Politecnico di Milano

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in Ingegneria Matematica

Tesi di Laurea Magistrale

Multiple-Gradient Descent Algorithm for isogeometric shape optimization

Relatore:
Prof. Luca Formaggia

Correlatori:
Prof. Jean-Antoine Désidéri
Dr. Régis Duvigneau

Laureando:
Matteo Giacomini
Matricola 754590

Anno Accademico 2012-2013

Доверяй, но проверяй.
Trust, but verify.
(Russian proverb)

Abstract

This thesis has been developed during a six-month internship within the OPALE research team at INRIA Méditerranée (*Institut Nationale de Recherche en Informatique et Automatique* - Sophia Antipolis, France) under the supervision of Jean-Antoine Désidéri and Régis Duvigneau. This work focuses on the research field of optimization. Moreover it deals with topics related to numerical methods for the approximation of PDEs and applications to computational mechanics.

This work faces some theoretical issues in the cooperative phase of multiobjective optimization and applies the resulting methodology to a shape optimization problem in linear elasticity.

In the first part of this thesis, a general paradigm for the treatment of multiobjective optimization is presented, introducing the concepts of Pareto-optimal solutions and Pareto fronts. Then we describe a methodology that extends classical *Steepest-Descent Method* to the case of concurrent optimization of several criteria by means of the so-called *Multiple-Gradient Descent Algorithm.*

Moreover we present an application to a linear elasticity problem, formulated using *Iso-Geometric Analysis.* In particular, we propose to analyze a classical problem of shape optimization in structural engineering within a multiobjective optimization framework. Thus the following variants of *Multiple-Gradient Descent Algorithm* are tested: *MGDA* using gradients approximated by Finite Difference Method; *MGDA* assisted by statistical-based metamodels in order to predict the values of the objective functionals; *MGDA* enhanced by the information contained in the gradients analytically extracted from the *NURBS*-based formulation of the problem.

Some numerical simulations for a test case are presented and the results are cross-validated using all the variants of this multiobjective optimization algorithm, a strategy based on shape derivatives and a genetic algorithm widely used in the literature.

Eventually, an introduction to multiobjective competitive optimization is given and the general framework to formulate a proper Nash game is established.

Keywords: Multiobjective optimization, Pareto-optimal solutions, gradient descent, IsoGeometric Analysis, shape optimization, shape gradient, kriging models

Sommario

Questo lavoro di tesi è stato sviluppato durante un periodo di stage di sei mesi svolto all'interno del gruppo di ricerca OPALE presso INRIA Méditerranée (*Institut Nationale de Recherche en Informatique et Automatique* - Sophia Antipolis, Francia). Sviluppato sotto la supervisione del Prof. Jean-Antoine Désidéri e del Dr. Régis Duvigneau, questo lavoro si inquadra nell'area di ricerca dell'ottimizzazione e tratta temi legati ai metodi numerici per le equazioni a derivate parziali con applicazioni alla meccanica computazionale.
Il fulcro di questa tesi è costituito dalla trattazione teorica e algoritmica della fase cooperativa delle procedure di ottimizzazione multiobiettivo. La metodologia risultante è poi applicata a un problema di ottimizzazione di forma formulato mediante l'analisi isogeometrica in un contesto di elasiticità lineare.
Nella prima parte di questo lavoro, si introduce un paradigma generale per la trattazione dei problemi di ottimizzazione multiobiettivo, presentando i concetti di soluzioni Pareto-ottimali e di fronti di Pareto. Successivamente si generalizza il classico algoritmo di massima discesa per l'ottimizzazione di un singolo criterio al caso dell'ottimizzazione contemporanea di molteplici funzionali tramite la strategia denominata *Multiple-Gradient Descent Algorithm*.
Si presenta inoltre l'applicazione a un problema di elasitictà lineare formulato usando elementi finiti isogeometrici. In particolare, proponiamo di analizzare un classico problema di ottimizzazione di forma in ingegneria strutturale ambientandolo in un contesto di ottimizzazione multiobiettivo. Si testano pertanto le seguenti varianti dell'algoritmo di ottimizzazione precedentemente descritto: *MGDA* accoppiato a una procedura per il calcolo approssimato dei gradienti mediante differenze finite; *MGDA* assistito da metamodelli statistici per la predizione dei funzionali costo; *MGDA* potenziato dall'utilizzo delle informazioni sui gradienti estratte in maniera analitica dalla formulazione del problema basata sulle *NURBS*.
Si forniscono i risultati di alcune simulazioni numeriche per un caso test e si esegue una validazione degli stessi usando le varie versioni presentate per l'algoritmo di ottimizzazione multiobiettivo, una strategia basata sulle derivate di forma e un algoritmo genetico largamente impiegato in letteratura per la trattazione di questa categoria di problemi.
Infine, si presenta un'introduzione ai problemi di ottimizzazione multiobiettivo competitiva e si fornisce l'ambientazione generale per la corretta formulazione del corrispondente gioco di Nash.

Parole chiave: Ottimizzazione multiobiettivo, soluzioni Pareto-ottimali, metodo di discesa, analisi isogeometrica, ottimizzazione di forma, gradiente di forma, modelli di kriging

List of Figures

List of Tables

Contents

Introduction

In recent years optimization has gained an increasing interest among scientists for the interdisciplinary character of the topics it deals with, both from a theoretical point of view and for the wide range of possible applications.
When dealing with real life problems, the first step one has to face is the mathematical model of the phenomena under analysis. These problems are usually extremely complex and involve several quantities of different nature. In general, a detailed study involves the analysis of different aspects at the same time and issues on the possibility of properly comparing and optimizing the relevant quantities arise.

