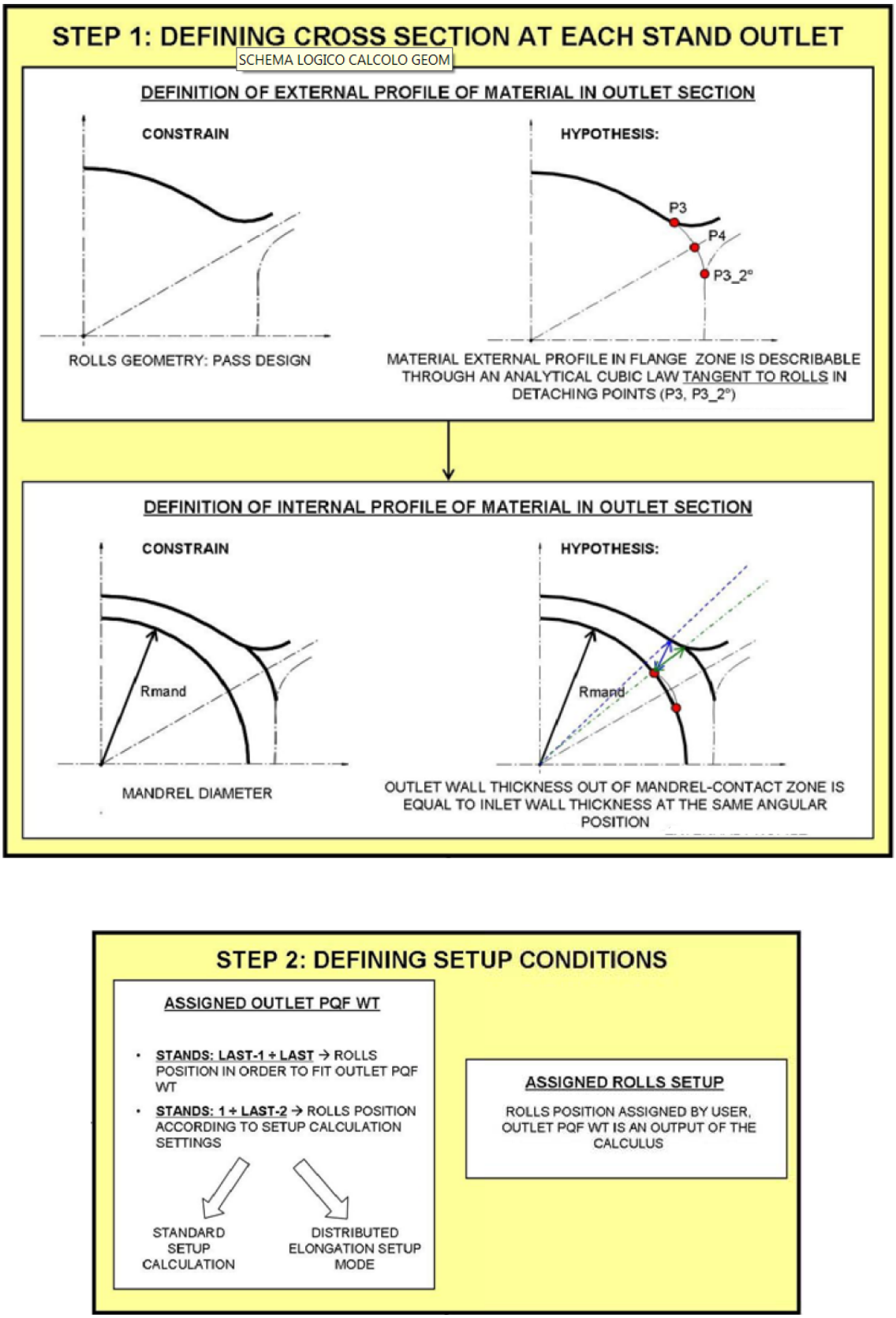

Figure 4-7: Geometrical Module - Flow Chart

A description of the criteria used for the setup definition is presented, concerning the present work. Setup evaluation is divided by two sequential steps:

- *finishing stands setup (N-1 up to Nstand)*: setup is obtained by adjusting rolls position of the same quantity, until the relation (4-7) is satisfied:

$$\left| WT_{ave,out} - WT_{obj} \right| \leq WT_{tolerance} \qquad (4\text{-}7)$$

where
  - WT_{obj} is the scheduled wall thickness at mandrel mill exit;
  - $WT_{tolerance}$ is the adopted tolerance, equal to 0.001 mm;
  - $WT_{ave,out}$ is the average wall thickness evaluated at each i-th stand outlet, according to the following formula:

$$WT_{ave,out@i} = \frac{\sum_{\vartheta=-30°,0.5°}^{90°} WT(\vartheta)_{out@i}}{241} \qquad (4\text{-}8)$$

where $WT(\theta)$ is the wall thickness evaluated at θ circumferential angle.

- *roughing stands setup (1up to N-2 stand)*: different possible option are available. In this work has been used the '*distributed elongation with fixed reference values'* option: setting of the roll position on the roughing stands based upon reference deformation distribution along the mandrel mill. Basically rolls position at each stand is calculated in order to obtain a particular elongation $\lambda_{work@i}$, according to the following formula:

$$\lambda_{work@i} = \lambda_{ref@i}^{kw} \qquad (4\text{-}9)$$

where
  - $\lambda_{ref@i}$ isa reference average elongation at stand i;
  - kw is a generic coefficient evaluated by the total elongation in both conditions, working and reference:

$$kw = \frac{\ln(\lambda_{work-tot})}{\ln(\lambda_{ref-tot})} \qquad (4\text{-}10)$$

The coefficient kw modifies each stand elongation, ensuring that the real elongation distribution produces the target total elongation.

Notice that in this point the Preset Model satisfies the constraint on the total elongation of the mandrel mill.

Kinematics module

Concerning the present work, a description of the definition of the following quantities is presented:

- *pipe speed*: based on the pipe speed of subsequent stand, the pipe speed at the outlet of each stand is obtained as explained in (4-11):

$$V_{out@i} = \frac{V_{out@i+1}}{\lambda_{work@i+1}} \quad (4\text{-}11)$$

- *rolls rpm*: rolls speed of each stand is obtained as explained in (4-12):

$$RPM_{roll@i} = \frac{60 * V_{out@i}}{\pi * D_{pitch@i}} \quad (4\text{-}12)$$

where

- $D_{pitch@i}$ isthe *Roll Pitch Diameter* of stand-i, used to consider the variation of roll diameter along the circumferential direction:

$$D_{pitch@i} = RD_{nom@i} - PICTH\_COEFF_{@i} * OD_{eq@i} \quad (4\text{-}13)$$

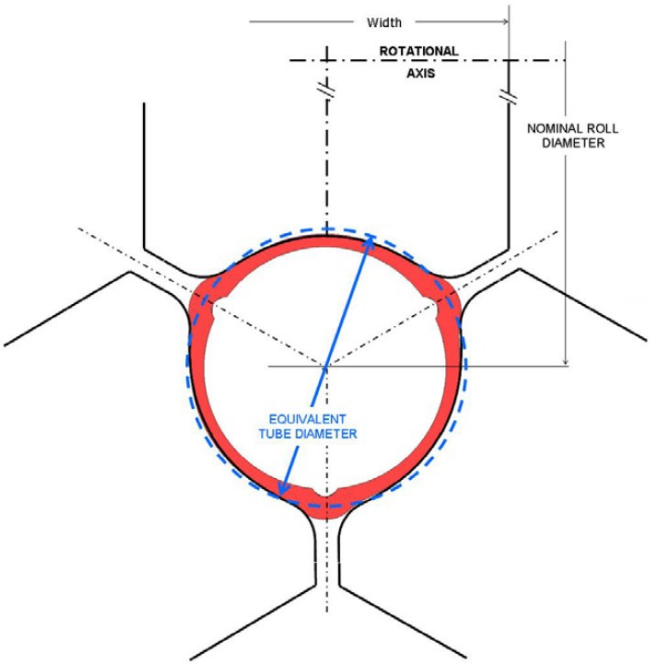

Figure 4-8: Roll Pitch diameter

- $OD_{eq@i}$ is the *Material Equivalent Outer Diameter* of stand-i, calculated in respect of the cross section area and the average wall thickness;
- *Pitch Coefficient*: coefficient [0,1].

- *Mandrel speed*: mandrel speed is defined by the following formula:

$$VMR = VMR_{start} * \frac{CVMP}{100} \quad (4\text{-}14)$$

where

- *VMR*: mandrel retaining speed;
- *CVMP*: master percentage applied to the initial value of the mandrel retaining speed [0, 100%];
- VMR_{start}: initial value of the mandrel retaining speed that the Preset Model uses for the mandrel kinematics calculation, according to the following formula:

$$VMR_{start} = \min\left(VMR_{\max}; \frac{V_{out-0} + V_{out-1}}{2}\right) \quad (4\text{-}15)$$

where

- VMR_{max} is the maximum mandrel retaining speed, depending on plant limits;
- $\frac{V_{out-0} + V_{out-1}}{2}$ is the average shell speed under the first stand; this condition is necessary to satisfy the technological constraints by which mandrel speed < material speed during rolling.

If the mandrel retaining speed *VMR* satisfies the boundary on the first stand, the Preset Model accepts this value, otherwise it progressively reduces this value, until the boundary satisfaction.

Notice that at this point the Preset Model satisfies the constraint on the retaining mandrel speed of the mandrel mill.

Dynamics Module

Concerning the present work, a description of the methodology used by the Preset Model to calculate relative speeds is presented.

Flat Rolling Equivalence

In order to have a simplified two-dimensional approach to the problem of the mandrelmMill dynamics, the Preset Model uses the “flat rolling equivalence”, considering the rolling of a flat plate.

The first step of the Dynamics Module is to evaluate the flat plate and the roll geometry to use in following calculation steps. According to Figure 4-9, for each i-th working stand it is possible to define the following correspondences between geometrical parameters in plate and the rolling of pipe.

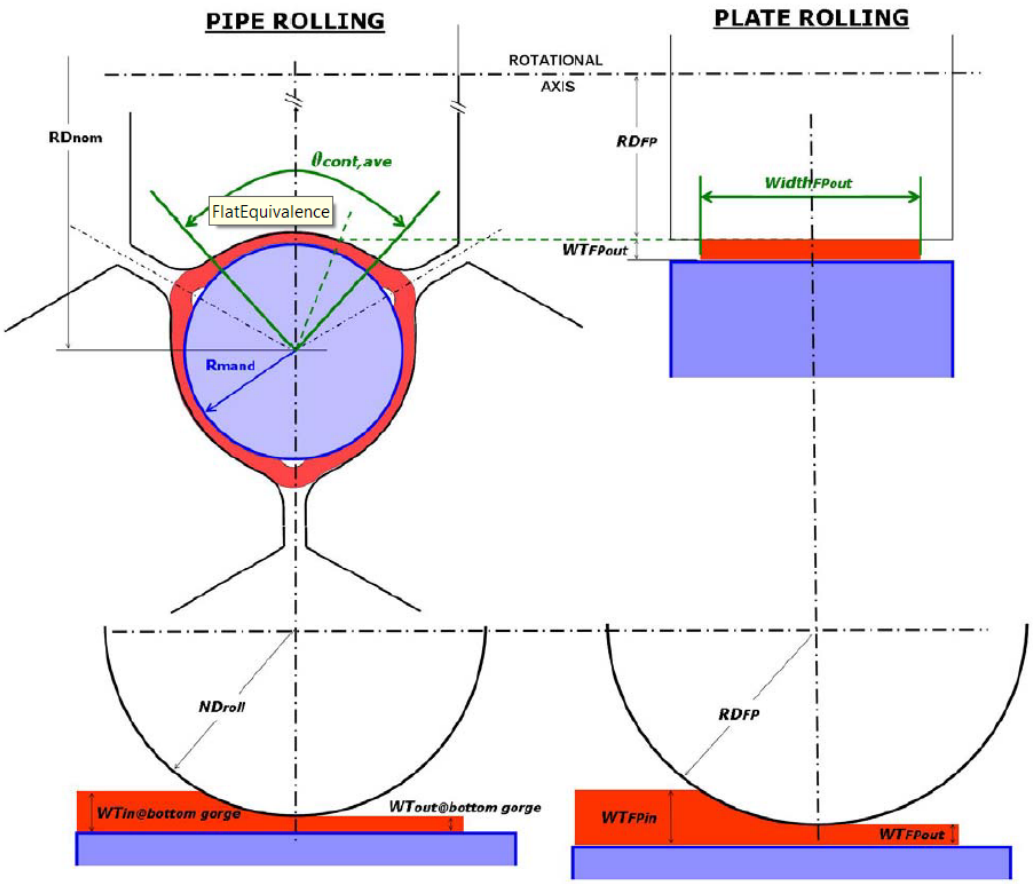

Figure 4-9: Flat Rolling Equivalence - Calculation of Flat Plate and Roll

The wall thickness of the plate before and after the rolling stage is assumed to be equal to the wall thickness of the pipe in the rolled zone:

$$\begin{cases} WT_{FPin@i} = WT_{in-roll@i} \\ WT_{FPout@i} = WT_{out-roll@i} \end{cases} \quad (4\text{-}16)$$

The plate width is evaluated depending on the wall thickness of the plate and the average contact angle:

$$Width_{FPout@i} = \vartheta_{cont,ave@i} * \left(R_{mand} + WT_{FPout@i}\right) \quad (4\text{-}17)$$

where

- R_{mand} is the mandrel radius;
- $\theta_{cont,ave@i}$ is the average contact angle (this calculation is not presented in this document).

In correspondence of each stand, the flat roll diameter is defined as:

$$RD_{FP@i} = RD_{nom@i} - \left(R_{mand} + WT_{FPout@i}\right) * \cos\left(\frac{\vartheta_{cont,ave@i}}{4}\right) \quad (4\text{-}18)$$

Relative speeds

This section concerns formulas and hypothesis made by the preset model to the evaluation of relative speeds between material, roll and mandrel at contact interfaces.

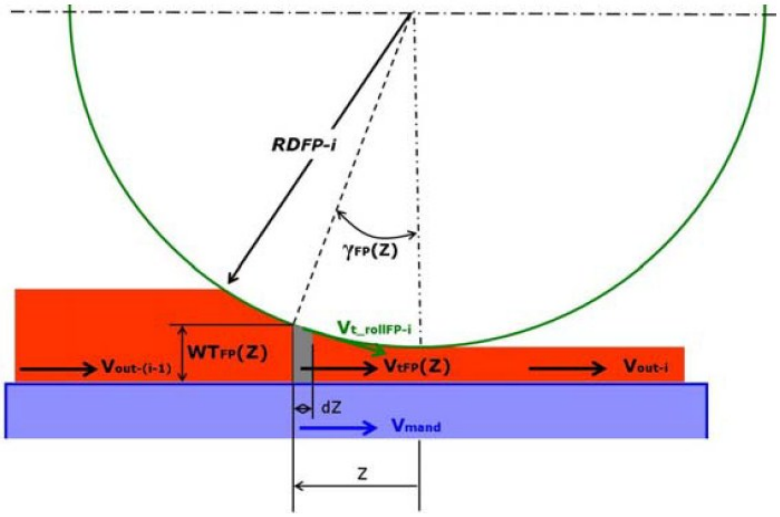

Figure 4-10: Calculation of Relative Speed

According to Figure 4-10 it is easy to understand that the mandrel speed is always constant and equal to the retaining speed *VMR*.

The definition of the material speed in correspondence of each linear position Z is made by assuming that the material flow, during the rolling of the flat plate, is the same of the corresponding stand of the mandrel mill. In other words, the assumption is that inlet and outlet speeds are the same.

In order to make this condition verified, the preset model considers in correspondence of each position Z a cross section composed by two contributes: a deformed area $A_{CS\_defFP}(Z)$ and a non-deformed area (constant along the contact zone) $A_{CS\_indefFP@i}$.

The non-deformed area is determined according to material flow conservation:

$$A_{CSFP@(i-1)} * V_{out@(i-1)} = A_{CSFP@(i)} * V_{out@(i)} \quad (4\text{-}19)$$

where $A_{CSFP@i}$ is the total outlet cross section area in correspondence of the i-th stand:

$$A_{CSFP@i} = A_{CS\_defFP@i} + A_{CS\_indefFP@i} = With_{FPout@i} * WT_{FPout@i} + A_{CS\_indefFP@i} \quad (4\text{-}20)$$

Hence

$$A_{CSindefFP@i} = \frac{\left(Width_{FPout@(i-1)} * WT_{FPout@(i-1)}\right) * V_{out@(i-1)} - \left(Width_{FPout@i} * WT_{FPout@i}\right) * V_{out@i}}{V_{out@i} - V_{out@(i-1)}} \quad (4\text{-}21)$$

Once the non-deformed area has been fixed, it is possible to evaluate the speed of material in each subdivision of the contact zones:

$$V_{tFP}(Z) = \frac{A_{CSFP@i} * V_{out@i}}{A_{CSFP}(Z)} \quad (4\text{-}22)$$

where $A_{CSFP}(Z)$ is the cross section area in correspondence of the linear position Z of considered stand (its calculation is not presented in the present work).

Hence, it is possible to calculate relative speeds on internal surface of the flat plate in correspondence of each Z position of the contact zone:

$$\Delta V_{\mathrm{int}FP}(Z) = V_{tFP}(Z) - V_{mand} \quad (4\text{-}23)$$

Out of the contact zone, after the stand exit in the inter-stand area, the relative speed between material and mandrel is:

$$\Delta V_{\mathrm{int}\_SE@i} = V_{out@i} - V_{mand} \quad (4\text{-}24)$$

The relative speed between mandrel and pipe, output of the preset model, is the average of relative speeds values calculated in each contact length subdivision.

4.3.2.2 FEM Model

The Rolling Process has been simulated by a Finite Element Model (FE), using the commercial code ABAQUS EXPLICIT, currently supplied at the R&D Centre of TenarisDalmine S.p.A.. The FE model receives as input the values of the mandrel mill setup from the Preset Model, the flow stress curves of the pipe material, the temperature of the pipe and tribological laws concerning pipe-roll and pipe-mandrel contact.

In a previous phase of the present work, the FE model has been calibrated on rolling loads measured on plant and on the pipe geometry obtained at the exit of each stand, requiring the cutting of pipes during the rolling process.

The FE model is an axial symmetric model: only a circular sector equal to half roll cylinder is simulated, exploiting the circumferential symmetry of the mandrel mill. In addition, the FE model simulates each single stand separately: not all the pipe is simulated but only a shorter piece; the outlet geometry from a stand determines the inlet geometry of the subsequent stand. The simulation is referred to the steady rolling phase, with the aim to evaluate contact lengths and contact pressure between the pipe and the mandrel in that condition. Figure 4-11 represents a circular sector equal to an entire roll cylinder, highlighting contact areas and contact pressures. In respect of Figure 4-11, the FE model simulates only the half portion of the shown configuration.