Within this framework, classical single objective optimization presents several limitations and a more general paradigm to analyze multiple objectives is necessary. Thus, in this work we focus on multiobjective optimization and we present a general setting for the treatment of complex optimization problems.

In the literature, multiobjective optimization problems have been classically treated by means of penalty approaches or other methodologies strongly dependent on the calibration of the parameters in the model and this limited their efficacy. An alternative approach relies on the definition of the so-called *Pareto-optimal solutions*: these configurations optimize the objective functionals with different orders of accuracy, thus leading to a set of optima for the global problem. However, a ranking among the Pareto-optimal solutions cannot be established since every configuration focuses on the optimization of one specific criterion (or one subset of criteria) whereas the remaining ones could worsen or not improve. In this scenario, the choice among the different final configurations can be performed only if additional information on the nature of the problem or the user's goal is known.

In this work we seek a deterministic approach for the treatment of optimization problems with multiple criteria and we start from the recent works of J.-A. Désidéri on *Multiple-Gradient Descent Algorithm* (*MGDA*). We will extensively present this methodology which is a generalization of the classical *Steepest-Descent Method* to the case of several objective functionals. From a theoretical point of view, we present a detailed description of the algorithm and some important results for the finite-dimensional case. An extension to infinite-dimensional problems in Hilbert spaces is possible but some issues on the convergence of the iterates of the method are still open.

From a computational point of view, a major drawback of the deterministic methods for multiobjective optimization is the high number of different configurations that have to be evaluated for every objective functionals. This can result in a bottleneck if the solver invoked by the problem under analysis is very demanding, for example a code to simulate turbulent Navier-Stokes equations in non-trivial three-dimensional geometries. For this reason, the coupling of optimization algorithms with prediction methodologies has been studied, especially focusing on statistical-based techniques for spatial data such *kriging models*.

Concerning the applications, in this thesis we focus on the field of continuum mechanics and we deal with a linear elasticity problem. In particular, we are interested in studying a shape optimization problem arising from structural engineering. For the numerical approximation of the problem, we use *IsoGeometric Analysis* (*IGA*), where the finite-dimensional space used in the Galerkin method is constructed using *Non-Uniform Rational B-Splines* (*NURBS*). *NURBS* represent a *de facto* standard in *Computer Aided Design* and allow to save the computational cost associated with the mesh generation in classical numerical methods since the same parametrization is used for both geometry and unknown fields. For the shape optimization problem, issues related to the differentiation with respect to the domain are faced and an optimization procedure based on shape derivatives is tested. The same problem is reformulated within a multiobjective optimization paradigm and is solved by using several variants of *MGDA*: first, an optimization strategy is implemented by using Finite Difference Method to approximate the gradients of the objective functionals; then, the multiobjective descent strategy is coupled with a kriging model in order to reduce the overall computational cost due to the evaluation of the functionals; eventually, the information on the gradients of the objective functionals is extracted directly from the isogeometric solver and *MGDA* is tested using analytical gradients.
The thesis is organized as follows:
Chapter 1: We present a brief overview of several topics in optimization that will be later treated in this thesis. In particular, we present an introduction to multidisciplinary optimization highlighting cooperative and competitive phases. Concerning the shape optimization problem, we recall some concepts of functional differentiation in Banach spaces and we present a basic introduction to differentiation with respect to the domain.
Chapter 2: We introduce the general concepts of *Pareto-optimality* and *Pareto-stationarity* for multiobjective optimization and we establish a *Multiple-Gradient Descent Algorithm* to perform cooperative optimization. Then we state some major results of existence and convergence of the sequence of *MGDA* iterates and we provide some details on the strategy for the numerical implementation of the procedure. Eventually metamodel-assisted variants of the optimization algorithm are proposed, especially focusing on *kriging models*.
Chapter 3: We introduce the linear elasticity problem and the isogeometric paradigm for the formulation of a solver based on *IsoGeometric Analysis*. We recall the construction procedure to obtain *Non-Uniform Rational B-Splines* and we write the differential problem within this framework. Then we present a shape optimization problem widely studied in structural engineering and several approaches to its analysis based on penalty parameters, Lagrange multipliers and shape derivatives. Eventually we reformulate this problem in order to be able to treat it within a multiobjective optimization paradigm.
Chapter 4: We combine the results from the previous chapters and we detail classical *MGDA* procedure where the gradients of the functionals have to be numerically approximated as well as metamodel-assisted *MGDA* where we couple the optimization procedure with the statistical information arising from kriging prediction models. Eventually we use the isogeometric parametrization to extract the gradients of the objective functionals and we improve classical *MGDA* by means of the analytical expressions of the gradients.
Chapter 5: We present some numerical results for a two dimensional test case well described in the literature. We compare first the configurations obtained using different variants of *MGDA* with the one arising from a *Steepest Descent Method* formulated using shape derivatives. Then, we cross-validate the description of the Pareto front with the results of a genetic algorithm widely used in the literature. Eventually some observations

on the performance of these algorithms are presented.
A conclusive chapter summarizes the results of this thesis and highlights open issues and future developments. Two additional appendixes contain further information not essential to the understanding of this work but useful for the comprehension of some major issues in multiobjective optimization. In particular, appendix A provides a brief introduction to competitive optimization: we introduce the general framework to formulate a multiobjective optimization strategy using Nash games and we highlight the possible extension of the discussed procedures, starting from the Pareto-optimal solutions given by *MGDA*. Eventually, in appendix B, we report the proofs of the main theoretical results presented for cooperative optimization.