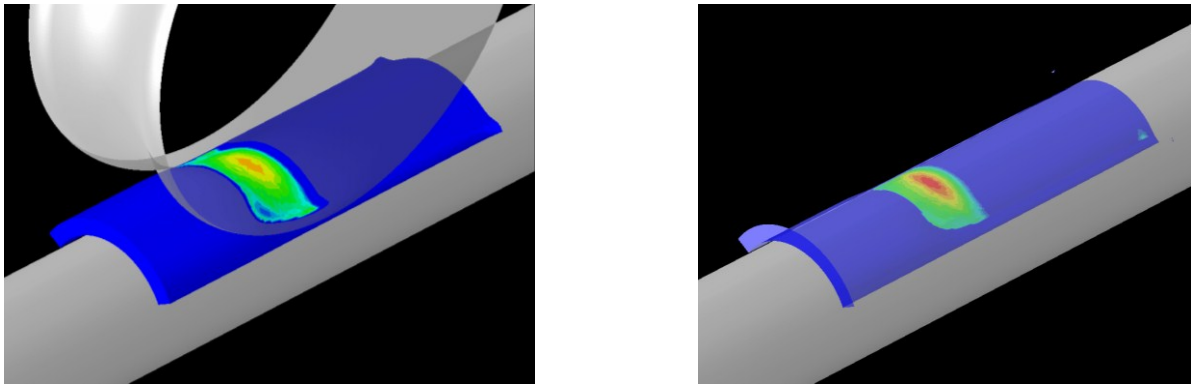

Figure 4-11: FE Model: Pipe-Roll and Pipe-Mandrel Contact Area and Contact Pressure

Each FE analysis simulates stands 1, 2, 3, 4, requiring about 1.2 hours of CPU time.

At the end of the FE analysis, contact lengths and contact pressures between the pipe and the mandrel have been extracted. For each stand the extraction has been carried out longitudinally, in correspondence of the bottom gorge and the intermediate circumferential sections, where the thermal analysis has then been done.

In order to make automatic the extraction, a specific Abaqus-Phyton Script has been created.

4.3.2.3 Friction Model

The contact between the pipe and the mandrel occurs under pipe plasticization in presence of lubricant composed by a mixture of glass and borax. In these conditions is necessary to model the friction following appropriate laws. The validity of these laws has been calibrated on the *tribometer* present at the R&D Centre of TenarisDalmine. These laws have been introduced in the FE model of the rolling process and in a Friction Model, with the aim to obtain contact shear stress components from contact pressures. A brief description of involved principles is presented.

In plastic deformation processes, contact pressure present is in general high and one of two bodies in contact undergoes an intense plastic deformation. Is observed empirically that increasing contact pressure (p), initially contact shear stress (τ) increases too: plastic deformation is confined to the contact asperities and the materials in contact are subjected to a *macroscopic elastic deformation*. Increasing again contact pressure, contact shear stress reaches a limit value (τ_{lim}) after which remains constant: the extension of real contact areas has reached a limit value in which there is an intense superficial plastic deformation and contact is *fully plastic*. Possible presence of lubricant causes a friction coefficient reduction (μ); this action deals with both the two macro areas:

- in the macroscopic elastic deformation region the lubricant affects on friction coefficient value:

$$\mu = \mu(\text{lubricant}) \qquad (4\text{-}25)$$

- in the fully plastic deformation region the lubricant affects on shear stress limit value:

$$\tau_{\lim} = m * K \qquad (4\text{-}26)$$

where

- m is a coefficient function of lubricant present:

$$m = m(\text{lubricant}) \qquad (4\text{-}27)$$

- K is the limit shear stress of material subject on plastic deformation.

Figure 4-12 and Figure 4-13 show respectively the difference between elastic and plastic contact and the effect of lubricant on the maximum shear stress transmissible. Values present in figures are only indicative and are not referred to the case analyzed in the present work.

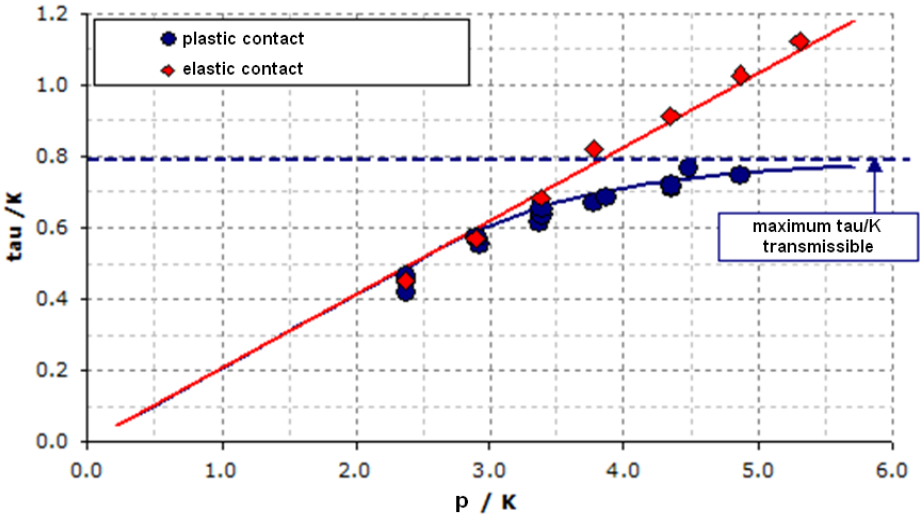

Figure 4-12: Friction Model - Elastic and Plastic Contact Region

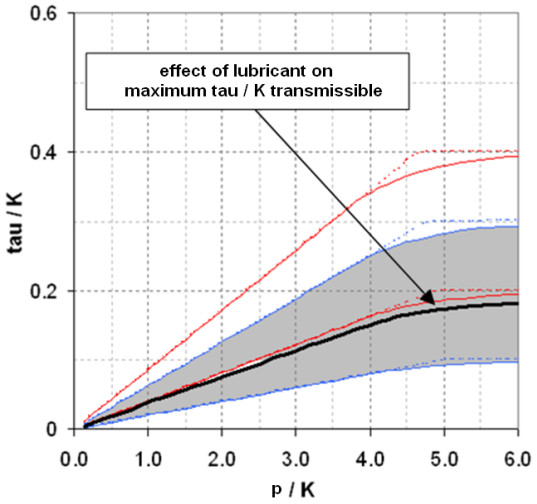

Figure 4-13: Friction Model - Effect of Lubricant on Maximum Transmissible Shear Stress

Following these laws and using appropriated coefficient values from literature, the friction model determines under each stand the average contact shear stress present, based on contact pressure distribution.

4.3.2.4 Thermal Model

The Thermal Model determines the evolution of the mandrel temperature during the rolling process and its hardness fail at the end of the process. The model has been calibrated on the hardness fail measured on mandrels used in the production.

As explained, in the preset work the mandrel temperature is evaluated on the following three longitudinal sections of the mandrel:

- *bottom gorge section* of *even* stands;
- *bottom gorge section* of *odd* stands;
- *intermediate section*.

Each section crosses all stands of the mandrel mill during rolling phase. At each section, the peak temperature of the mandrel is obtained. Hence, the peak temperature reached by the mandrel is determined by the *average* of peak temperatures reached on the three circumferential sections considered.

Concerning the history evolution of the mandrel temperature, the rolling process is divided in single phases, when conditions remain the same. At each phase thermal equations referred to sections of interest are solved, applying its (constant) boundary conditions. At each stand, phase involved on the rolling process are the following:

- *under-stand phase*: in this phase the mandrel is under the roll and the thickness reduction occurs. Boundary conditions applied on the thermal model are the following:
  - *contact under-stand*: contact between the pipe and the mandrel;
  - *friction*: friction between the pipe and the mandrel;
- *inter-stand phase*: in this phase the mandrel is between two subsequent rolls. Boundary conditions applied on the thermal model are the following:
  - *contact inter-stand*: partial contact between the pipe and the mandrel;
  - *radiation*: radiation between the pipe and the mandrel.

This paragraph includes a brief description of the model, explained in [**21**].

The evolution of temperature

The equation that describes the *heat conduction* in an isotropic material in *cartesian coordinates* is the following:

$$\frac{1}{\alpha}\frac{\partial T}{\partial t} = \nabla^2 T + \frac{H}{k} \qquad \text{(4-28)}$$

where:

- T is the *temperature*;
- k is the *thermal conductivity* of material;
- k is the *thermal diffusivity* of material;
- t is the *time*;
- H is the *internal power generation per unit volume.*

In *cylindrical polar coordinates* the equation begins the following:

$$\frac{1}{\alpha}\frac{\partial T}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 T}{\partial \theta^2} + \frac{\partial^2 T}{\partial z^2} + \frac{H}{k} \qquad (4\text{-}29)$$

where

- r is the *radial coordinate*;
- θ is the *circumferential coordinate*;
- z is the *third coordinate.*

Figure 4-14 shows a cylindrical polar coordinates system:

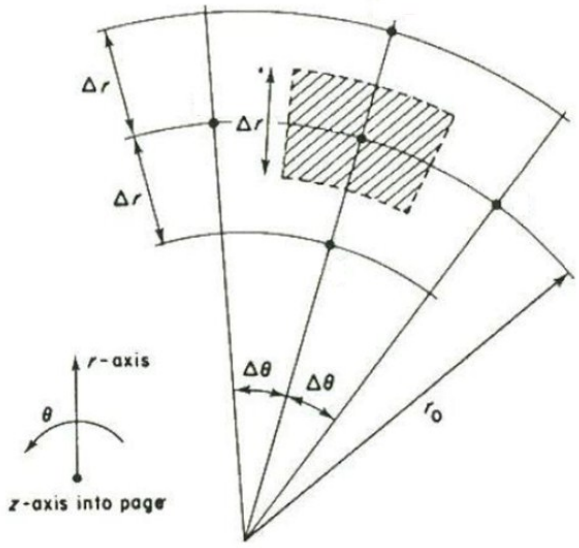

Figure 4-14: Cylindrical Polar Coordinate System

The thermal analysis is made under the assumption that the temperature distribution depends only on the radial coordinate, hence the equation of problem is the following:

$$\frac{1}{\alpha}\frac{\partial T}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right) + \frac{H}{k} \qquad (4\text{-}30)$$

Using a *central finite difference spatial* discretization scheme and an *explicit time integration* scheme, the equation of problem becomes the following:

$$\frac{1}{\alpha}\frac{T_{(k,j)} - T_{(k,j-1)}}{\Delta t} = \frac{T_{(k-1,j-1)} - 2T_{(k,j-1)} + T_{(k+1,j-1)}}{\Delta r^2} + \frac{T_{(k-1,j-1)} + T_{(k+1,j-1)}}{2r_0\Delta r} + \frac{H}{k} \qquad (4\text{-}31)$$

where:

- $T_{(k,j)}$ is the temperature on *k-th node* and *j-th instant;*
- $T_{(k+1,j)}$ is the adjacent inner node to *k-th node* and *j-th instant;*
- $T_{(k-1,j)}$ is the adjacent outer node to *k-th node* and *j-th instant.*

Figure 4-15 shows the described cylindrical polar coordinates grid:

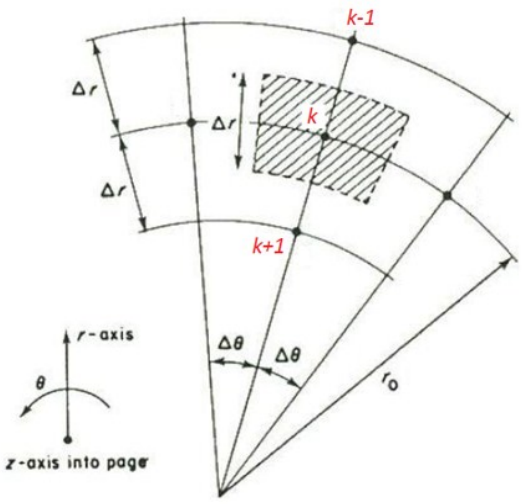

Figure 4-15: Cylindrical Polar Coordinates – Regular Grid

The high gradient present on the external surface of the mandrel requires an introduction of an irregular discretization grid. Hence, the equation of the problem becomes the following:

$$\frac{1}{\alpha}\frac{T_{(k,j)}-T_{(k,j-1)}}{\Delta t}=\left[T_{(k-1,j-1)}\left(\frac{2}{s_e\left(s_e+s_w\right)\Delta r^2}+\frac{1}{r_0\Delta r}\right)-\frac{T_{(k,j-1)}}{s_e s_w\Delta r^2}+\right.$$

$$\left.+T_{(k+1,j-1)}\left(\frac{2}{s_e\left(s_e+s_w\right)\Delta r^2}-\frac{1}{r_0\Delta r}\right)\right]+\frac{H}{k} \qquad (4\text{-}32)$$

where:

- s_e is an outer grid scale coefficient [0÷1];
- s_w is an inner grid scale coefficient [0÷1];
- $s_e+s_w=1$

Figure 4-16 shows the described irregular cylindrical polar coordinates grid:

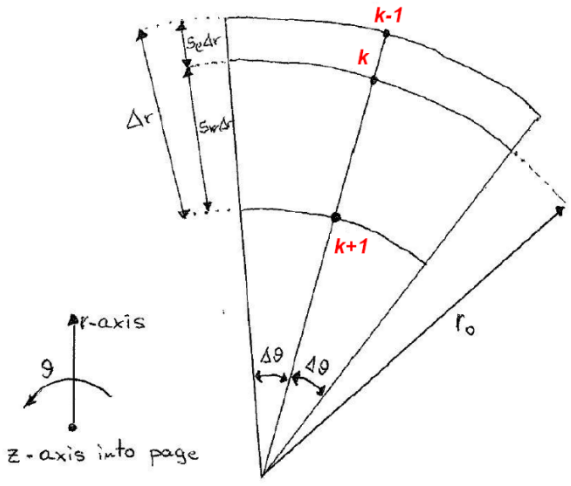

Figure 4-16: Cylindrical Polar Coordinates – Irregular Grid

Concerning boundary treatment, initially the method referred on cartesian system coordinates is presented. The equation of problem solved on the *k-th* boundary node is the following:

$$T_{(k,j)} = 2Fo\left[T_{(k+1,j-1)} + Bi \cdot T_{AMB} + T_{(k,j-1)}\left(\frac{1}{2Fo} - 1 - Bi\right)\right] \quad \text{(4-33)}$$

where:

- T_{AMB} is the temperature of ambient*;*
- *Fo* is *Fourier number*:

$$Fo = \frac{k\,\Delta t}{\rho\, c\,(\Delta r)^2} \quad \text{(4-34)}$$

- *Bi* is *Biot number*:

$$Bi = \frac{h\,\Delta r}{\mathrm{k}} \quad \text{(4-35)}$$

where *h* is the convective heat transfer coefficient.

Radiation and *Conduction* boundary conditions are considered as a convection, using an *equivalent heat convection coefficient*.

Friction boundary condition is applied separately, by adding to the thermal equation of the boundary node a temperature variation. This variation is evaluated by an energy balance at the boundary, imposing on it the friction thermal flux.

Hence, the equation solved on the boundary node with cylindrical polar coordinate system and irregular grid is the following:

$$T_{(k,j)} = 2\,Fo\left[\left(1 - \frac{DR_{(k)}}{D_{EST}}\right)T_{(k+1,j-1)} + Bi_{EST}\left(1 - \frac{DR_{(k)}}{D_{EST}}\right)T_{EST} + \right.$$

$$\left. + \left(\frac{1}{2\,Fo} - \left(1 - \frac{DR_{(k)}}{D_{EST}}\right) - Bi_{EST}\left(1 + \frac{DR_{(k)}}{D_{EST}}\right)\right)T_{(k,j-1)}\right] + \Delta T_{FR} \cdot ABS_{FR} \quad \text{(4-36)}$$

where:

- $DR_{(k)}$ is the size of the grid on *k-th* node;
- D_{EST} is the diameter of the mandrel;
- Bi_{EST} is the Biot number calculated on the boundary *k-th* node;
- T_{EST} is the external temperature;
- ΔT_{FR} is the increasing of temperature due to friction;
- ABS_{FR} is a coefficient concerning a repartition of the energy friction between inner and outer regions.

Hardness fail

The hardness fail is evaluated by the *Dorn law*:

$$Dp = N_{CYC} \int_{0}^{END_{CYCLE}} e^{-\frac{Q_{ACT}}{RT}} dt \qquad (4\text{-}37)$$

where:

- Dp is the *Dorn Parameter* concerning the *hardness fail;*
- N_{CYC} is the number of rolling cycles considered in the hardness fail evaluation;
- END_{CYC} is the end time of the single rolling phase simulated;
- Q_{ACT} is the activation energy for *tempering* phenomena activation;
- R is the *universal constant of perfect gases*;
- T is the instant temperature of the mandrel;
- t is the time.

4.3.3 Optimization

This paragraph concerns a description of the optimization process.

4.3.3.1 Flowchart building

The optimization has been carried out by the use of the commercial software *Matlab*, available at the R&D Centre of TenarisDalmine S.p.A.. Initially has been necessary to automate the entire process, making the following activities:

- building of Matlab scripts for the connection of each used tool;
- development of Matlab scripts for the transfer of data to the cluster for the FE analysis;
- building of an Abaqus Script for the extraction of contact pressures between the pipe and the mandrel from FE analysis results.

Figure 4-17 shows the flowchart of involved variables and each tool used to their evaluation.