Chapter 1

Multiobjective optimization and shape optimization

This work deals with several aspects related to the fields of numerical optimization, differential geometry and scientific computing. From a theoretical point of view, this thesis focuses on multiobjective optimization, that is the treatment of complex systems by concurrently optimizing several criteria. For this reason in this section we present a brief overview of main concepts in the field of multidisciplinary optimization and we describe different approaches to the problem. In details, we will focus on the identification of the quantities that describe the concept of optimality in a multiobjective optimization problem and on the techniques to achieve this characterization (Sections 1.1 and 1.2). A very good introduction to these topics is available in [Mie99] and [NW99].

Concerning the applications of multiobjective optimization, in this work we study a shape optimization problem arising from structural engineering. First we parametrize the domain and the boundary using *Non-Uniform Rational B-Splines*, then we solve an elasticity problem by means of *IsoGeometric Analysis*. Thus the shape optimization problem is set in a multiobjective optimization framework and the objective functionals are concurrently optimized. An introduction to *NURBS* will be given in chapter 3 whereas in sections 1.3 and 1.4 we present an overview of the most important results about functional differentiation in Banach spaces and differentiation with respect to the domain, namely Fréchet derivatives and shape derivatives. For more details on these topics we refer to [HM03] and [DZ01].

1.1 Basic concepts of multiobjective optimization

In this section we introduce some basic notions in the field of multiobjective optimization. For the general framework, we consider n objective functionals $J_i(\Omega)\ ,\ i = 1, \dots, n$; for the sake of readibility we will restrict to the specific case of two functionals when the notation of the general scenario is excessively complicated.

First of all we recall the definition of dominance of a design point within a multidisciplinary paradigm:

Definition 1.1 (*Dominance*)**.** Let our problem be the concurrent minimization of n criteria $J_i(\Omega)\ ,\ i = 1, \dots, n$. A design point $\mathbf{Y}^{(0)}$ in the parameter space $\Omega \subset \mathbb{R}^N$ is said to *dominate* the design point $\mathbf{Y}^{(1)} \in \Omega$ in efficiency if and only if

$$J_i(\mathbf{Y}^{(0)}) \leq J_i(\mathbf{Y}^{(1)}) \quad , \quad \forall i = 1, \dots, n$$

and for at least one criterion the inequality is strict.
Under these assumptions, we can write $\mathbf{Y}^{(0)} \succ \mathbf{Y}^{(1)}$, otherwise it holds

$$\mathbf{Y}^{(0)} \not\succ \mathbf{Y}^{(1)} \quad \text{and} \quad \mathbf{Y}^{(1)} \not\succ \mathbf{Y}^{(0)}$$

and the design points are said to be *non-dominated.*

The notion of dominance is essential for the definition of a sorting criterion within a population of design points with respect to the objective functionals $J_i(\Omega)\ ,\ i = 1, \ldots, n$. Thus it is possible to identify the so-called *Pareto fronts*, that is, subsets of design points which are non-dominated with respect to each other.

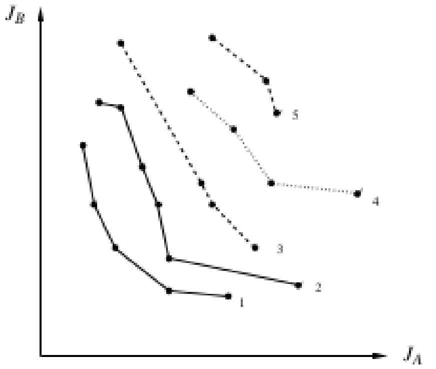


Figure 1.1: Different Pareto fronts in multiobjective optimization of functionals J_A and J_B. Pareto-optimal solutions belong to the first Pareto front, namely the lowest set of points in the space $J_A - J_B$. Figure from [Dés11].

In general, there exist multiple Pareto fronts (Fig. 1.1) and the aim of the optimization procedure is to identify the *first Pareto front*, meaning the set of all design points dominated by no others. For this reason we introduce the notion of Pareto-optimal solution as follows:

Definition 1.2 (*Pareto-optimality*)**.** Let $\mathbf{Y}^{(0)}$ be an admissible design point in $\Omega \subset \mathbb{R}^N$. $\mathbf{Y}^{(0)}$ is said to be *Pareto-optimal* if it is not possible to reduce the local value of any criterion without increasing at least one of the other criteria.

Definition 1.2 can be reformulated by means of the concept of first Pareto front: a design point is said to be Pareto-optimal if it belongs to the first Pareto front.

1.2 MultiDisciplinary Optimization

In the literature, the expression *MultiDisciplinary Optimization* generally refers to the optimization of complex systems where several objective functionals have to be considered. For a general introduction to the methodologies to deal with these methods we refer to [KN05] and [Ted07]. A common approach consists of strategies that first analyze and locally minimize subsystems considering only one discipline at time, then integrate them in a global coupled system by means of an appropriate combination of the results ([SSH97] and [SSAPS03]). Other approaches have been proposed by Y. Parte et al. ([PAC$^+$10] and [AL99]) and are based on the idea of adding a non-linear equation to establish the equivalent of a constitutive relationship among the design variables. Nevertheless, all these strategies tend to increase both the theoretical difficulty and the computational cost of the optimization procedures due to the additional information they require in order to obtain a problem that can be solved.