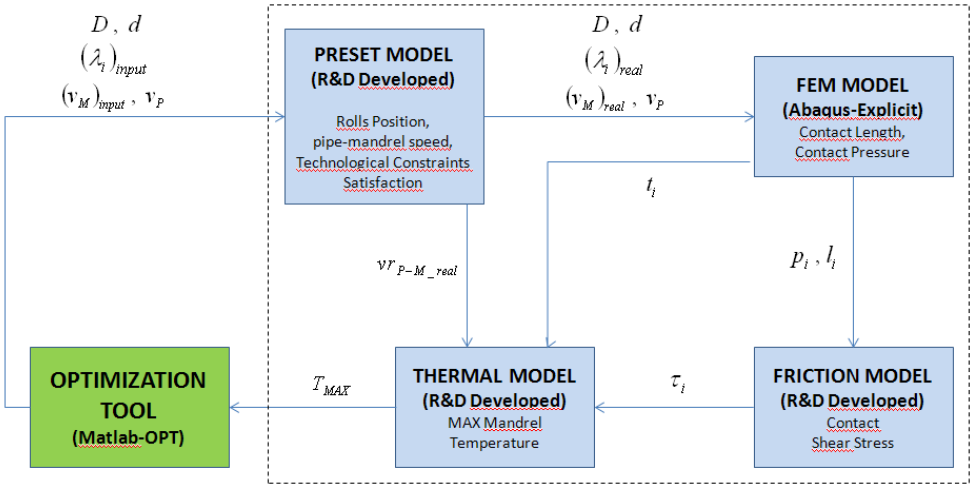

Figure 4-17: Flowchart of Variables and Used Tools

4.3.3.2 Design of Experiment (DOE)

The first phase of the optimization has been a *Design of Experiment* (*DOE*), with the aim to identify the set of most influential parameters on the objective function. First of all, the range of each single parameter has been defined, based on technological and engineering aspects explained in 4.3.1, referring to the case analyzed. The effect of each parameter has been investigated considering every possible combination of variables, with a *full factorial* scheme, with two levels for each parameter and three levels for the elongation. In sequence, configurations investigated are the following:

- inter-stand distance (D):
  - I. $D_{MIN} = D_{REFERENCE\ CASE}$;
  - II. $D_{MAX} = 1.1 \times D_{REFERENCE\ CASE}$;
- roll diameter of stand 4 (d):
  - I. $d_{MIN} = d_{REFERENCE\ CASE}$;
  - II. $d_{MAX} = 1.2 \times d_{REFERENCE\ CASE}$;
- pipe outlet speed (v_P):
  - I. $v_{P\ MIN} = 0.8 \times v_{P\ REFERENCE\ CASE}$;
  - II. $v_{P\ MAX} = 1.2 \times v_{P\ REFERENCE\ CASE}$;
- mandrel speed(v_M) - limit values vary in respect of the pipe outlet speed, for the constraint present on the first stand:
  - *I.* $v_{M\ MIN}$:
    - $v_{P\ MIN} \rightarrow v_{M\ MIN} = 0.6 \times v_{M\ REFERENCE\ CASE}$
    - $v_{P\ MAX} \rightarrow v_{M\ MIN} = 0.8 \times v_{M\ REFERENCE\ CASE}$;
  - *II.* $v_{M\ MAX}$:
    - $v_{P\ MIN} \rightarrow v_{M\ MAX} = 0.85 \times v_{M\ REFERENCE\ CASE}$
    - $v_{P\ MAX} \rightarrow v_{M\ MAX} = 1.15 \times v_{M\ REFERENCE\ CASE}$;
- elongation distribution (λ):
  - I. λ_{IN}: λ concentrated towards the mandrel mill outlet;
  - II. λ_{FLAT}: λ uniformly distributed along the mandrel mill;
  - III. λ_{OUT}: λ concentrated towards the mandrel mill inlet;

Figure 4-18 shows the three configurations concerning the tested real elongation distribution:

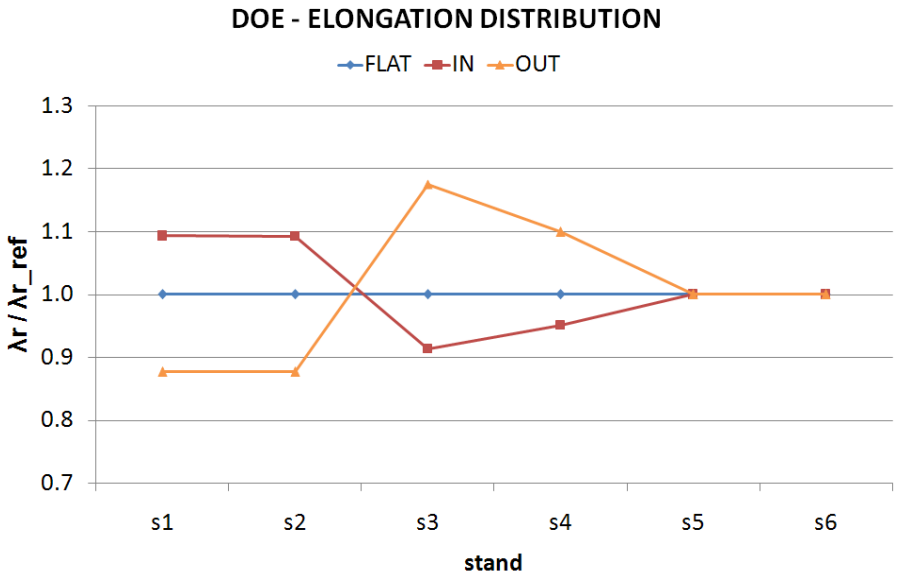

Figure 4-18: DOE - Tested Real Elongation Distribution

Table 4-2 show a resume of tested experiments and the obtained mandrel peak temperature:

CASE	INTERSTAND DISTANCE	S4 ROLL DIAMETER	PIPE OUTLET SPEED	MANDREL SPEED	ELONGATION DISTRIBUITION	T MAX [T MAX / T MAX_REF]
1	1	1	0,8	0,6	OUT	1,18
2					FLAT	1,10
3					IN	1,17
4				0,85	OUT	1,01
5					FLAT	0,95
6					IN	0,99
7			1,2	0,85	OUT	1,26
8					FLAT	1,23
9					IN	1,34
10				1,15	OUT	1,07
11					FLAT	1,03
12					IN	1,12
13		1,2	0,8	0,6	OUT	1,18
14					FLAT	1,10
15					IN	1,17
16				0,85	OUT	1,01
17					FLAT	0,95
18					IN	0,99
19			1,2	0,85	OUT	1,26
20					FLAT	1,23
21					IN	1,34
22				1,15	OUT	1,07
23					FLAT	1,03
24					IN	1,12
37	1,1	1,2	0,8	0,6	OUT	1,17
38					FLAT	1,10
39					IN	1,16
46			1,2	1,15	OUT	1,06
47					FLAT	1,02
48					IN	1,12

Table 4-2: DOE - Tested Cases and Results

Concerning the configuration of maximum inter-stand distance, only cases with the maximum roll diameter on fourth stand and extreme mandrel speeds have been tested.

Inter-stand distance influence

Figure 4-19 shows the influence of the inter-stand distance on the mandrel peak temperature:

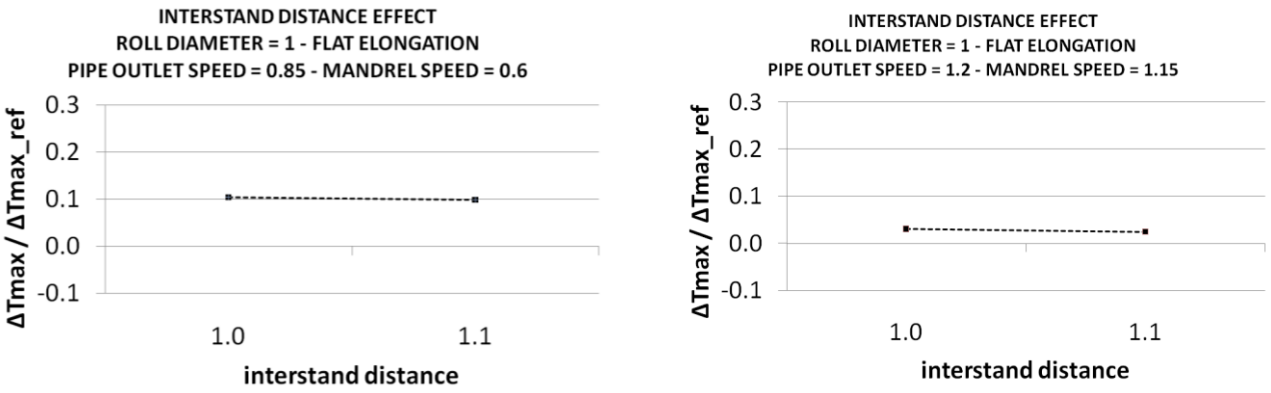

Figure 4-19: DOE – Inter-Stand Distance Influence

Roll diameter of stand 4 influence

Figure 4-20 shows the influence of roll diameter of the fourth stand on the mandrel peak temperature:

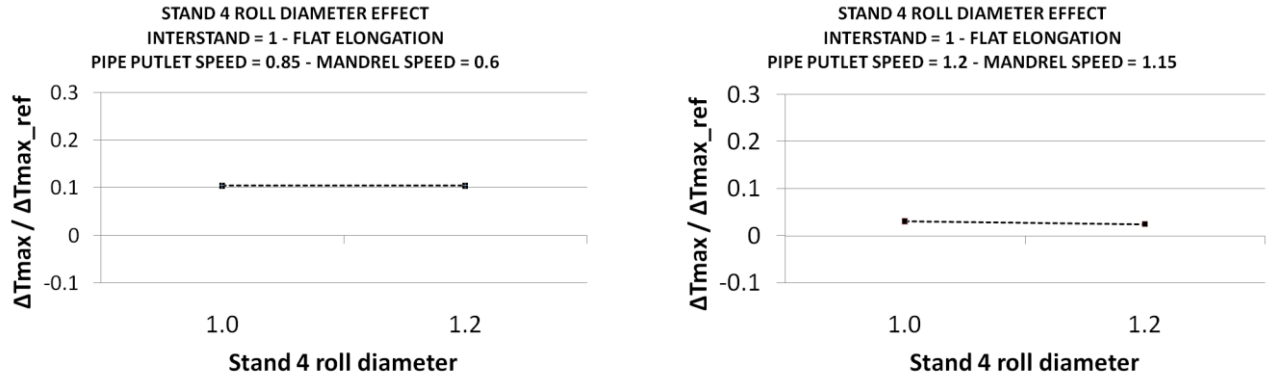

Figure 4-20: DOE – Roll Diameter on Fourth Stand Influence

Pipe outlet speed influence

Figure 4-21 shows the influence of pipe outlet speed on the mandrel peak temperature:

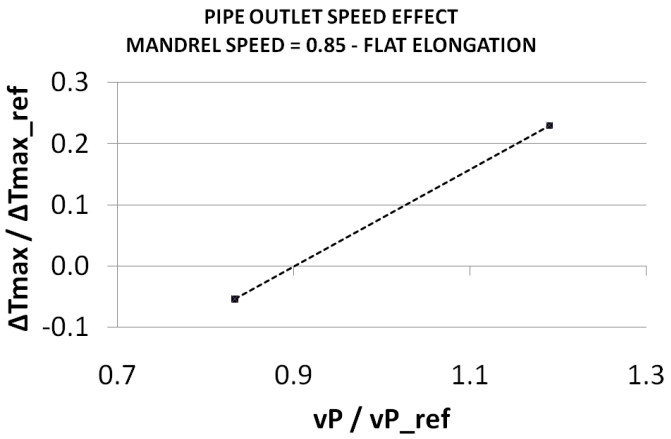

Figure 4-21: DOE - Pipe Outlet Speed Influence

Mandrel speed influence

Figure 4-22 shows the influence of the mandrel speed on the mandrel peak temperature:

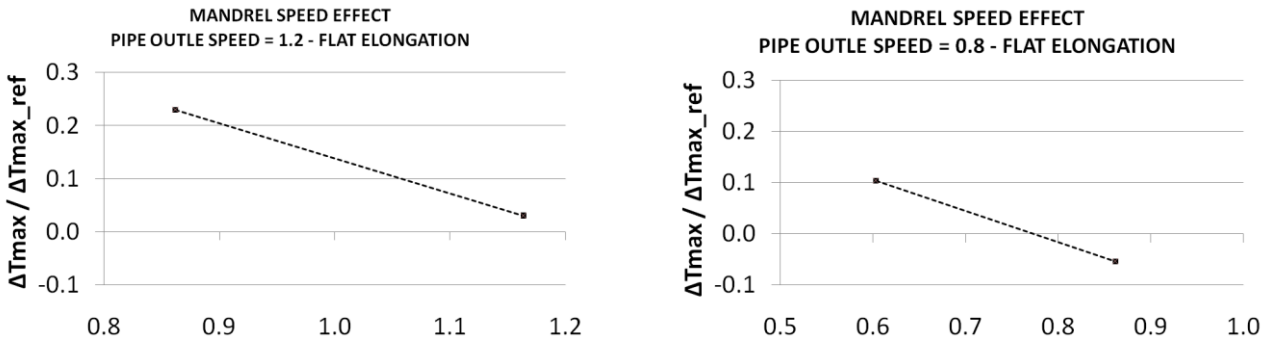

Figure 4-22: DOE - Mandrel Speed Influence

Elongation distribution influence

Figure 4-23 shows the influence of elongation distribution on the mandrel peak temperature:

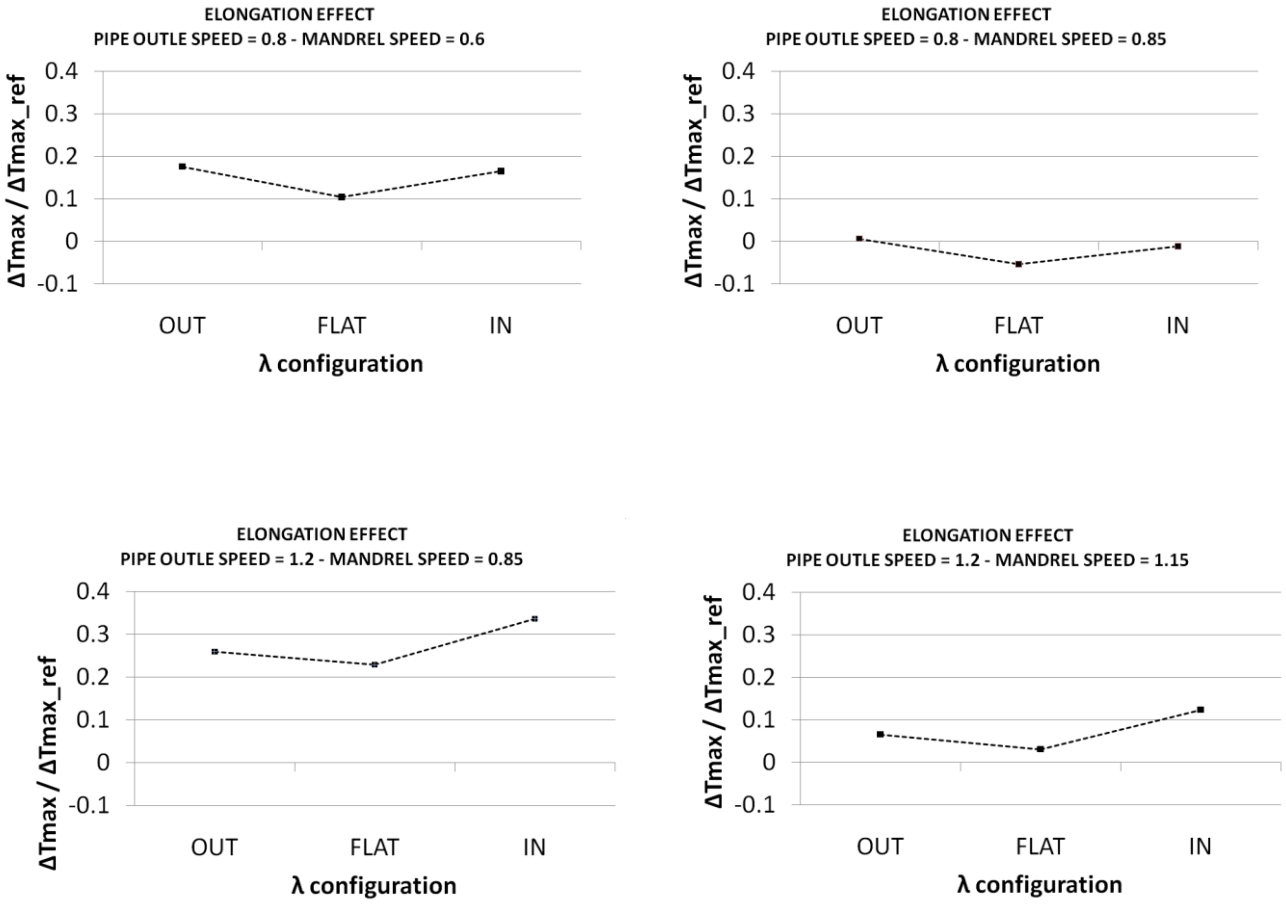

Figure 4-23: DOE: Elongation Distribution Effect

In conclusion, DOE Analysis has shown that pipe outlet speed, mandrel speed and elongation distribution have a significant influence on the peak temperature of the mandrel, while inter-stand distance and roll diameter of the fourth stand have no significant influence on the mandrel peak temperature.

4.3.3.3 Optimization Problem

After the DOE analysis, the optimization has been carried out excluding variables with no relevant influence on the objective function. Not all of the influence variables have been considered: the optimization has been carried out keeping fixed the pipe outlet speed, with the aim to find an optimal configuration corresponding to a determined productivity of the mandrel mill. Furthermore, only elongations of stands 1-2-3-4 have been considered, keeping fixed the elongation of last 5-6 finishing stands.

Hence, the main features of the optimization problem are the following:

- objective: reduction of the peak temperature of the mandrel (T_{MAX});
- influence parameters:
  - elongation of roughing stands (λ_1, λ_2, λ_3, λ_4);
  - mandrel speed (v_M);
- constraints: bound constraint on all variables (Table 4-3)

RANGE OF INPUT PARAMETERS [values/values_REF]			
λ		vM	
MIN	MAX	MIN	MAX
0.7	1.35	0.8	1.1

Table 4-3: Optimization - Range of Input Parameters

As explained previously, constraints on the elongation distribution and on the mandrel speed are considered directly by the Preset Model.

Figure 4-24 shows the flowchart of variables involved in the optimization process:

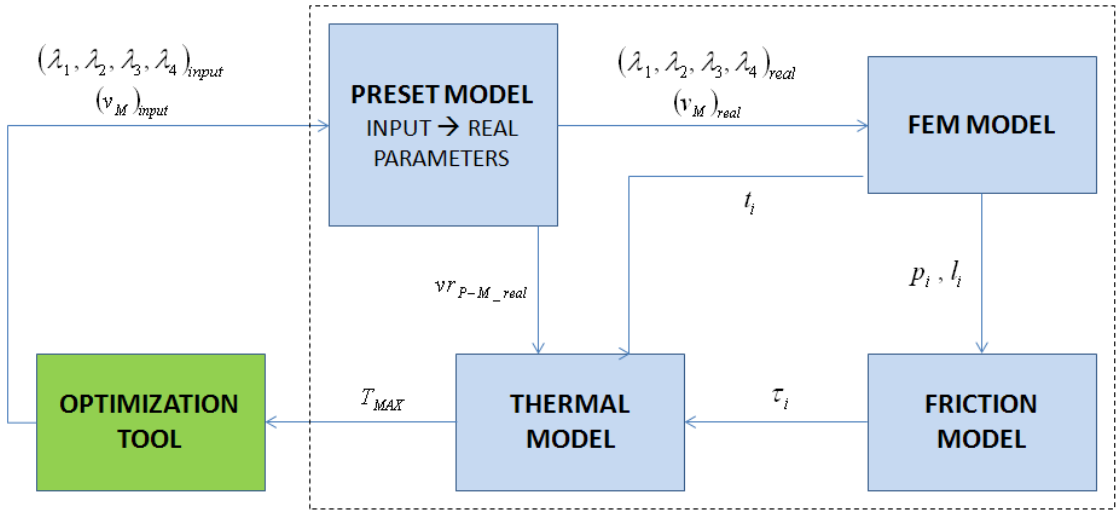

Figure 4-24: Optimization - Flowchart

4.3.3.4 Choice of the Optimization Algorithms

In order to find the best solution, the optimization has been carried out in two steps: a first step, using an appropriate algorithm with the aim to find the global solution, and a second step, with an algorithm adapt to a solution refinement.

The *Matlab Optimization Toolbox* (M-OPT) includes different algorithms: their use is advised in function of the problem under investigation. The case object of the present work has the following main features:

- single objective;
- smooth - non linear objective function;
- bound constraints on all parameters.

Table 4-4 shows a list of algorithms available in the M-OPT:

SINGLE GLOBAL SOLUTION		
FUNCTIONS	SMOOTH OBJECTIVE AND CONSTRAINTS	
SOLVER	GlobalSearch	start point selection
	Multistart	**NO start point selection**
	Patternsearch	free derivative
	GA	**genetic**
	Simulatedannealing	free derivative
MULTIPLE LOCAL SOLUTIONS		
SOLVER	GlobalSearch	start point selection
	Multistart	**NO start point selection**
SINGLE LOCAL SOLUTION		
CONSTRAINTS TYPE	bound	
OBJECTIVE FUNCTION	smooth-nonlinear	
SOLVER	fminbd	1D problem
	fmincon	**general problem**
FMINCON SOLVER		
ALGORITHMS	**interior-point**	**large size - first choise**
	trust-region-reflective	need of gradient
	sqp	small-medium size
	active set	small-medium size

Table 4-4: Matlab Optimization Toolbox – Available Algorithms

As suggested by the Matlab Documentation, the algorithms chosen for the analysis are the following:

- *global solution search*: as shown in Table 4-4, under the section 'single global solution' different algorithms were available. Due to the fact that there were not experience on the behavior of the objective function of interest, the chosen algorithm has been the G*enetic Algorithm*, the only stochastic available.
- *local solution search*: for the refinement phase the algorithm has been chosen with the aim to obtain major information on the objective function behavior on the region under investigation. The *global search* algorithm has been discharged because it makes a selection on the start points, based on obtained objective function. This approach is efficient for the evaluation of a single global solution, but it is not so good for a detailed (complete) analysis of a function around a determined point. Hence, the *Multistart* algorithm has been chosen, because it doesn't make any selection of its start point during the optimization, spanning all the neighborhood of a point. Concerning the solver used in each local solution analysis, the

Interior Point Algorithm has been chosen, as suggested by Matlab Documentation as the first choice to try.

4.3.3.5 Global Solution by Genetic Algorithm

The global solution has been searched by a genetic algorithm (GA).

Algorithm setup

GA has been used exploiting default settings provided by M-OPT. Concerning the present work, the following quantities are mentioned:

- *size of population (number of individuals)*: as explained in [**3**], in order to guarantee the algorithm convergence, a general rule for its determination is the following:

$$N_{GA\_INDIVIDUALS} \geq 16$$
$$N_{GA\_INDIVIDUALS} \geq 2 \cdot \left(N_{OBTV} \cdot N_{PAR}\right) \qquad \text{(4-38)}$$

  where:

  - $N_{GA\_INDIVIDUALS}$ is the number of individuals composing a single population;
  - N_{GA_OBTV} is the number of objective functions of the problem;
  - N_{PAR} is the number of *parameter design* of the problem.

  Concerning the present case, the more strictly condition on population size is that $N_{GA\_POPULATION} \geq 16$. Considering the available CPU time and having not concrete experience about the convergence speed, the default value provided by the M-OPT, equal to 20, has been used;

- *Population composition:* the GA population used is the population provided by default by M_OPT; Table 4-5 shows the population composition:

GA - POPULATION COMPOSITION	
Total Individuals [nr]	20
Elite Individuals [nr]	2
Crossover Individuals [nr]	14
Mutation Individuals [nr]	4

Table 4-5: Opt by GA - Population Composition

- *Other setup parameters:* for details about other setup parameters and their meaning see [**22**].

Using described parameters and considering that each FE analysis has required about 1.2 hours of CPU time, an estimate of the analysis duration is the following:

OPT ANALYSIS - DURATION ESTIMATE	
Generations [nr]	20
Function Evaluations [nr]	420
CPU time [days]	20

Table 4-6: OPT by GA: Estimate of Duration

Optimization – parameters evolution

Figure 4-25 shows the evolution of *input parameters and the peak temperature* during the optimization:

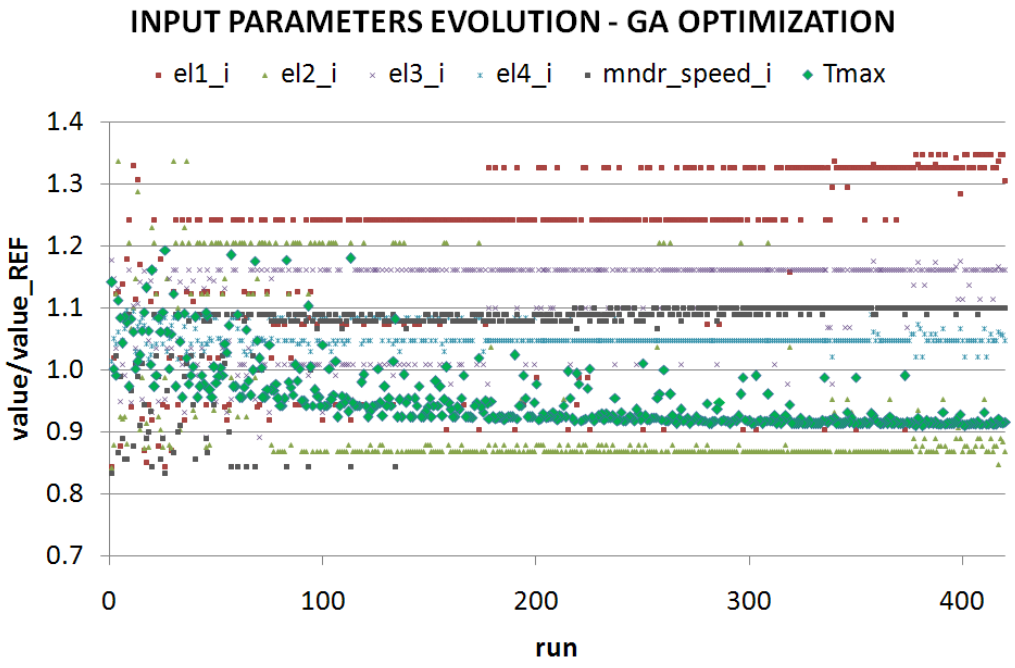

Figure 4-25: OPT by GA: Input Parameter Evolution

Figure 4-26 shows the evolution of *real parameters* after the correction by the preset model *and the peak temperature* during the optimization:

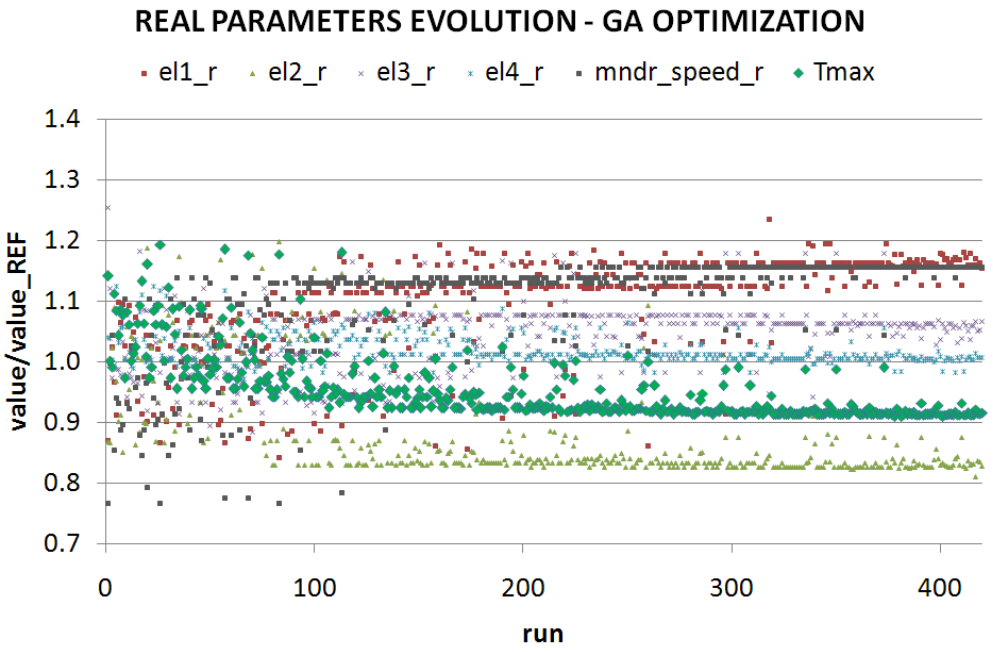

Figure 4-26: OPT by GA: Real Parameter Evolution

Considerations about the parameter evolution are the following:

- after a first phase of *space spanning*, the algorithm has converged towards stable values of involved parameters;
- inside the "*stable value zone*" samples with different value are evident, due to *mutation* operations on individuals;
- the number of used population has been sufficient to reach a stable value of involved parameters.

Figure 4-27 shows the evolution of *real elongation on stand 1 and the peak temperature* during the optimization; the algorithm has been progressively increased the elongation on stand 1:

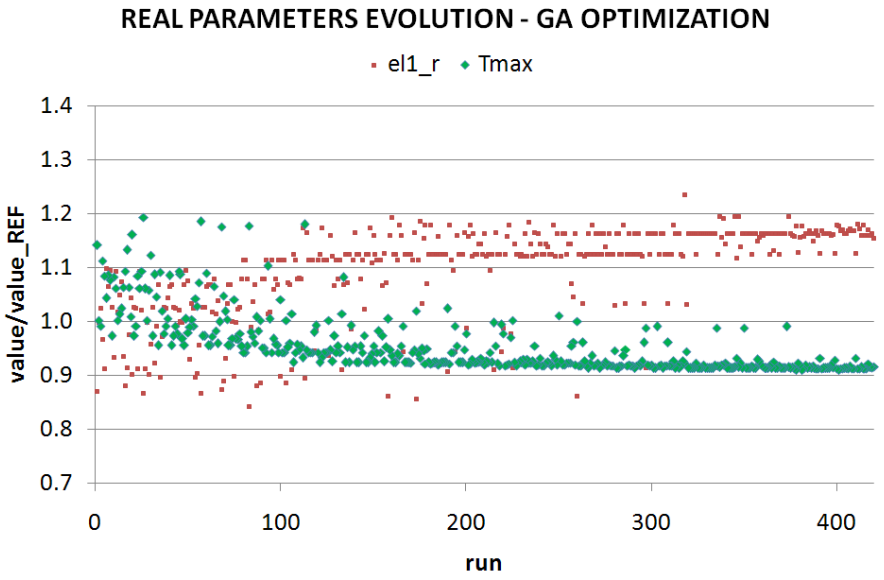

Figure 4-27: OPT by GA: Stand 1 - Real Elongation Evolution

Figure 4-28 shows the evolution of *real elongation on stand 2 and the peak temperature* during the optimization; the algorithm has been progressively decreased the elongation on stand 2:

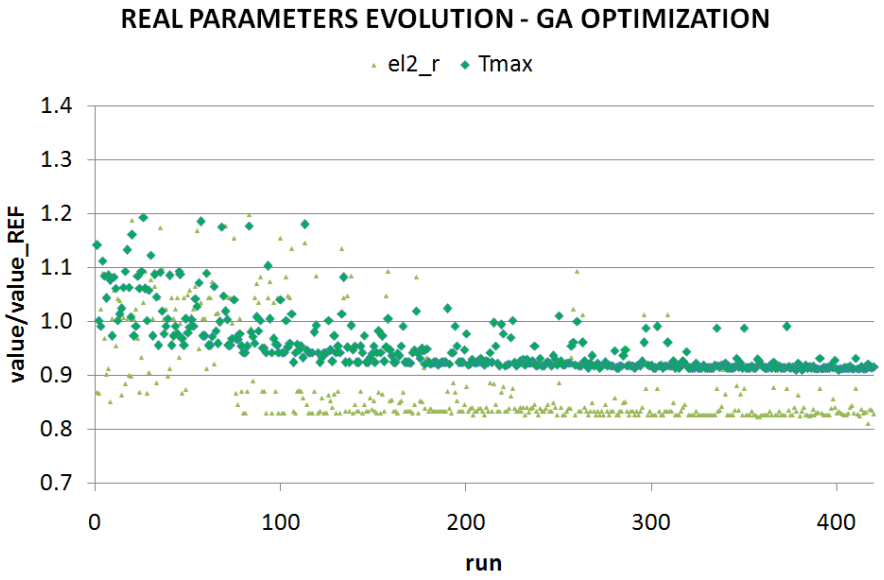

Figure 4-28: OPT by GA: Stand 2 - Real Elongation Evolution

Figure 4-29 shows the evolution of *real elongation on stand 3 and the peak temperature* during the optimization; the algorithm has been progressively increased the elongation on stand 3:

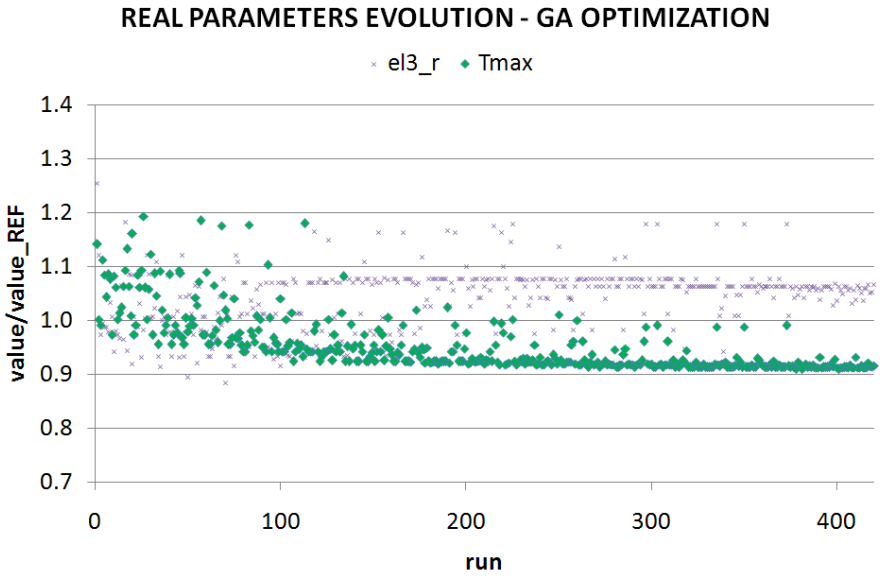

Figure 4-29: OPT by GA: Stand 3 - Real Elongation Evolution

Figure 4-30 shows the evolution of *real elongation on stand 4 and the peak temperature* during the optimization; the algorithm has been progressively increased the elongation on stand 4:

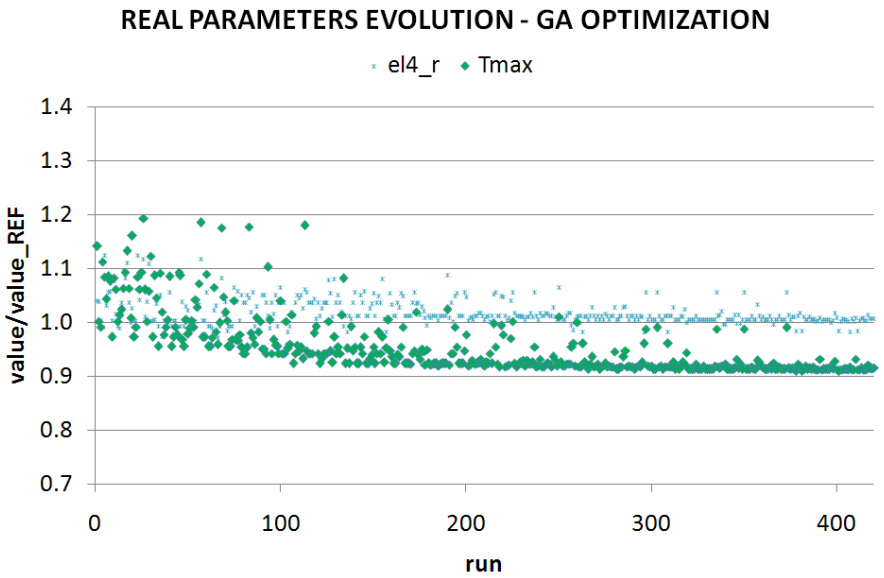

Figure 4-30: OPT by GA: Stand 4 - Real Elongation Evolution

Figure 4-30 shows the evolution of *real mandrel speed and the peak temperature* during the optimization; the algorithm has been progressively increased the mandrel speed:

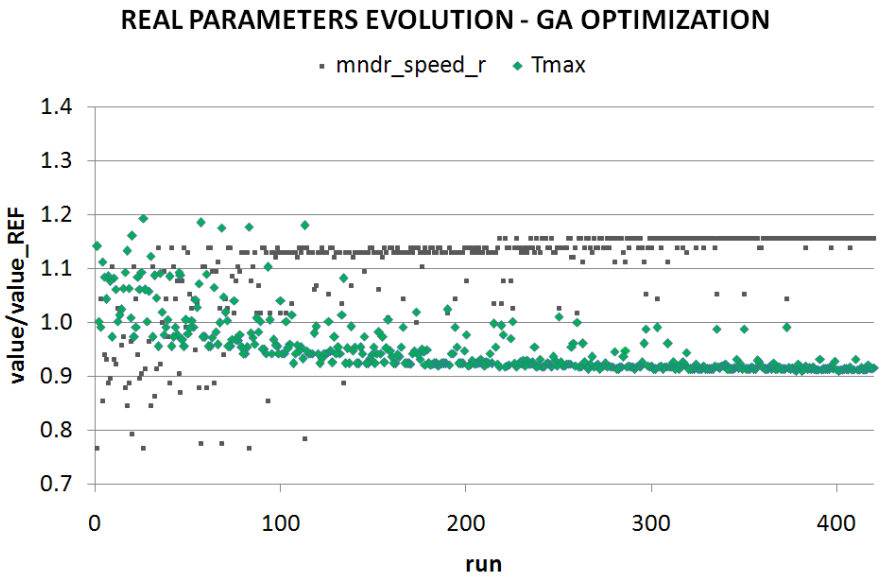

Figure 4-31: OPT by GA: Real Mandrel Speed Evolution

Figure 4-32 shows the evolution of *real mandrel speed and elongation on stand 1* during the optimization; it's evident the correlation between two parameters due to constraint on mandrel speed on stand 1:

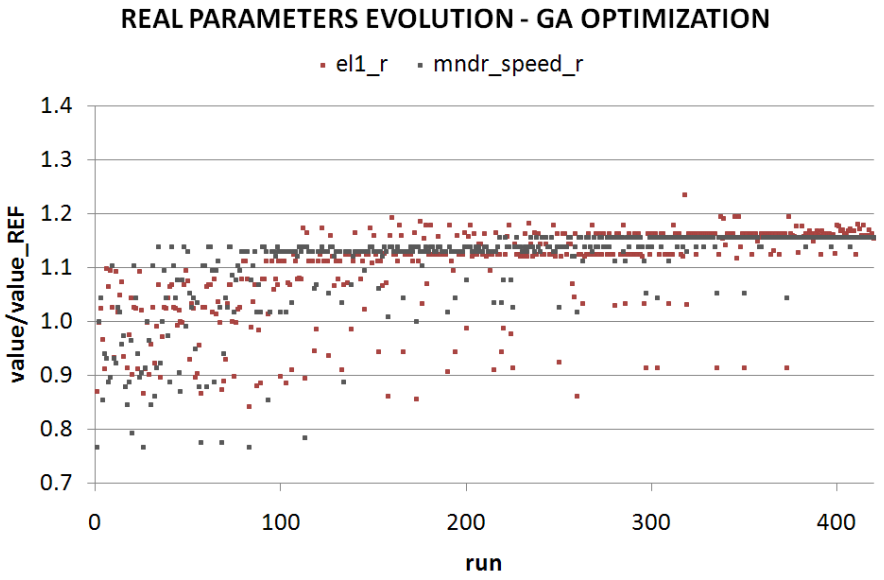

Figure 4-32: OPT by GA: Real Mandrel Speed and Elongation on Stand 1

Global Solution

Figure 4-33and Table 4-7show respectively the found optimal elongation distribution and the optimal value of parameters of the case with the lowest peak temperature of the mandrel.