In general, MultiDisciplinary Optimization involves the evaluation of a large number of different configurations of the design parameters and this leads to high computational costs for which efficient approaches are required. Among the possibilities we recall the use of hierarchical model adaptivity, geometrical shape parametrizations, high performance implementations and efficient optimizers. In this work, we focus on the last field for which we present a brief overview in this section.

A classical strategy to take into account several objective functionals J_i's $i = 1, \dots, n$ consists of agglomerating them in a single criterion which is a convex combination achieved by means of appropriate weights:

$$\overline{J} = \sum_{i=1}^{n} \alpha_i J_i \quad , \quad \alpha_i \geq 0 \quad \forall i = 1, \dots, n$$

As widely observed in the literature, this approach presents heavy limitations due to the arbitrariness of the calibration required by the weight coefficients and in general provides significant results only when the initial design point is close to be satisfactory and minor improvements are sought.

An alternative approach consists of a two-step procedure. First the objective functionals J_i's $i = 1, \dots, n$ are optimized alone within a single objective paradigm - possibly under given constraints - and $\forall i = 1, \dots, n$ an optimal solution J_i^* is identified; in a second step the following minimization problem is established

$$\min_{s \in \mathcal{S}_{ad}} s$$

where $\mathcal{S}_{ad}$ is said to be the set of admissible perturbations s such that $\forall i = 1, \dots, n$ the variations of the objective functional J_i's are bounded by a small perturbation about the optimal values J_i^*'s:

$$\mathcal{S}_{ad} = \left\{ s \in \mathbb{R} \ \middle| \ J_i \leq J_i^* + \alpha_i s \quad , \quad \forall i = 1, \dots, n \right\}$$

Thus we minimize the error introduced by the procedure of concurrent optimization as a perturbation of the optimal values J_i^*'s. As for the convex combination of the objective functionals, this approach presents major issues of stability and robustness.

A totally different approach relying on an alternative formulation of the problem consists of considering a two-phase optimization strategy where in the first place the objective functionals are concurrently minimized (*Cooperative phase*) then a policy to optimize one criterion without excessively worsening the others is established and performed (*Competitive phase*) ([DPAM00] and [Dés11]). We remark that this approach clearly holds only if the analyzed criteria are antagonistic, that is, a global optimal solution does not exist and we seek the design points that realize Pareto-optimal solutions as defined in the previous section. On the contrary, if the problem under analysis admits a global optimum, the Pareto front is degenerate and there exists one unique configuration that dominates all the remaining ones. In this scenario, cooperative optimization is still feasible even if trivial but the competitive phase can no longer be applied.

In the following subsections, we will present the general idea of this two-step algorithm and during the rest of the thesis we will focus on the cooperative phase which is the main subject of this work.

1.2.1 Cooperative phase

Starting from a generic design point $\mathbf{Y}^{(0)}$, first of all we are interested in optimizing all the criteria, that is, we want to identify a new design point $\mathbf{Y}^{(1)}$ such that $\mathbf{Y}^{(1)} \succ \mathbf{Y}^{(0)}$. In the literature, this problem is often faced by means of genetic algorithms which are suitable but have major drawbacks because of their computational costs.

Here we treat this problem by using a gradient-based approach that extendes the *Steepest Descent Method* to multiobjective optimization. In particular, this work focuses on the phase of cooperative optimization, that is the procedure by which all the objective functionals improve at the same time. We refer to section 2 for a detailed presentation of *Multiple-Gradient Descent Algorithm* and to section 4 for the application of this methodology to a problem of shape optimization.

1.2.2 Competitive phase

Basic idea of the competitive optimization consists of simulating a dynamic game and seeking a Nash equilibrium. In particular, a split of the territory is performed by identifying complementary subsets of the design variables and each criterion to be optimized is assigned to a virtual player as individual strategy. If an equilibrium point is achieved, it represents a trade-off among the criteria, that is, a configuration where the objective functionals are optimized and the values that do not improve do not suffer from excessive worsening either. In a symmetrical Nash game ([Nas51]), each player accomodates its own strategy to the other players in order to optimize only the criterion he has been assigned.

A brief introduction to optimization based on Nash games is presented in appendix A. In this work we focus on the cooperative phase during multiobjective optimization thus we will not provide a detailed presentation of this topic. However, the design points arising from cooperative optimization can be either interpreted as final configurations or starting point for the competitive optimization strategy thus a a general overview is mandatory since competitive optimization represents the second phase of the overall two-step optimization strategy. Further details including some theoretical results on existence and performance are available in [Dés11] and [AEM07].

1.3 Basic concepts of differential calculus in Banach spaces

Here we introduce the concepts of Gâteaux and Fréchet derivatives for given operators in Banach spaces in order to extend classical notions of differentiability and directional derivative to these spaces. For a detailed description of these topics we refer to [Sal08] and [Eva10].

Let us consider two Banach spaces X, Y and an operator $F : U \subseteq X \to Y$ where U is an open set.

Definition 1.3 (*Gâteaux derivative*)**.** F is said to be *Gâteaux differentiable* in $x_0 \in U$ if $\forall h \in X$ and $\forall t$ such that $x_0 + th \in U$, there exists an operator $d_G F \in \mathcal{L}(X, Y)$ such that

$$d_G F(x_0)h = \lim_{t \searrow 0} \frac{F(x_0 + th) - F(x_0)}{t} \tag{1.1}$$

In particular, if F is G-differentiable in every point of U, the application $d_G F : U \to \mathcal{L}(X, Y)$ is unique and is said to be the *Gâteaux derivative* of F.

Definition 1.4 (*Fréchet derivative*)**.** F is said to be *Fréchet differentiable* in $x_0 \in U$ if $\forall h \in X$ such that $x_0 + h \in U$, there exists a bounded linear operator $dF \in \mathcal{L}(X, Y)$ such that

$$\lim_{h \searrow 0} \frac{\|F(x_0 + h) - F(x_0) - dF(x_0)h\|_Y}{\|h\|_X} = 0 \tag{1.2}$$

In particular, if F is F-differentiable in every point of U, the application $dF : U \to \mathcal{L}(X, Y)$ is unique and is said to be the *Fréchet derivative* of F.

If $F(x)$ is Fréchet differentiable then it is also Gâteaux differentiable and its Fréchet and Gâteaux derivatives are the same. However, Gâteaux differentiability is only a necessary condition for Fréchet differentiability since the Gâteaux derivative may fail to be linear or continuous. Moreover, it is even possible for the Gâteaux derivative to be linear and continuous but for the Fréchet derivative not to exist. Here we recall a general result that establishes a relationship between Gâteaux and Fréchet derivatives:

Theorem 1.5. *If $F : U \subseteq X \to Y$ is Gâteaux differentiable in U and $d_G F$ is continuous in $x_0 \in U$, then F is Fréchet differentiable and $dF(x_0) = d_G F(x_0)$.*

Remark 1.6 (Composition and chain rule)*.* Let X, Y, Z be three Banach spaces and consider two operators $F : U \subseteq X \to Y$ and $G : V \subseteq Y \to Z$ where U and V are open subsets. If F is Gâteaux (respectively Fréchet) differentiable in $x_0 \in U$ and G is Gâteaux (respectively Fréchet) differentiable in $y_0 = F(x_0)$, then

$$d_G(G \circ F)(x_0) = d_G G(y_0) \circ F(x_0) \quad , \quad d(G \circ F)(x_0) = dG(y_0) \circ F(x_0)$$

Remark 1.7 (Functional case). Let us consider a functional $F : U \subseteq X \to \mathbb{R}$. Thus $dF(x_0) \in X^*$ and we get $dF : U \to X^*$. If X is a Hilbert space with given internal product $(\cdot, \cdot)_X$ we get

$$dF(x_0)h = (\mathcal{F}, h)_X = (\nabla F(x_0), h)_X \tag{1.3}$$

where the element $\mathcal{F} \in X$ that satisfies equation (1.3) is said to be the gradient of F in x_0.

1.4 Basic concepts of shape optimization

In this section we present a general introduction to the concept of shape optimization following [SZ92] and [BB05]. In particular, we deal with problems of optimization where the design variable is the domain of the problem itself. Thus the optimization problem can be read as the minimization of a vector of objective functionals $\mathbb{J}(D) = (J_1(D), \ldots, J_n(D))^T$ subject to some constraints that state the admissibility of the shapes under analysis. We define the set of admissible shapes $\mathcal{O}_{ad}$ and we seek a domain $\widehat{D}$ such that

$$\widehat{D} = \underset{D \,\in\, \mathcal{O}_{ad}}{\operatorname{argmin}} \; \mathbb{J}(D)$$

Using this formulation, the control variable of the problem is the domain D which can be represented by means of different parametrizations depending on the nature of the problem. In this work, we implemented a problem of shape optimization within an isogeometric paradigm using a parametrization based on *Non-Uniform Rational B-Splines.* Thus our design variable $\mathbf{Y}$ will be the vector of control points that define the *NURBS* curves and surfaces.

The forms of the objective functionals depend on the nature of the problem under analysis. For example, in problems of *Computational Fluid Dynamics*, the aim usually is to determine the optimal shape in order to reduce the drag subject to a lift constraint and some structural limitations. In the field of elasticity, main goal is to reduce the compliance given some volume constraints. Thus, the functionals usually depend on the domain D, on an unknown field $\mathbf{u}(D)$ - for example the displacement, the deformation or the velocity - and on its gradient $\nabla\mathbf{u}(D)$ as follows:

$$J(D) = J(D, \mathbf{u}(D), \nabla\mathbf{u}(D))$$

In the following sections we will introduce the concept of deformation of the domain driven by a given map and some basic techniques of differentiation in Banach spaces. Moreover we will present an overview of the differentiation with respect to the domain in order to be able to write the optimality conditions for a shape optimization problem.

1.4.1 Deformation of the domain

In [DZ01] J.-P. Zolésio et al. present the theoretical foundations of the concept of *shape derivative* which has a major role in the context of shape optimization we will introduce in the following sections.
Let $\mathcal{T}$ be the space of diffeomorfism transformations in $\mathbb{R}^d$ defined as

$$\mathcal{T} = \left\{ T : \mathbb{R}^d \to \mathbb{R}^d \;\mid\; (T - I) \in W^{1,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right) \;,\; (T^{-1} - I) \in W^{1,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right) \right\} \tag{1.4}$$

We introduce the space of the admissible shapes arising from a deformation of D

$$\mathcal{O}_{\mathcal{T}}(D) = \left\{ D_0 \;\middle|\; \exists T \in \mathcal{T} \text{ such that } D_0 = T(D) \right\} \tag{1.5}$$

Let $\mathbf{v} : \mathbb{R}^d \to \mathbb{R}^d$ be an admissible vector field that defines a sufficiently smooth shape deformation arising from the optimization process. Moreover, we define the transformation that maps the initial domain D into the moving domain at time t as $T_t : \mathbb{R}^d \to \mathbb{R}^d$ such that

$$D_t = T_t(\mathbf{v})(D) = D + t\mathbf{v}(D) = (I + t\mathbf{v})(D) \tag{1.6}$$

In general, equation (1.6) describes a map known as *perturbation of the identity* which takes into account the transformation arising from a displacement field $\mathbf{v}$ and can be written as

$$T_t(\mathbf{v}) = I + t\mathbf{v} \quad , \quad \mathbf{v} \in W^{1,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right) \tag{1.7}$$

The following lemma holds and for the proof we refer to [All06].