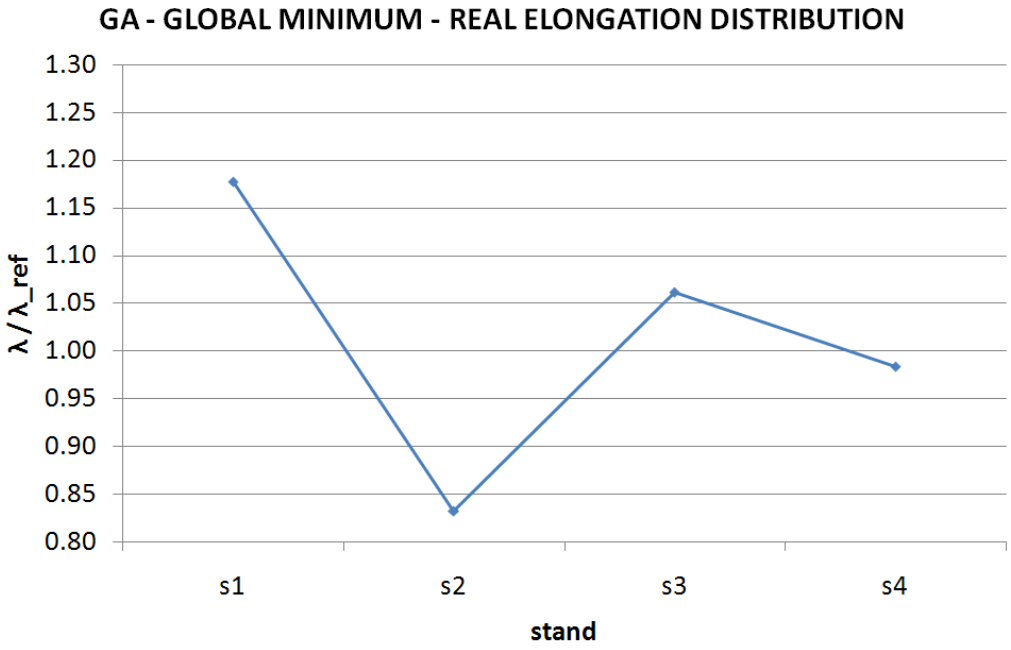

Figure 4-33: OPT GA - Optimal Elongation Distribution

Results respect a REFERENCE CASE - [value/value_ref]						
CASE	λ real				*v*M	T MAX
	s1	s2	s3	s4		
GA	1.18	0.83	1.06	0.98	1.16	0.91

Table 4-7: OPT GA: Global Optimal Solution

The optimal global solution corresponds to a mandrel peak temperature less than 10% in respect of the reference case.

At the end of the present chapter a paragraph devoted on engineering consideration of the optimal solution is present.

4.3.3.6 Building of a Fitting Model

The second phase of the optimization has concerned the realization of a *fitting model,* with the aim to avoid the execution of other FE analysis in the subsequent steps. The main aspects and the procedure concerning the developed model are the following:

- for each stand has been created a fitting model of the following quantities:
  - *contact length*, evaluated at:
    - bottom gorge section;
    - intermediate section;
  - Shear stress component of contact between the pipe and the mandrel, evaluated at:
    - bottom gorge section;
    - intermediate section;
- the fitting model has been created determining all quantities in respect of the *real elongation* on the stand of interest, with the assumption that the elongation on previous stands doesn't influence the parameter value;
- the model created is a *polynomial expansion model*, developed through the commercial software Matlab;
- the fitting model has been developed by fitting of first 200 cases of GA Optimization, cases with a good space spanning;
- for each parameter a polynomial expansion model of the following order has been created:
  - MIN ORDER = 1;
  - MAX ORDER = 20;
- for each order, coefficients of the polynomial expansion has been determined by a *last square minimization* of the fitting error:

$$\arg\min \sum_{i=1}^{N} \left(y_{fit} - y_r \right)^2 \qquad (4\text{-}39)$$

where:

  - y_{fit} is the value of quantity of interest obtained by the fitting model;
  - y_r is the real value of quantity of interest;
  - N is the number of samples used to build the fitting model;
- the fitting order that has been considered *accurate* and, then, used in the following analysis is the lower order corresponding to the following condition:

$$\text{norm2}\left(y_{fit} - y_r \right) \leq tol = 0.05 \qquad (4\text{-}40)$$

where tol is an acceptable tolerance value, equal to 0.05.

Figure 4-34 shows the fitting error of the *contact length on bottom gorge section on stand 1,* in respect of the polynomial order and the real elongation:

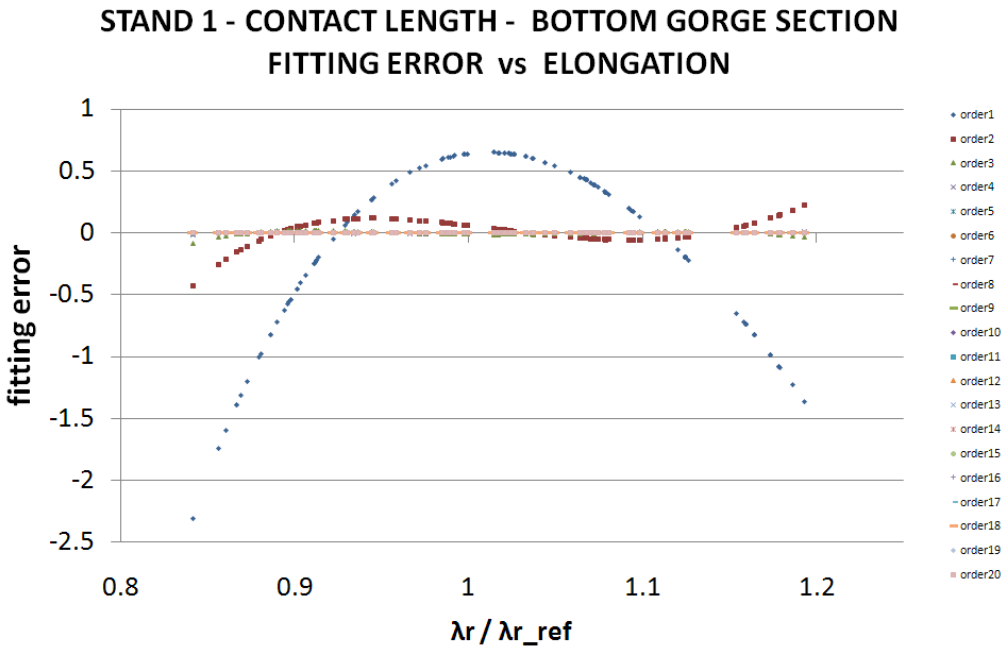

Figure 4-34: Fitting Model - Stand 1 - BG Section - Fitting Error

Figure 4-35 shows the norm2 of the fitting error of the *contact length on bottom gorge section on stand 1* ,in respect of the polynomial order:

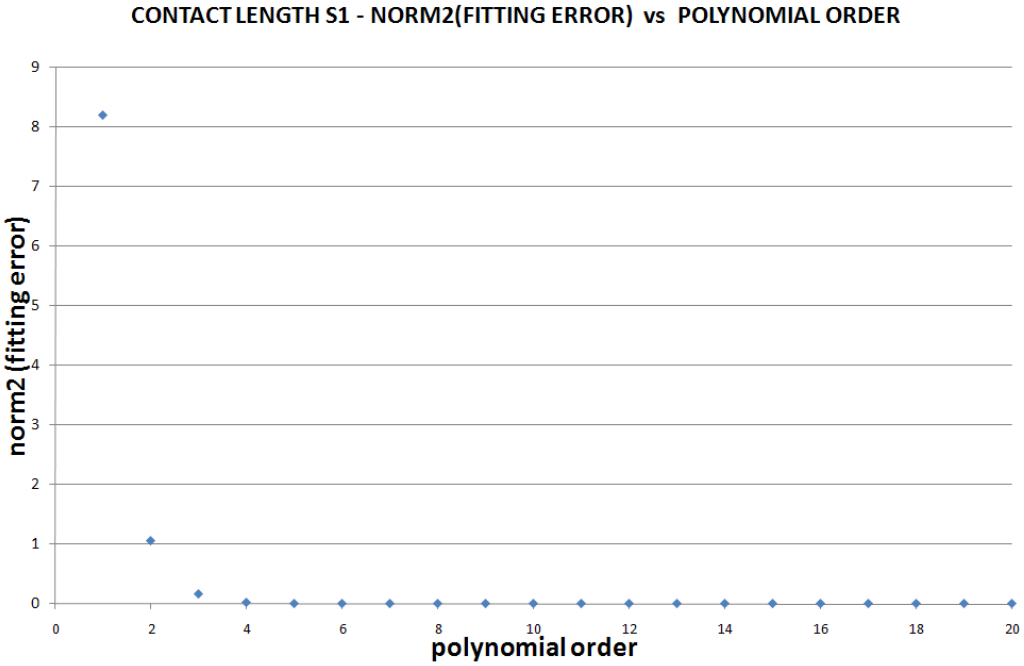

Figure 4-35: Fitting Model - Stand 1 - BG Section – Norm2(fitting error)

As shown in Figure 4-36, for each stand there is a limit on the value range of real elongation in order to avoid boundary divergence of quantities obtained by the fitting model.

Table 4-8 shows the *polynomial order degree* obtained for each quantities and each stands:

	STAND	POLYNOMIAL ORDER
l_cnt_BG	1	4
	2	12
	3	10
	4	9
	5	6
	6	1
l_cnt_IS	1	4
	2	12
	3	10
	4	6
	5	7
	6	1

	STAND	POLYNOMIAL ORDER
tau_BG	1	5
	2	12
	3	10
	4	2
	5	8
	6	7
tau_IS	1	2
	2	12
	3	10
	4	2
	5	6
	6	8

Table 4-8: Fitting model - Polynomial Order Degree Obtained for each Quantities

Figure 4-36 shows the *contact length and contact shear stress on stand 1* used in the analysis, in respect of the real elongation:

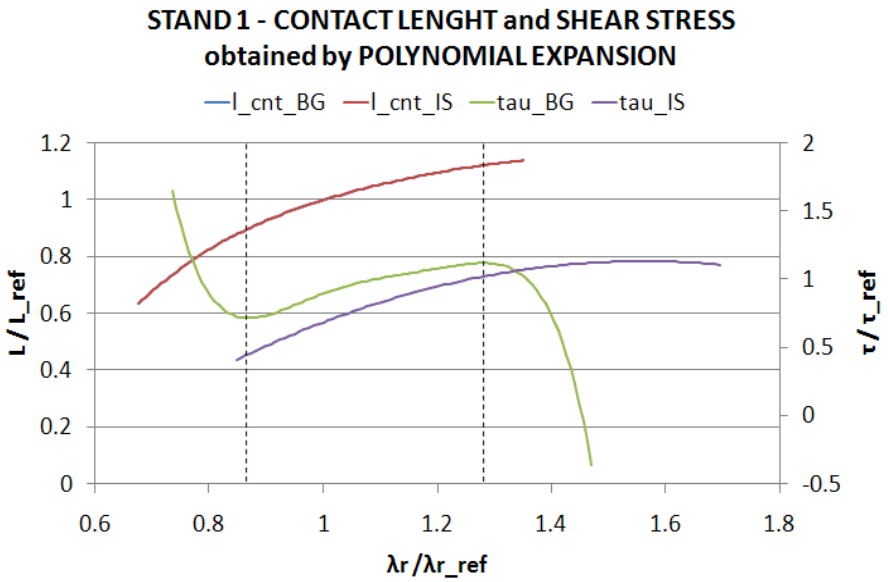

Figure 4-36: Fitting Model - Stand 1 - BG Section – Quantities Behavior

Figure 4-37 shows the *contact length and contact shear stress on each stand* obtained by the fitting model, in respect of the real elongation. The range of the used real elongation must be consistent with plant setting of each stand; further it has been further restricted in order to avoid possible boundary divergence of quantities for high degree polynomial order. Some oscillation of quantities due to used high polynomial degree is evident. The effect of this aspect on the possible generation of local minima on the mandrel peak temperature has not been investigated.

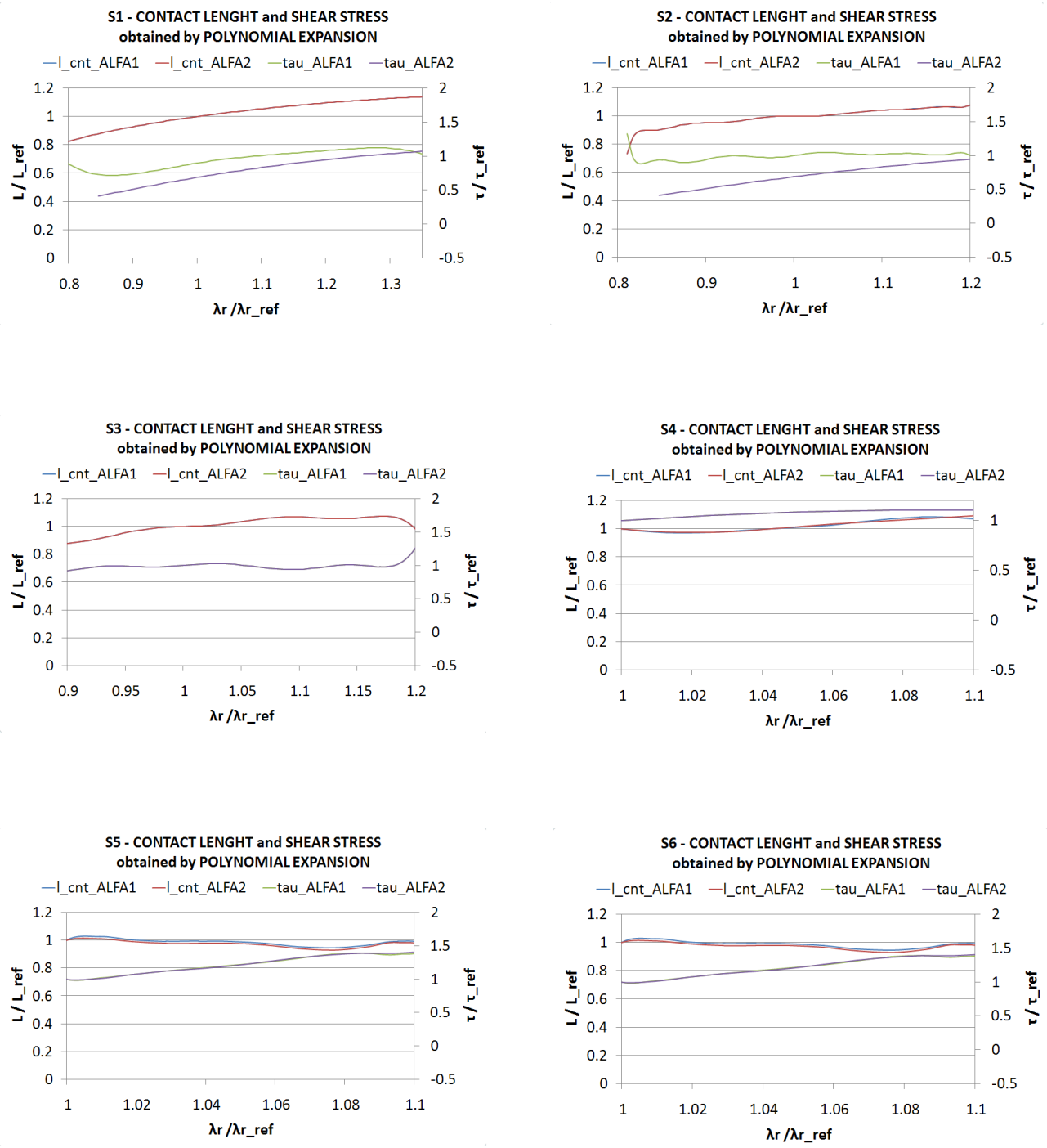

Figure 4-37: Fitting Model - Contact Length and Shear Stress of each Stand

Table 4-9 shows the obtained polynomial expansion coefficients:

	STAND	ORDER	POLYNOMIAL EXPANSION COEFFICIENTS												
			c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13
Icnt BG	1	4	-38.8598	271.2671	-731.338	924.7873	-367.076								
	2	12	-1.9E+09	3.47E+10	-2.9E+11	1.47E+12	-5E+12	1.22E+13	-2.2E+13	2.78E+13	-2.6E+13	1.75E+13	-7.9E+12	2.14E+12	-2.7E+11
	3	10	2.9E+08	-4E+09	2.53E+10	-9.4E+10	2.27E+11	-3.8E+11	4.31E+11	-3.4E+11	1.74E+11	-5.3E+10	7.21E+09		
	4	9	-2.7E+11	3.04E+12	-1.5E+13	4.33E+13	-8E+13	9.91E+13	-8.2E+13	4.31E+13	-1.3E+13	1.82E+12			
	5	6	-7.2E+08	4.77E+09	-1.3E+10	1.93E+10	-1.6E+10	7.04E+09	-1.3E+09						
	6	1	-3.1E-14	32.142											
Icnt IS	1	4	-38.4966	268.7319	-724.504	916.1444	-363.646								
	2	12	-1.4E+09	2.49E+10	-2.1E+11	1.06E+12	-3.6E+12	8.76E+12	-1.5E+13	2E+13	-1.9E+13	1.25E+13	-5.6E+12	1.54E+12	-1.9E+11
	3	10	2.4E+08	-3.4E+09	2.1E+10	-7.8E+10	1.88E+11	-3.1E+11	3.57E+11	-2.8E+11	1.44E+11	-4.4E+10	5.97E+09		
	4	6	-2.1E+07	1.6E+08	-5E+08	8.31E+08	-7.8E+08	3.88E+08	-8.1E+07						
	5	7	-5.8E+09	4.41E+10	-1.4E+11	2.58E+11	-2.8E+11	1.81E+11	-6.5E+10	1E+10					
	6	1	-1.1E-13	24.27894											
Tau BG	1	5	-458.974	3429.618	-10205.4	15108.01	-11115.9	3253.751							
	2	12	4.68E+08	-8.6E+09	7.18E+10	-3.6E+11	1.24E+12	-3E+12	5.3E+12	-6.9E+12	6.44E+12	-4.3E+12	1.93E+12	-5.3E+11	6.55E+10
	3	10	-1.4E+08	2.02E+09	-1.3E+10	4.82E+10	-1.2E+11	1.99E+11	-2.3E+11	1.86E+11	-9.7E+10	3.01E+10	-4.2E+09		
	4	2	-68.6336	176.818	-105.502										
	5	8	3.11E+10	-2.7E+11	1.05E+12	-2.3E+12	3.16E+12	-2.8E+12	1.52E+12	-4.8E+11	6.56E+10				
	6	7	-8.1E+10	5.86E+11	-1.8E+12	3.1E+12	-3.2E+12	1.97E+12	-6.7E+11	9.91E+10					
Tau IS	1	2	-5.81826	21.44968	-13.2814										
	2	12	3.49E+08	-6.4E+09	5.36E+10	-2.7E+11	9.27E+11	-2.2E+12	3.96E+12	-5.1E+12	4.81E+12	-3.2E+12	1.44E+12	-3.9E+11	4.89E+10
	3	10	-1.1E+08	1.62E+09	-1E+10	3.85E+10	-9.5E+10	1.59E+11	-1.9E+11	1.49E+11	-7.8E+10	2.4E+10	-3.3E+09		
	4	2	-54.9069	141.4544	-84.4012										
	5	6	67161226	-4.4E+08	1.23E+09	-1.8E+09	1.49E+09	-6.6E+08	1.2E+08						
	6	8	3.36E+10	-3.8E+11	1.73E+12	-4.3E+12	6.54E+12	-6.2E+12	3.59E+12	-1.2E+12	1.67E+11				

Table 4-9: Fitting Model - Polynomial Coefficients

Table 4-10 shows the error on the mandrel peak temperature obtained by the FE Analysis and the fitting model. Cases present in Table 4-10 are only the main cases used to evaluate the error on the peak temperature of the mandrel.

CASE	λ real						vM	T MAX ERROR
	s1	s2	s3	s4	s5	s6		
1	1.00	0.87	1.13	1.04	1.00	1.00	1.00	-0.001
2	0.87	0.87	1.26	1.04	1.00	1.00	0.77	-0.005
3	1.16	0.83	1.07	1.01	1.00	1.00	1.16	0.003

Table 4-10: Fitting Model – Error on The peak temperature of the mandrel Respect FE Analysis

Figure 4-38 shows the flowchart of variables involved in the optimization process. The time of analysis of each single case has been reduced to 5÷10 minutes, depending on the number of iterative correction made by the Preset Model.

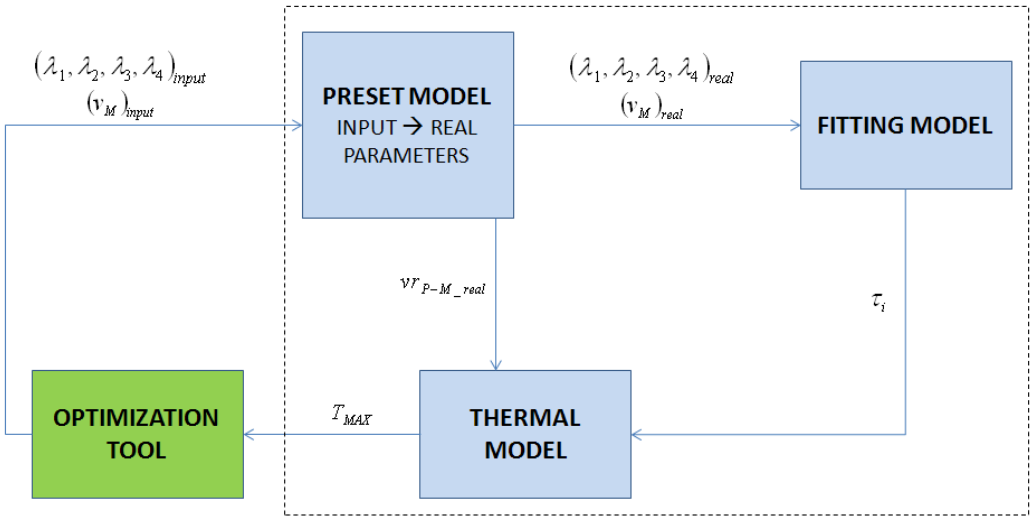

Figure 4-38: Fitting Model - Flowchart

4.3.3.7 Solution Refinement by Gradient Based Algorithm

After the evaluation of the global solution by GA, a second step of the optimization process has been carried out, with the aim to refine the solution and to verify the possible existence of local minima points in a neighborhood of that global solution.

Space restriction

Exploiting the space selection made in the first step by GA, the space investigated for the solution refinement has been reduced. The criteria followed for the restriction of the refinement space region is based on the mandrel peak temperature obtained by GA:

$$T_{MAX\_REFINEMENT} \leq 0.95 \cdot T_{MAX\_REFERENCE\_CASE} \quad \text{(4-41)}$$

where

- $T_{MAX\_REFINEMENT}$ is the higher peak temperature of the mandrel considered in the refinement phase (value referred to the peak temperature obtained during GA optimization);
- $T_{MAX\_RERFERENCE\_CASE}$ is the mandrel peak temperature of the reference case.

Hence, for obvious reasons, the solution refinement step has excluded cases with high values of the mandrel peak temperature. The range of obtained *input parameters* is the following:

[values/values_ref]				
STAND	λ INPUT		*v*M	
	MIN	MAX	MIN	MAX
1	1.05	1.35	1.07	1.11
2	0.84	1.21		
3	0.96	1.18		
4	1.01	1.10		

Table 4-11: Multistart Optimization - Input Parameter Range of Solution Refinement Space

Table 4-11 is internal to the limit of input parameters shown in Figure 4-37. Hence, boundary divergence phenomena are excluded.

Algorithm setup

The Multistart Interior-Point Algorithm (MS-IP) has been used, initially exploiting default settings provided by M-OPT. Unlike GA, a changing of the following parameter has been necessary:

- *Step length*: the gradient of the objective function at a generic point is built evaluating the variation of the objective in respect of each parameter. This variation is computed by means of a *forward finite difference* approach (after that the Hessian is built using the quasi-Newton Method). The entity of variation of each parameter is determined by the *step length* factor:

$$\Delta x_i = step\_length \cdot x_i \qquad (4\text{-}42)$$

where:

- Δx_i is the variation of *i-th* parameter;
- *step_length* is the step length parameter;
- x_i is the value of *i-th* parameter at the current point.

The default value of the step length provided by M-OPT is the following:

$$step\_length_{DEFAULT} = \sqrt{\varepsilon} \qquad (4\text{-}43)$$

where:

- ε is the machine error, equal to 1.49e-08 (referred to the working in double precision by matlab).

The default value of the step length is too small for the problem and the used tools: the Preset Model doesn't change its variable value in presence of very small variation of them. Hence, during a first trial test, parameters didn't change and the optimization has not been started. After the first derivative estimation, M-OPT has stopped the process at a final point equal to the start point.

As suggested by M-DOC, a relaxation of the step length has been necessary. Its value has been progressively increased: from the reference case, a series of single gradient based trials has been made, increasing progressively the step length, until a variation of design variables during the process.

The minimum step length able to vary all involved parameters has been equal to 0.01.

Figure 4-39 shows the evolution of *input parameters* and *the peak temperature of the mandrel* during a gradient base descent from the reference case, with the step length equal to 0.01.

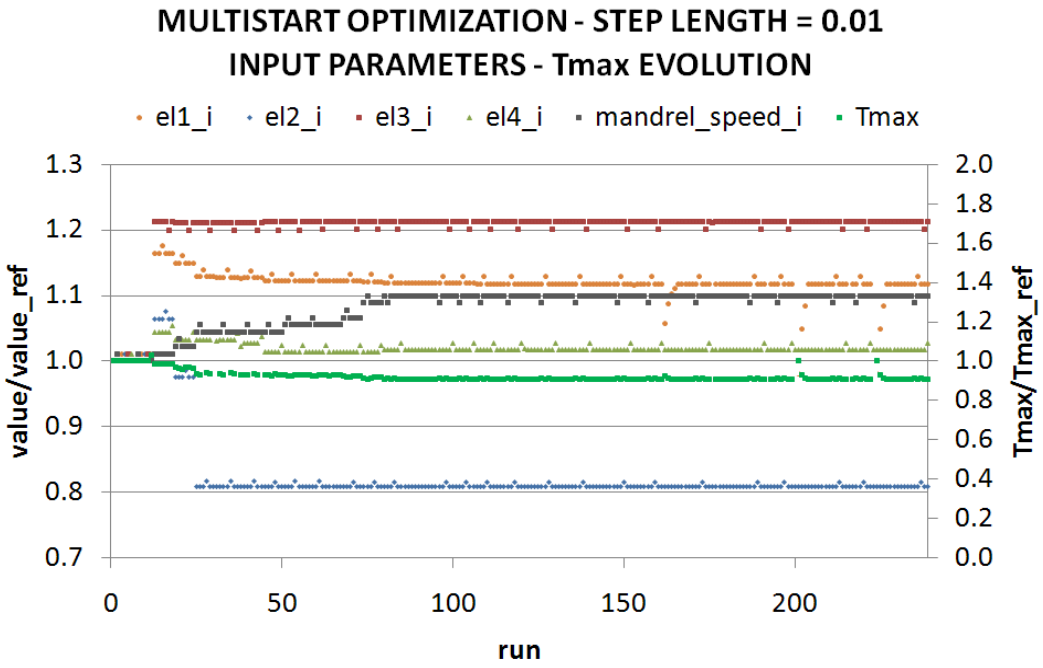

Figure 4-39: Gradient Based Optimization - Step Length = 0.01

Start points

The aspects involved in the start point definitions are the following:

- *number of start* points: the number of start points of each single gradient descend has been chosen considering the time available and the time necessary to make a single descend. The time necessary to carry out a single analysis has been assumed equal to the time of descend of the test with the step length equal to 0.01. That analysis has required 240 function evaluation and a CPU time of 12 hours. Considering a serial computing (a parallel analysis has not been performed) and fixing a total time of all the analysis equal to one week, a number of start points equal to 10 has been chosen;
- *space subdivision*: in order to investigate uniformly the space, the default setting of MS-IP has been used, with the aim to cover the largest number of combinations of involved parameters.

Figure 4-40 and Figure 4-41 show respectively the *input elongation* and *input mandrel speed* of the obtained 10 start points:

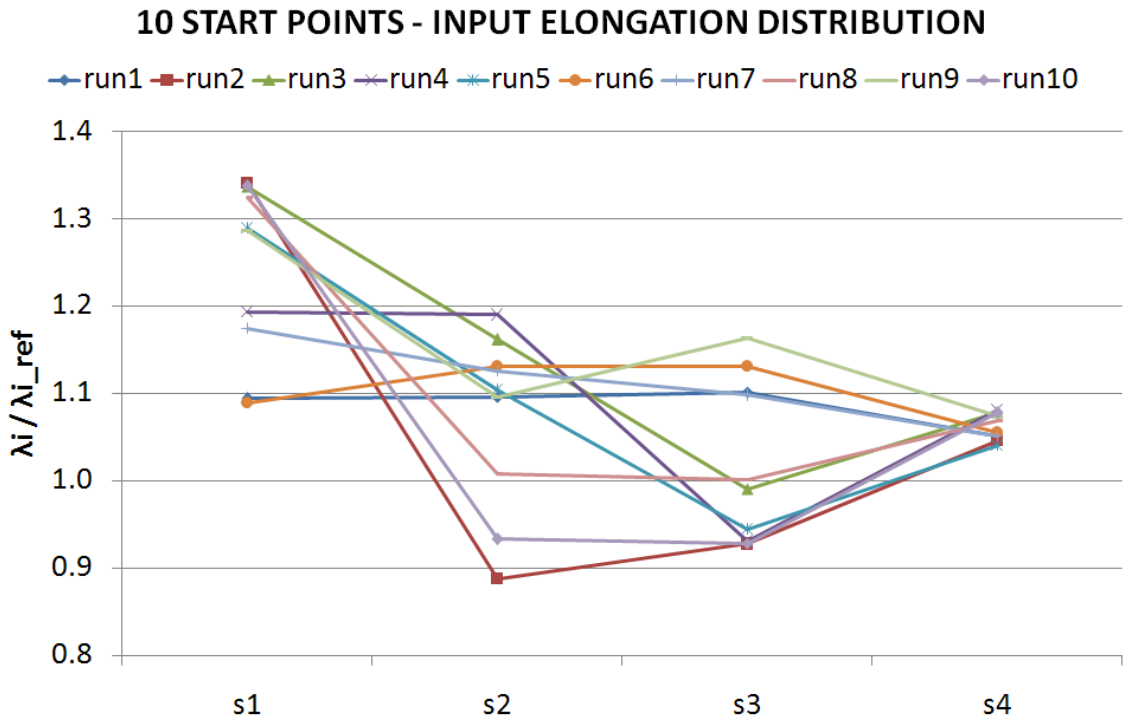

Figure 4-40: Multistart Optimization - Start Points - Input Elongation

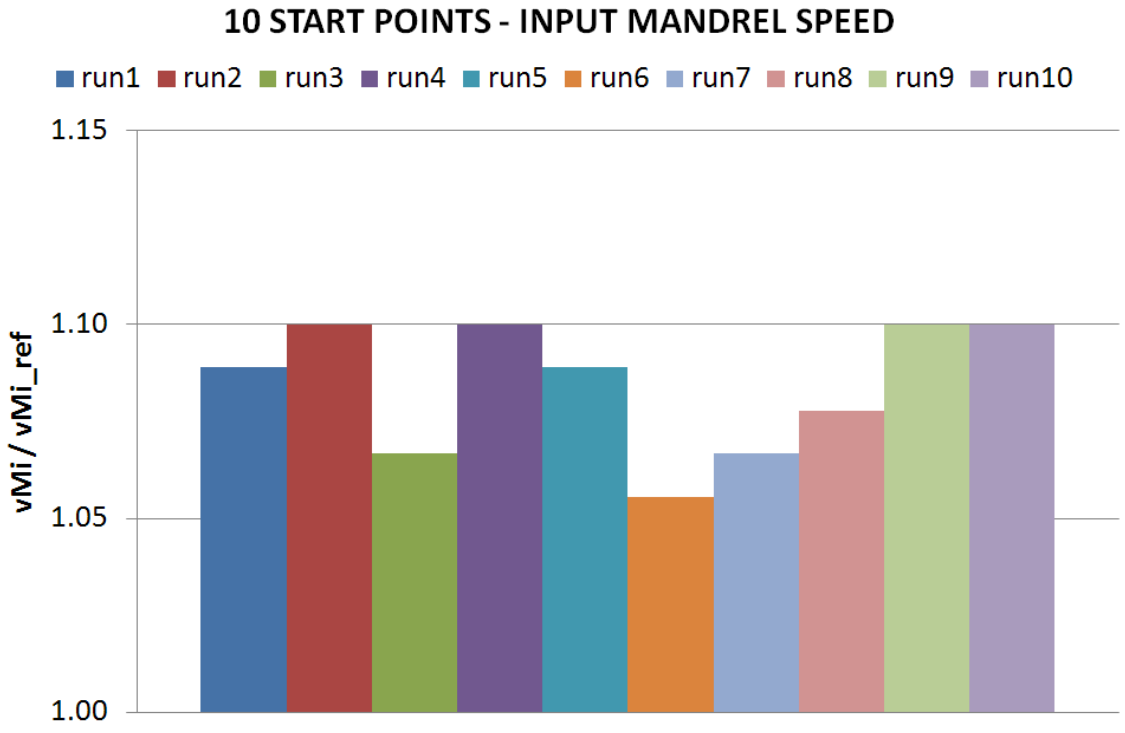

Figure 4-41: Multistart Optimization - Start Points - Input Mandrel Speed

Figure 4-42 and Figure 4-43 show respectively the *real elongation* and *real mandrel speed* of obtained 10 start points:

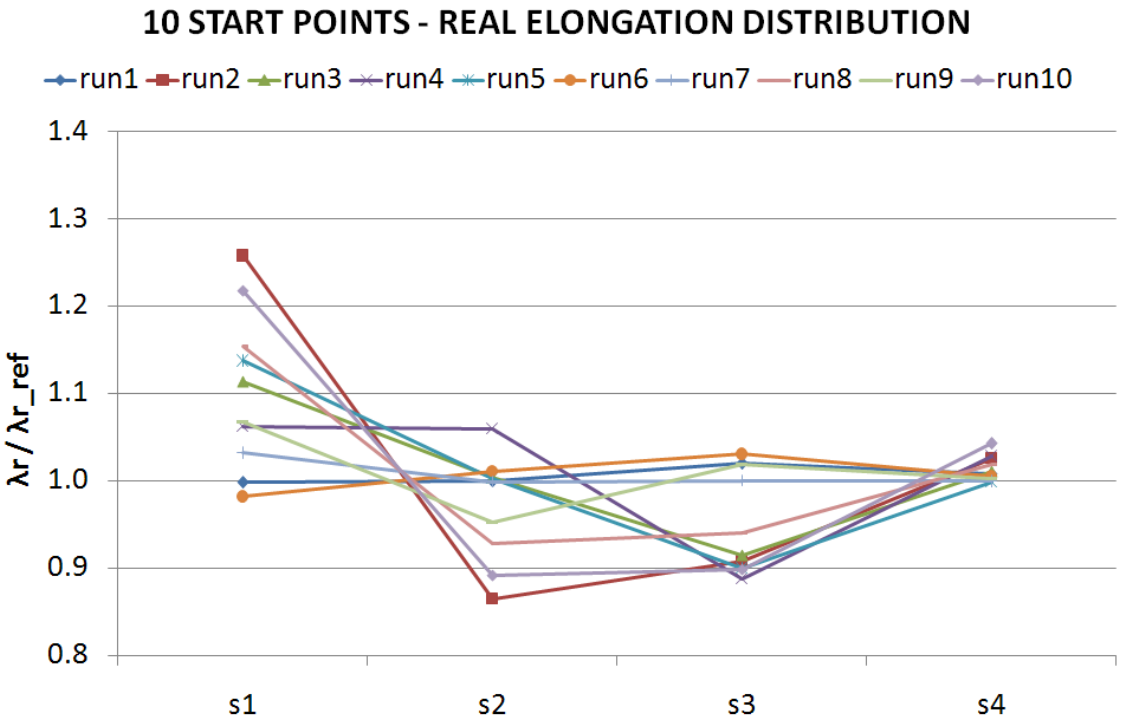

Figure 4-42: Multistart Optimization - Start Points - Real Elongation

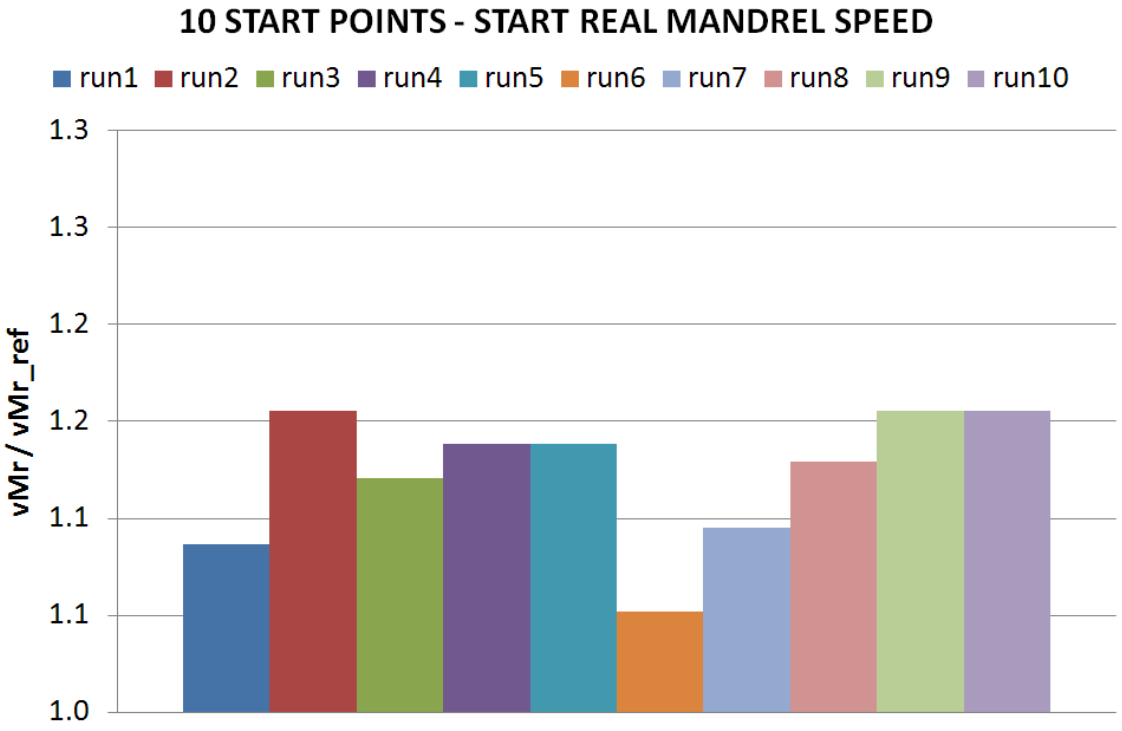

Figure 4-43: Multistart Optimization - Start Points - Real Mandrel Speed

Optimization

The optimization has required 1240 function evaluations and 6 days of CPU time. Different solutions have been found; hence, different local minima are present.

Figure 4-44 shows the evolution of *peak temperature* during the optimization:

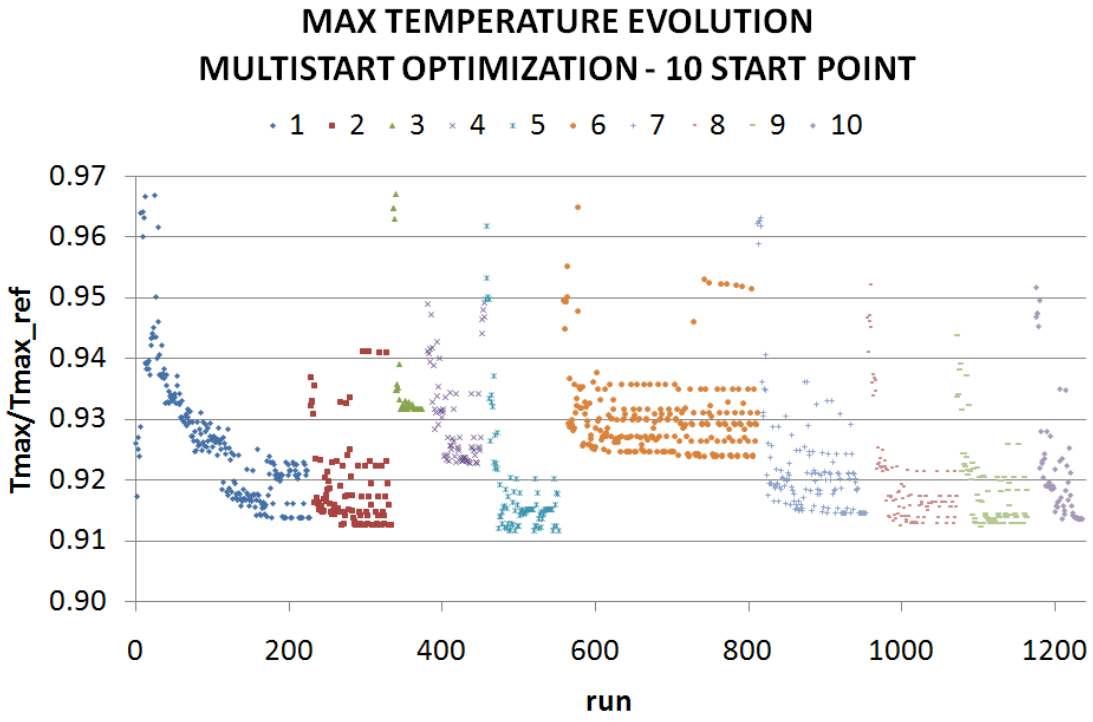

Figure 4-44: Multistart Optimization - The Mandrel Peak Temperature Evolution

Figure 4-45 shows the *real elongation distribution* relative to 10 local minima present:

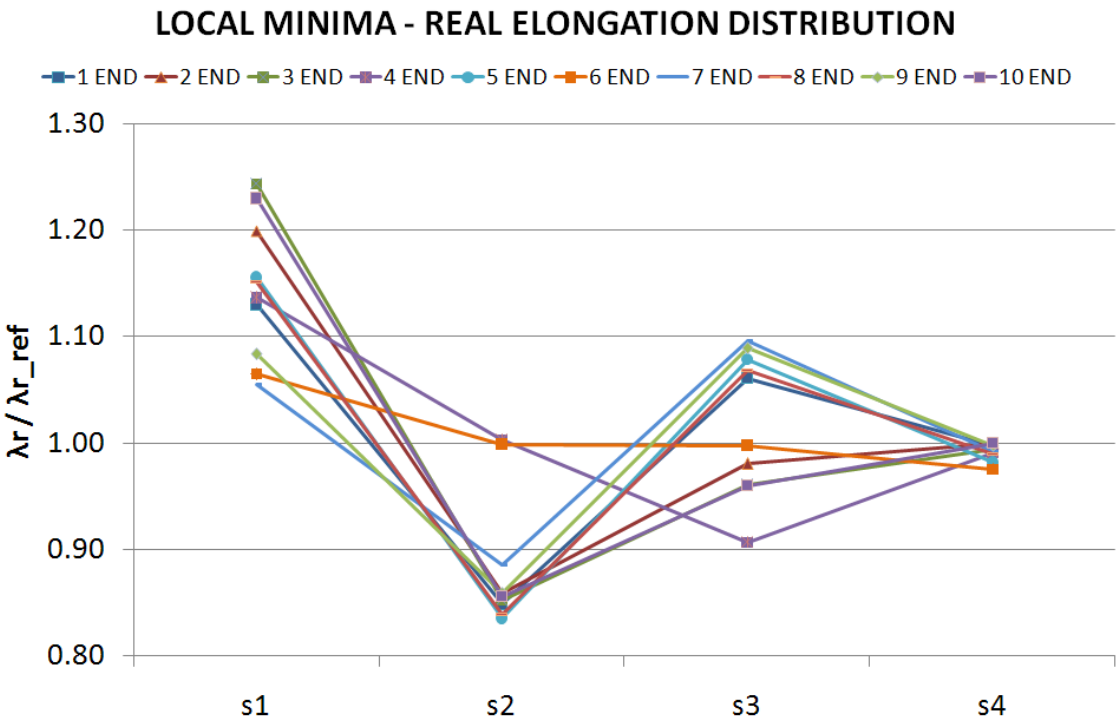

Figure 4-45: Multistart Optimization - 10 Local Minima - Optimal Real Elongation Distribution

Figure 4-4 shows the *real mandrel speed* relative to 10 local minima present:

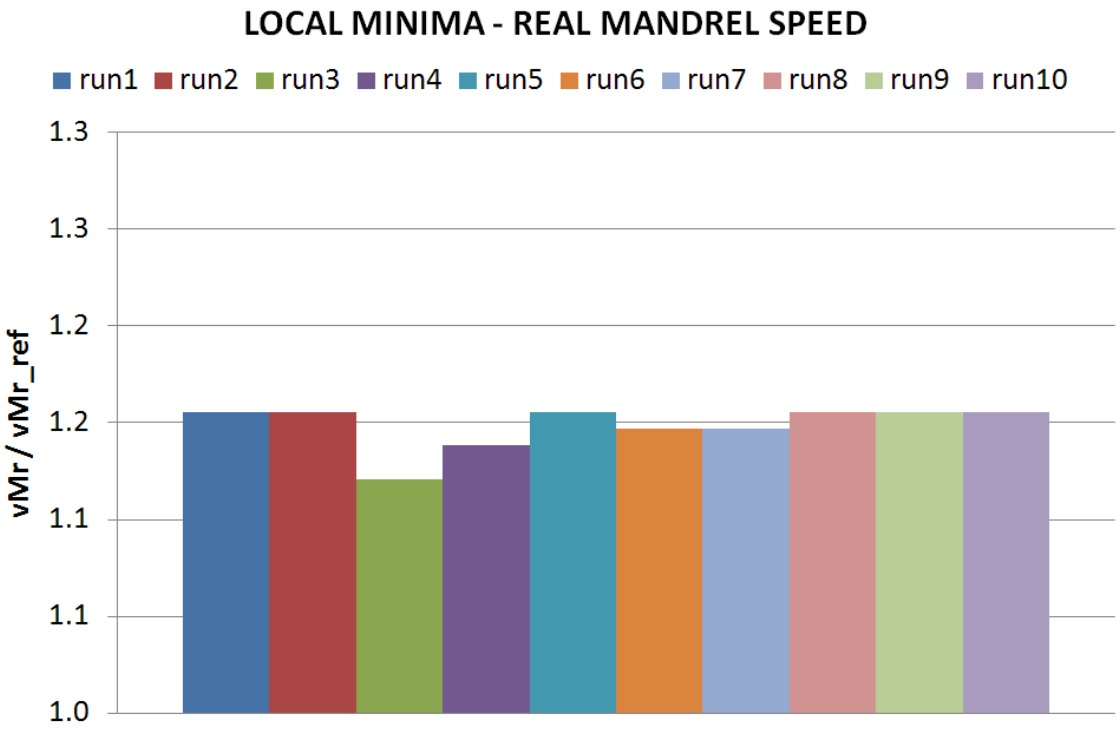

Figure 4-46: Multistart Optimization - 10 Local Minima - Optimal Real Mandrel Speed

Figure 4-47 and Table 4-12 show respectively the *optimal real elongation distribution* and the *optimal value of parameters* of the case with the lowest mandrel peak temperature:

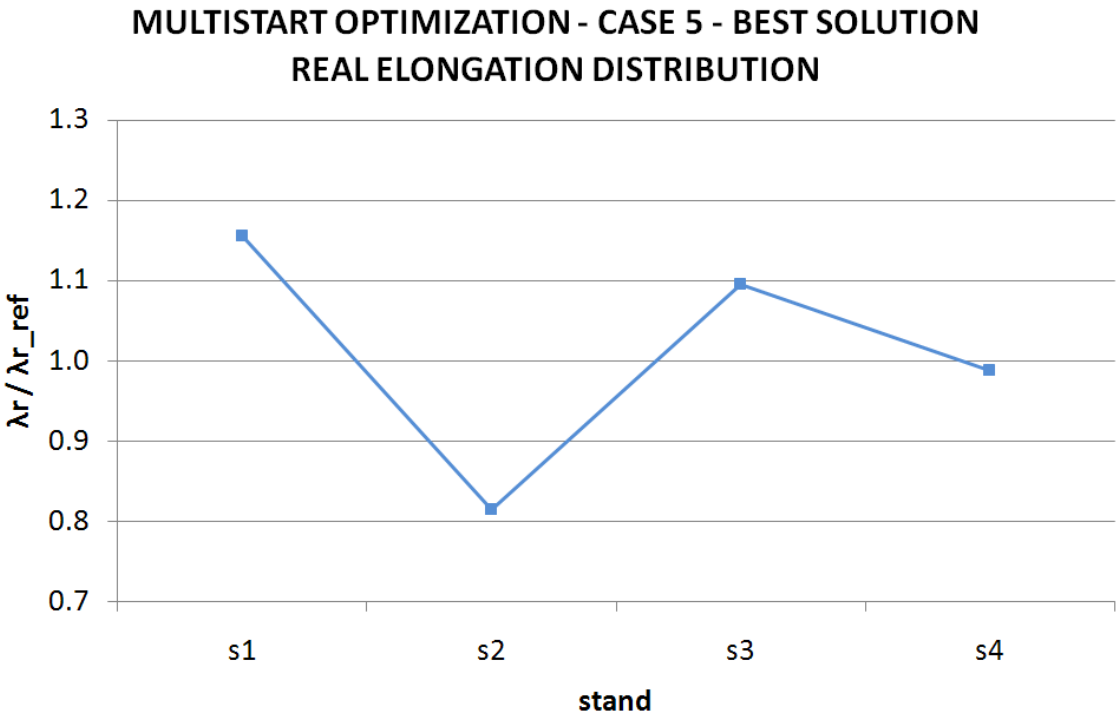

Figure 4-47: Multistart Optimization – case 5 – Best Solution - Real Elongation Distribution

Results respect a REFERENCE CASE - [value/value_ref]						
CASE	λ real				*v*M	T MAX
	s1	s2	s3	s4		
MULTISTART	1.16	0.84	1.08	0.98	1.16	0.91

Table 4-12: Multistart Optimization – Case 5 – Best Solution – Parameters Value

4.4 Results

4.4.1 Solution Comparison

The optimal solution obtained using a gradient based method has been found to be in close agreement with the solution obtained by GA optimization. Figure 4-48 and Table 4-13 show respectively a comparison of the best solution obtained by GA and MS-IP and the optimal value of parameters corresponding to those cases:

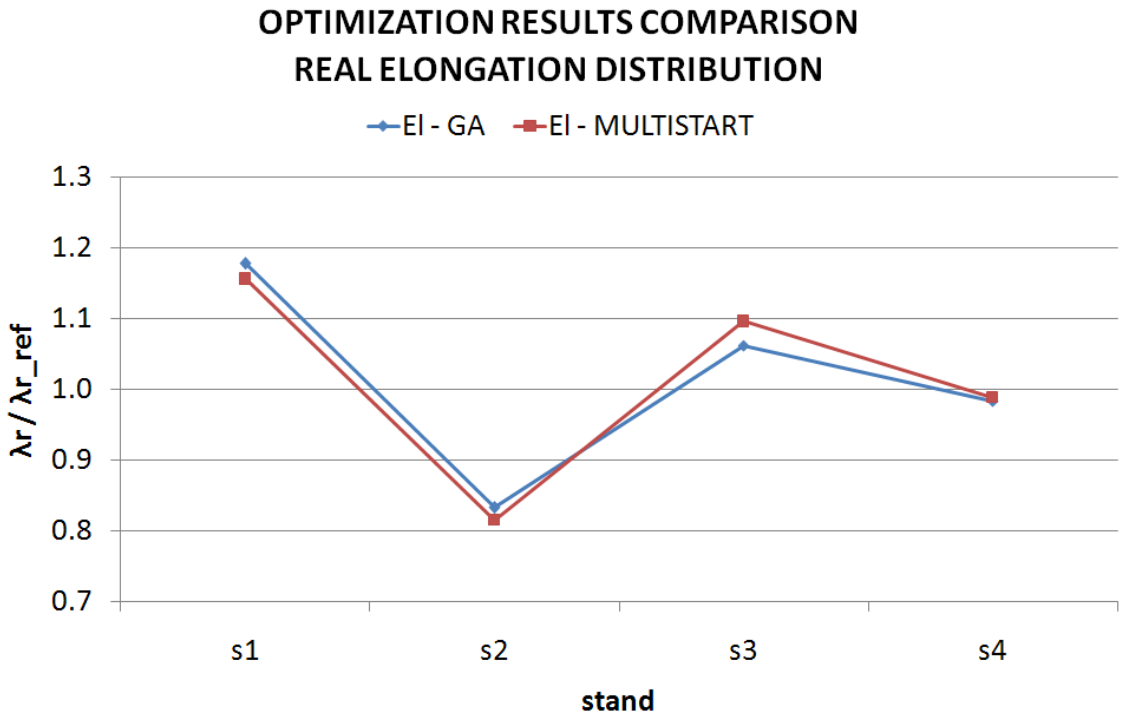

Figure 4-48: GA and MS-IP Optimal Solution – Real Elongation Comparison

Results respect a REFERENCE CASE - [value/value_ref]						
CASE	λ real				*v*M	T MAX
	s1	s2	s3	s4		
GA	1.18	0.83	1.06	0.98	1.16	0.91
MULTISTART	1.16	0.84	1.08	0.98	1.16	0.91

Table 4-13: GA and MS-IP Optimal Solution – Parameters Comparison

As shown, the best solution obtained by two algorithms is in close agreement: there is only a little difference on the elongation distribution but this difference doesn't affect the peak temperature of the mandrel.