Proposition 1.8. *Let us consider* $\mathbf{v} \in W^{1,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$ *such that* $\|\mathbf{v}\|_{W^{1,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)} < 1$. *Then the map* $T_t(\mathbf{v}) = I + t\mathbf{v}$ *belongs to the space* $\mathcal{T}$.
Moreover, if $\exists\alpha > 0$ *such that* $\|\mathbf{v}\|_{W^{1,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)} \leq 1 - \alpha$ *then*

$$\|T_t(\mathbf{v})^{-1} - I\|_{W^{1,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)} \leq \frac{1}{\alpha}.$$

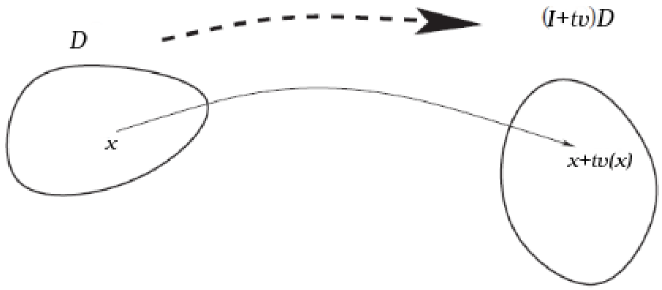


Figure 1.2: Transformation that maps the initial domain D into the moving domain at time t $D_t = (I + t\mathbf{v})D$. Figure from [All06].

1.4.2 Differentiation with respect to the domain

Within the framework of problems of shape optimization, the domain changes during the evolution of the optimization algorithm, thus a way to take into account these changes is required. We introduce the concepts of shape gradient and shape derivative in order to consider the way the objective functionals of the optimization problem change with respect to the transformation of the domain.

Definition 1.9 (*Shape gradient*)**.** Let $J : \mathcal{O}_{\mathcal{T}}(D) \to \mathbb{R}$ and T_t be as in equation (1.7). We define the *Eulerian derivative* of J in the domain D along the direction $\mathbf{v}$ as

$$dJ(D; \mathbf{v}) = \lim_{t \searrow 0} \frac{J(D_t) - J(D)}{t} = \lim_{t \searrow 0} \frac{J((I + t\mathbf{v})(D)) - J(D)}{t} \tag{1.8}$$

If equation (1.8) holds $\forall \mathbf{v} \in \mathcal{D}(\mathbb{R}^d; \mathbb{R}^d)$ and the map $\mathbf{v} \mapsto dJ(D; \mathbf{v})$ is linear and continuous, then J is said to be *differentiable with respect to* D and we get

$$dJ(D; \mathbf{v}) = \langle \nabla J(D)\ ,\ \mathbf{v} \rangle_{\mathcal{D}'(\mathbb{R}^d;\mathbb{R}^d) \times \mathcal{D}(\mathbb{R}^d;\mathbb{R}^d)}$$

where $\nabla J(D)$ is known as *shape gradient*.

In the rest of this section we recall some theoretical results on the differentiation of objective functionals with respect to the domain.

Theorem 1.10. *Let D be a regular domain.*

1. *Let $f \in W^{1,1}(\mathbb{R}^d)$ and $J : \mathcal{O}_{\mathcal{T}} \to \mathbb{R}$ the objective functional defined as follows*

$$J(D_0) = \int_{D_0} f(\mathbf{x}) d\mathbf{x}$$

Then $J(D_0)$ is differentiable with respect to D along the direction $\mathbf{v}$ and we get

$$dJ(D; \mathbf{v}) = \int_{\partial D} f(\mathbf{x}) \mathbf{v} \cdot \mathbf{n} d\mathbf{x} \qquad \forall \mathbf{v} \in W^{1,\infty}\left(\mathbb{R}^d, \mathbb{R}^d\right)$$

where $\mathbf{n}$ is the outward normal to the surface ∂D.

2. *Let $f \in W^{2,1}(\mathbb{R}^d)$ and $J : \mathcal{O}_{\mathcal{T}} \to \mathbb{R}$ the objective functional defined as follows*

$$J(D_0) = \int_{\partial D_0} f(\sigma) d\sigma$$

Then $J(D_0)$ is differentiable with respect to D along the direction $\mathbf{v}$ and we get

$$dJ(D; \mathbf{v}) = \int_{\partial D} \left(\frac{\partial f}{\partial \mathbf{n}} + Hf \right) \mathbf{v} \cdot \mathbf{n} d\sigma \qquad \forall \mathbf{v} \in \mathcal{C}^1 \left(\mathbb{R}^d, \mathbb{R}^d \right)$$

where $\mathbf{n}$ is the outward normal to the surface ∂D and $H = div\mathbf{n}$ is the mean curvature of the surface ∂D.

Theorem 1.10 is a first statement concerning the differentiation of a functional $J(D)$ with respect to the domain but it is feasible only if the shape is fixed in time. Hence we have to introduce additional concepts to take into account the case where the argument of the integral function depends on the geometry of D as in shape optimization problems.

Definition 1.11 (*Material derivative* and *shape derivative*)**.** Let $y = y(D, \mathbf{x})$ be a function in a suitable Sobolev space $W(D)$ such that it depends both on the geometry of the domain and on the element $\mathbf{x}$ within it.
We define $\dot{y} = \dot{y}(D)$ the element in $W(D)$ such that

$$\dot{y}(D) = \lim_{t \searrow 0} \frac{y(D_t) - y(D)}{t} = \lim_{t \searrow 0} \frac{y((I + t\mathbf{v})D) - y(D)}{t} \tag{1.9}$$

and we name it *material derivative* of y.
We define $y' = y'(D)$ the element in $W(D)$ such that

$$y'(D) = \dot{y}(D) - \mathbf{v} \cdot \nabla y(D) \tag{1.10}$$

and we name it *shape derivative* of y.

Thus theorem 1.10 can be extended by using the concept of shape derivative defined in equation 1.10.

Theorem 1.12. *Let D be a regular domain.*

1. *Let $y : \mathcal{O}_{\mathcal{T}}(D) \to L^1(\mathbb{R}^d)$ such that $y(D) \in W^{1,1}(\mathbb{R}^d)$ and $J : \mathcal{O}_{\mathcal{T}} \to \mathbb{R}$ is the objective functional defined as follows*

$$J(D_0) = \int_{D_0} y(D_0, \mathbf{x}) d\mathbf{x}$$

Then $J(D_0)$ is differentiable with respect to D along the direction $\mathbf{v}$ and we get

$$dJ(D; \mathbf{v}) = \int_D y'(D) d\mathbf{x} + \int_{\partial D} y(D) \mathbf{v} \cdot \mathbf{n} d\mathbf{x} \qquad \forall \mathbf{v} \in W^{1,\infty} \left(\mathbb{R}^d, \mathbb{R}^d \right) \tag{1.11}$$

where $\mathbf{n}$ is the outward normal to the surface ∂D.

2. *Let* $y : \mathcal{O}_{\mathcal{T}}(D) \to L^1(\mathbb{R}^d)$ *such that* $y(D) \in W^{2,1}(\mathbb{R}^d)$ *and* $J : \mathcal{O}_{\mathcal{T}} \to \mathbb{R}$ *is the objective functional defined as follows*

$$J(D_0) = \int_{\partial D_0} y(D_0, \sigma) d\sigma$$

Then $J(D_0)$ *is differentiable with respect to* D *along the direction* $\mathbf{v}$ *and we get*

$$dJ(D; \mathbf{v}) = \int_{\partial D} y'(D) d\sigma + \int_{\partial D} \left(\frac{\partial y(D)}{\partial \mathbf{n}} + H y(D) \right) \mathbf{v} \cdot \mathbf{n} d\sigma \qquad \forall \mathbf{v} \in \mathcal{C}^1 \left(\mathbb{R}^d, \mathbb{R}^d \right) \tag{1.12}$$

where $\mathbf{n}$ *is the outward normal to the surface* ∂D *and* $H = div\mathbf{n}$ *is the mean curvature of the surface* ∂D.

Eventually, under some additional assumptions well detailed in [DZ01], it is possible to prove the following structure theorem due to J. Hadamard, which applies to a wide class of functionals:

Theorem 1.13 (Hadamard's structure theorem)**.** *Let* D *be a regular domain and* $J : \mathcal{O}_{\mathcal{T}}(D) \to \mathbb{R}$ *an objective functional. Then there exists a scalar distribution* $g(D)$ *with support within* ∂D *such that* $g(D) \in \mathcal{D}'(\partial D)$ *and*

$$dJ(D; \mathbf{v}) = \langle g(D), \mathbf{v} \cdot \mathbf{n} \rangle_{\mathcal{D}'(\partial D) \times \mathcal{D}(\partial D)} \qquad \forall \mathbf{v} \in \mathcal{D}(\mathbb{R}^d, \mathbb{R}^d)$$

Moreover, if $g(D) \in L^1(\partial D)$ *the following result - known as Hadamard's formula - stands*

$$dJ(D; \mathbf{v}) = \int_{\partial D} g(D) \mathbf{v} \cdot \mathbf{n} d\sigma \tag{1.13}$$

For the proofs of these theorems and for a detailed presentation of these topics we refer to [All06], [HM03] and [HP05].

Chapter 2

Multiple-Gradient Descent Algorithm

In this chapter we focus on the cooperative phase of the strategy previously presented for the treatment of multiobjective optimization problems. In details, we describe an approach for concurrent optimization based on gradient strategies, firstly proposed by J.-A. Désidéri in [Dés09]. Basic idea of the method known as *Multiple-Gradient Descent Algorithm* is to identify a direction common to all criteria along which the value of every functional improves: thus, for a multiobjective minimization problem we seek a descent direction common to all criteria leading to a strategy that generalizes the classical *Steepest Descent Method* for single objective optimization problems.

As pointed out in section 1.1, in multiobjective optimation problems the concept of optimal solutions is replaced by the notion of Pareto-optimal solutions, that is points belonging to the first Pareto front. In general, there is not uniqueness for Pareto-optimal solutions unless the objective functionals are dependent on each other; this is a trivial situation since the problem under analysis results in a single objective optimization problem and the whole approach to cooperative and competitive optimization does not make any sense.