4.4.2 Algorithms Comparison

The important aspects concerning the use of two algorithms are the followings:

- GA:
  - 420 function evaluation;
  - NO setting of parameters has been required;
- MS-IP:
  - 1240 function evaluation;
  - setting of the step length parameter has been required.

Since the optimal global solution found by both the algorithms is the same, the conclusion concerning their application on this problem is that GA are more fast and robust than MS-IP.

4.4.3 Engineering Considerations

This paragraph deals with engineering consideration concerning the obtained optimal solution. Figure 4-49 shows these main aspects:

- the optimal solution provides a high elongation value on the first stand: due to the mandrel speed constraint on the first stand, this allows to increase the mandrel speed;
- the low value of the elongation on the second stand is due to the fact that this generates an increasing of the elongation on the third and the forth stand, with the aim to obtain the correct total elongation. This allows to obtain an acceleration of the pipe towards the outlet of the mandrel mill;
- the high mandrel speed and the described elongation distribution allow to reduce the relative speed between the pipe and the mandrel on the mandrel mill. This is the main factor that reduces the peak temperature of the mandrel during the rolling phase;
- the elongation distribution has no relevant impact on contact pressures and on contact shear stress.

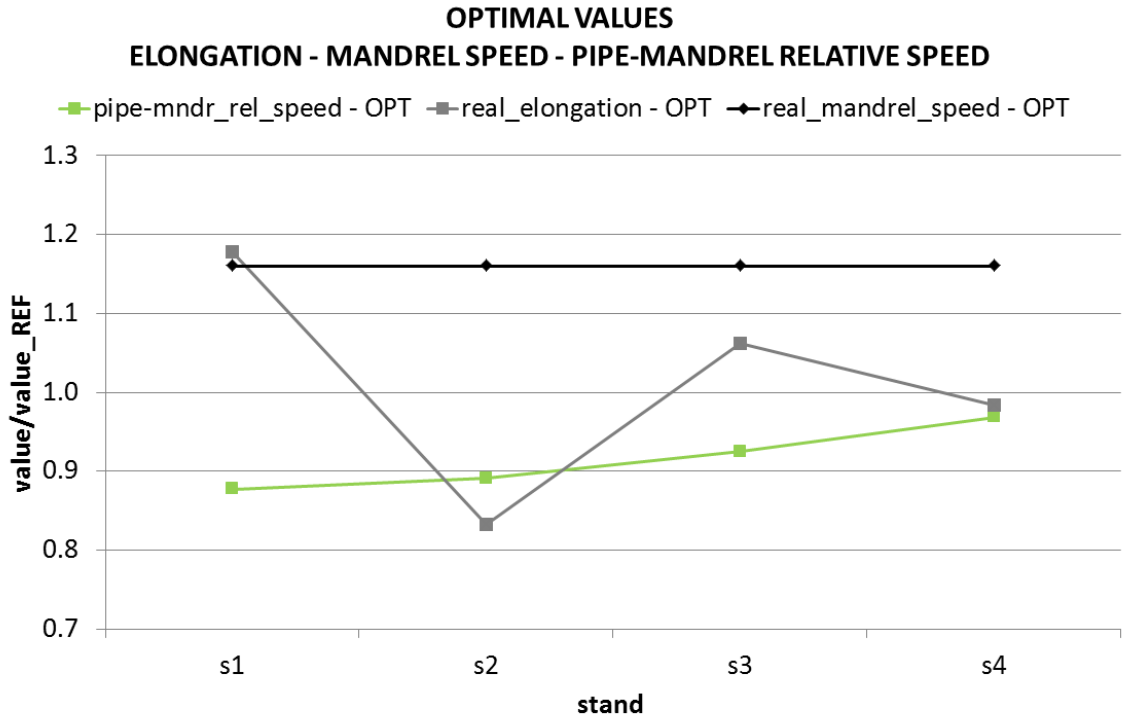

Figure 4-49: Global Optimal Solution – Engineering Considerations

4.4.4 Plant Improvements

This paragraph concerns a description of plant improvements obtainable by the optimal solution:

- *mandrel life improvement*: as explained in 4.2.3, the hardness of the mandrel is an important parameter related to the solidity of basement of the external chromium layer. In this sense, the found optimal solution allows to reduce the *HV hardness fail* of the mandrel during its use on the plant. An estimate carried out by the thermal model, simulating 1000 consecutive rolling cycles, shows that the optimal solution allows a reduction of the HV hardness fail of 15% in respect of the reference case. At the moment it's not possible to relate exactly the hardness reduction with an increasing of the mandrel life, but the obtained optimal configuration is certainly better;
- *productivity improvement*: alternatively, a productivity improvement can be obtained by using the optimal elongation distribution and increasing the pipe outlet speed, keeping fixed the mandrel peak temperature equal to the current configuration. In this sense, the operation limit is represented by the *start austenitizing temperature (Ac1)* at which a metallography changing of the basement of hard chromium coating starts. Figure 4-50 and Figure 4-51 show the *maps of operation* concerning respectively the actual configuration (the reference case) and the optimal case. The criteria used for the area subdivision is the following:
    - *green area* = $T_{MAX\_MANDREL} \leq T_{START\_AUSTENITIZING\ (Ac1)}$;
    - *yellow area* = $T_{START\_AUSTENITIZING(Ac1)} \leq T_{MAX\_MANDREL} \leq T_{END\_AUSTENITIZING\ (Ac3)}$;
    - *red area* = $T_{MAX\_MANDREL} \geq T_{END\_AUSTENITIZING\ (Ac3)}$;

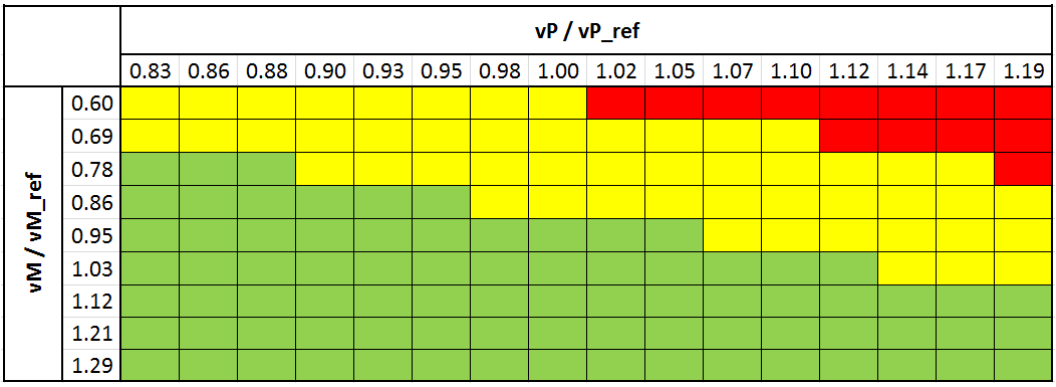

Figure 4-50: Maps of Operation – Current Elongation Distribution

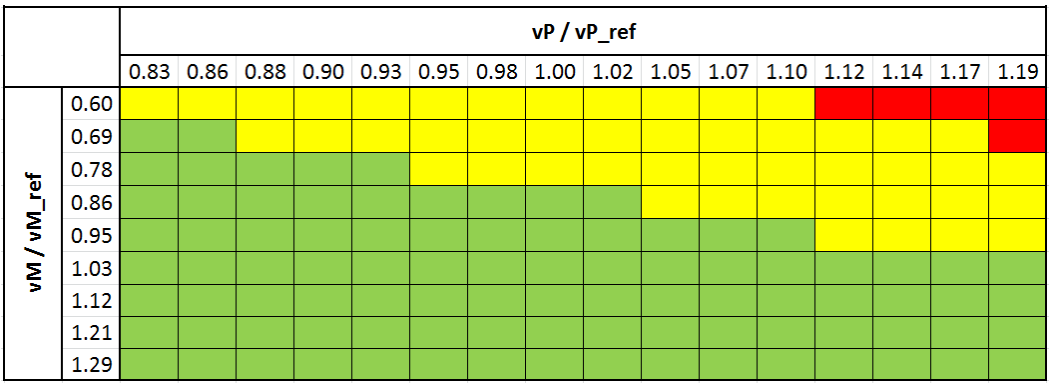

Figure 4-51: Maps of Operation – Optimal Elongation Distribution

The operation limit is represented by the boundary between the green and the yellow zone.

The optimal elongation allows to move the limit, permitting an increasing of the pipe outlet speed. This results in a 7% production increase.

4.5 Conclusions

The thermal cycle of the mandrel of a longitudinal mandrel mill has been investigated and optimized, with the objective of reducing the peak temperature of mandrel during the rolling phase. The activity has been carried out at the R&D Centre of TenarisDalmine S.p.A., one of the largest seamless steel pipe producers in the world.

A Design of Experiments has been carried out, with the aim to identify the most influential parameters on the quantity under investigation. As a result, it turned out that the peak temperature of mandrel is affected by the elongation distribution, the pipe outlet speed and the mandrel speed. Neither the distance between the stands nor the diameter of roll of the fourth stand have any influence instead. It has been found that the peak temperature reached by the mandrel decreases reducing the pipe outlet speed and increasing the mandrel speed. These trends show that there is an important influence of the relative speed between the pipe and the mandrel on the mandrel peak temperature.

A further optimization of the most influential parameters has been implemented, in order to determine the optimal configuration that minimizes the peak temperature of the mandrel. The optimization has been carried out keeping fixed the pipe outlet speed, with the aim to find the optimum concerning a determined productivity of the mandrel mill. A first step of the optimization has been performed using a stochastic genetic algorithm, in order to find the global optimum. A second step has been implemented with a deterministic gradient-based algorithm, to refine the solution in a neighborhood of the optimal solution found in the previous step. The global optimum obtained by two algorithms have been found to be in close agreement, while the gradient-based optimization has shown that many local minima exist on the neighborhood of the global optimum.

The identified optimal solution shows that a reduction of 10% of the mandrel peak temperature is possible, increasing the elongation on the first stand and moving the elongation distribution of the other stands towards the exit of the mandrel mill. The setup found for the first stand allows to increase the mandrel speed, satisfying the technological constraint of the mandrel which must always be slower than the pipe. Whereas the elongation distribution obtained for the other stands allows to increase the pipe speed towards the outlet of the mandrel mill. The consequent setup configuration allows to reduce the integral of the relative speed between the pipe and the mandrel along the entire mandrel mill, reducing the heat generated by the friction phenomena. In addition, it has been found that the variation of the contact pressure and of the contact length due to different elongation distributions has no effect on the mandrel peak temperature.

Compared to the current configuration, every 1000 consecutive cycles, the obtained optimal solution allows a 15% reduction of the mandrel hardness fail achieving a life increase. Alternatively, keeping fixed the peak temperature of the mandrel, which means maintaining the hardness fail and the mandrel life equal to the current configuration, the optimal elongation distribution permits a speed up of the pipe outlet, resulting in a 7% production increase.

With reference to a comparison between the two used optimization algorithms, for this specific application the genetic algorithm seems to be more robust. In fact it has required a lower number of simulations and it has not required for particular settings.

5 Conclusions

5.1 Shape Optimization with Adjoint method

A new direction of development within the research group operating at University of Bergamo on DG methods for Computational Fluid Dynamics has been started.

The present work is about the use of the adjoint method in the framework of aerodynamic shape optimization. The adjoint approach allows to solve local constrained optimization problems efficiently: the solution of only one linear system related to the constraint is required, independently from the number of the involved design parameters. The technique is applied to shape optimization and grid adaptivity problems for CFD applications, where the number of the involved parameters is generally high.

Starting from the simple quasi-1D Euler equations, the implementation of the discrete adjoint method has been validated by comparing the values of computed adjoint variables with results of analytical solutions available in the literature.

The method has then been applied to a shape optimization problem with a single design variable, using a gradient based algorithm with an inexact line search approach. The sensitivity of the objective function in respect of the design parameter has been evaluated by solving the adjoint system, while the length of the step at each iteration has been quantified requiring the satisfaction of the Armijo condition. The derivative of residual in respect of the design parameter has been computed by means of the finite difference approach. The target value of the objective function has been reached in less than 10 iterations and the values of derivatives in the course of optimization were found to be in close agreement with those obtained by means of the finite difference approach.

The application of the method to practical shape optimization problems requires further developments: the extension to other objective functions, the introduction of surface parameterization techniques, in order to deal with more complex geometries, the extension of the analysis to 2D-3D cases and to viscous and unsteady cases.

5.2 Thermal Cycle Optimization of a Mandrel Mill Component

The thermal cycle of the mandrel of a longitudinal mandrel mill has been investigated and optimized, with the objective of reducing the peak temperature of mandrel during the rolling phase. The activity has been carried out at the R&D Centre of TenarisDalmine S.p.A., one of the largest seamless steel pipe producers in the world.

A Design of Experiments has been carried out, with the aim to identify the most influential parameters on the quantity under investigation. As a result, it turned out that the peak temperature of mandrel is affected by the elongation distribution, the pipe outlet speed and the mandrel speed. Neither the distance between the stands nor the diameter of roll of the fourth stand have any influence instead. It has been found that the peak temperature reached by the mandrel decreases reducing the pipe outlet speed and increasing the mandrel speed. These trends show that there is an important influence of the relative speed between the pipe and the mandrel on the mandrel peak temperature.

A further optimization of the most influential parameters has been implemented, in order to determine the optimal configuration that minimizes the peak temperature of the mandrel. The optimization has been carried out keeping fixed the pipe outlet speed, with the aim to find the optimum concerning a determined productivity of the mandrel mill. A first step of the optimization has been performed using a stochastic genetic algorithm, in order to find the global optimum. A second step has been implemented with a deterministic gradient-based algorithm, to refine the solution in a neighborhood of the optimal solution found in the previous step. The global optimum obtained by two algorithms have been found to be in close agreement, while the gradient-based optimization has shown that many local minima exist on the neighborhood of the global optimum.

The identified optimal solution shows that a reduction of 10% of the mandrel peak temperature is possible, increasing the elongation on the first stand and moving the elongation distribution of the other stands towards the exit of the mandrel mill. The setup found for the first stand allows to increase the mandrel speed, satisfying the technological constraint of the mandrel which must always be slower than the pipe. Whereas the elongation distribution obtained for the other stands allows to increase the pipe speed towards the outlet of the mandrel mill. The consequent setup configuration allows to reduce the integral of the relative speed between the pipe and the mandrel along the entire mandrel mill, reducing the heat generated by the friction phenomena. In addition, it has been found that the variation of the contact pressure and of the contact length due to different elongation distributions has no effect on the mandrel peak temperature.

Compared to the current configuration, every 1000 consecutive cycles, the obtained optimal solution allows a 15% reduction of the mandrel hardness fail achieving a life increase. Alternatively, keeping fixed the peak temperature of the mandrel, which means maintaining the hardness fail and the mandrel life of the current configuration, the optimal elongation distribution permits a speed up of the pipe outlet, resulting in a 7% production increase.

With reference to a comparison between the two used optimization algorithms, for this specific application the genetic algorithm seems to be more robust. In fact it has required a lower number of simulations and it has not required for particular settings.

Bibliography

[1] http://dictionary.cambridge.org/.

[2] http://dictionary.reference.com/.

[3] Marco Cavazzuti, "Optimization method: from theory to design," Università degli Studi di Modena e Reggio Emilia, Phd Thesis 2008.

[4] Jorge Nocedal - Stephen J. Wright, *Numerical Optimization*.: Springer.

[5] Gianni di Pillo - Laura Palagi, "Nonlinear Programming: Introduction Unconstrained and Constrained Optimization," Università degli Studi di Roma "La Sapienza",.

[6] Gianni di Pillo - Laura Palagi, "Ottimizzazione," Università degli Studi di Roma "La Sapienza", Corso Ottimizazione - Lucidi Lezioni 2007.

[7] Giovanni Zilli, "Metodi di Ottimizzazione," Università di Padova,.

[8] Antony Jameson - John C.Vassberg, "Computational Fluid Dynamics for Aerodynamic Design: Its Current and Future Impact," 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 2001-0538, 2001.

[9] Giampietro Carpentieri, "An Adjoint-Based Shape-Optimization Method for Aerodynamic Design," Delft University of Technology, Faculty of Aerospace Engineering, Delft - Netherlands, PhD thesis ISBN 978-90-6464-352-1, 2009.

[10] Jacques E.V. Peter - Richard P. Dwight, "Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches," Computer & Fluids 39 (2010) 373-391 2009.

[11] Michael B.Giles - Niles A.Pierce, "An Introduction to the Adjoint Approach to Design," Oxford University, Oxford - UK, 2000.

[12] Antony Jameson, "Aerodynamic Design Via Control Theory," Princeton University, Princeton, 1988.

[13] Michael B.Giles - niles A.Pierce, "Analitic adjoint solution for the quasi-one-dimensional Euler equations ," oxford University, Oxford - UK, 2000.

[14] C. Lozano - J. Ponsis, "Remarks on the numerical solution of the adjoint quasi-one-dimensional euler equations," International Journal for Numerical Methods in Fluids 2011.

[15] Antony Jameson, "Aerodynamic Shape Optimization Using the Adjoint Method," Stanford University, Lectures at the von Karman Institute 2003.

[16] Eric J. Nielsen, "Aerodynami desing sensitivities on an unstructured mesh using the Navier-Stokes equations and a discrete adjoint formulation," Faculty of Virginia Polytechnic Insitute and State University, 1998.

[17] Renè Schneider, "Application of the Discrete Adjoint Method in Computational Fluid Dynamics," University of Leeds - UK, PhD Thesis 2006.

[18] Siva K. Nadarajah - Antony Jameson, "Study of the Continuous and Discrete Adjoint Approaches to Viscous Automatic Aerodynamic Shape Optimization," Stanford University, AIAA-2001-2530, 2001.

[19] F.Bassi - S.Rebay, "High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations," Università degli Studi di Ancona, Ancona - IT, 1996.

[20] Vamshi Mohan Korivi - Arthur C.Taylor - Perry A.Newman - Gene W.Hou - henry E.Jones, "An Approximately Factored Incremental Strategy for Calculating Consistent Discrete Aerodynamic Sensitivity Derivatives," Journal of Computational Physics 113, 336-346, 1994.

[21] David G. Lilley David R.Croft, *Heat transfer calculations using finite difference equations*.: Applied science published LTD, 1997.

[22] MathWorks. Matlab Documentation Centre. [Online]. http://www.mathworks.it/it/help/matlab/