In the following sections we will focus on the description of *Multiple-Gradient Descent Algorithm*, highlighting the main results about the existence of Pareto-optimal solutions and the algorithmic strategy for the classical and metamodel-assisted procedures. The proofs of the results presented in this section are available in appendix B and are all due to the works of J.-A. Désidéri ([Dés09], [Dés12a], [Dés12c] and [Dés12b]).

2.1 Introduction and general results

In this section, we focus on the concurrent phase of the optimization of n criteria $J_i(\mathbf{Y})$, $i = 1, \ldots, n$. Let $\Omega \subset \mathbb{R}^N$ be the space of admissible design points, we suppose that $\forall i = 1, \ldots, n$ the $J_i(\mathbf{Y})$'s have enough regularity, meaning that they are smooth functions of the design vector $\mathbf{Y} \in \Omega$:

$$J_i(\mathbf{Y}) \in \mathcal{C}^1(\Omega) \quad , \quad i = 1, \ldots, n \leq N$$

From now on, we consider the multiobjective optimization problem given by the concurrent minimization of n unconstrained criteria $J_i(\mathbf{Y})$'s in $\Omega \subset \mathbb{R}^N$ as our standard model problem:

$$\min_{\mathbf{Y} \in \Omega} \mathbb{J}(\mathbf{Y}) \qquad , \qquad \mathbb{J}(\mathbf{Y}) = (J_1(\mathbf{Y}), \ldots, J_n(\mathbf{Y}))^T \tag{2.1}$$

2.1.1 Pareto concepts

First of all we introduce the notion of Pareto-stationarity as in [Dés12c]:

Definition 2.1 (*Pareto-stationarity*)**.** Let $\mathbf{Y}^{(0)}$ be a design point at the center of an open ball within the admissible domain $\Omega \subset \mathbb{R}^N$. We assume that the n functional criteria $J_i(\mathbf{Y})$, $i = 1, \ldots, n$ are smooth in Ω and we consider the vector of the local gradients $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)^T$ where $\mathbf{u}_i = \nabla J_i(\mathbf{Y}^{(0)})$ is the gradient of the i-th objective functional. The design point $\mathbf{Y}^{(0)}$ is said to be *Pareto-stationary* if and only if there exists a convex combination of the gradient vectors $\mathbf{u}_i$'s that is equal to zero, that is

$$\sum_{i=1}^{n} \alpha_i \mathbf{u}_i = 0 \quad , \quad \alpha_i \geq 0 \qquad \forall i = 1, \ldots, n \tag{2.2}$$

where the weights α_i's constitute a partition of the unity $\sum_{i=1}^{n} \alpha_i = 1$.

Now we can state the following result that establishes a relationship between Pareto-stationarity and Pareto-optimality:

Proposition 2.2. *Let $\mathbf{Y}^{(0)}$ be an admissibile design point in Ω. If $\mathbf{Y}^{(0)}$ is Pareto-optimal, then it is Pareto-stationary.*

Thus, for our model problem (2.1) and in general for smooth unconstrained criteria, Pareto-stationarity is a necessary condition for Pareto-optimality. As a matter of fact, if the design point $\mathbf{Y}^{(0)}$ belongs to the Pareto set, by instinct we cannot find a new design point that dominates it, that is the elements of the gradient vectors have opposite signs and we can find a linear combination such that equation (2.2) holds.
On the other side, if a design point $\mathbf{Y}^{(0)}$ is not Pareto-stationary for the smooth criteria $J_i(\mathbf{Y})$, $i = 1, \ldots, n$, then a descent direction $\boldsymbol{\omega}$ common to all objective functionals exists and by choosing a step size ρ small enough, the evaluations of the J_i's at $\mathbf{Y} = \mathbf{Y}^{(0)} - \rho\boldsymbol{\omega}$ will improve the values previously computed in $\mathbf{Y}^{(0)}$.

2.1.2 Existence and uniqueness of the minimal-norm element

Let $(\cdot, \cdot) : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ be the classical scalar product in $\mathbb{R}^N$ and $\| \cdot \| : \mathbb{R}^N \to \mathbb{R}$ the corresponding norm.
Finding the descent direction common to all criteria is equivalent to finding a vector $\boldsymbol{\omega} \in \mathbb{R}^N$ such that

$$(\mathbf{u}_i, \boldsymbol{\omega}) \geq 0 \quad , \quad \mathbf{u}_i = \nabla J_i(\mathbf{Y}^{(0)}) \qquad \forall i = 1, \ldots, n$$

Thus $-\boldsymbol{\omega}$ is one descent direction for our multiobjective optimization problem.

We notice that a normalization of the gradients leaves the problem unchanged, so we generalize the previous form by introducing a family of strictly-positive scaling factors S_i , $i = 1, \ldots, n$:

$$(\mathbf{u}_i, \boldsymbol{\omega}) \geq 0 \quad , \quad \mathbf{u}_i = \frac{1}{S_i} \nabla J_i(\mathbf{Y}^{(0)}) \qquad \forall i = 1, \ldots, n \tag{2.3}$$

Using the gradient vectors $\mathbf{u}_i$, $i = 1, \ldots, n$ as defined in (2.3) we can construct the following convex hull

$$\overline{\mathcal{U}} = \left\{ \mathbf{u} \in \mathbb{R}^N \;\middle|\; \mathbf{u} = \sum_{i=1}^{n} \alpha_i \mathbf{u}_i \quad , \quad \alpha_i \geq 0 \quad \forall i = 1, \ldots, n \quad \sum_{i=1}^{n} \alpha_i = 1 \right\} \tag{2.4}$$

and by exploiting its properties we can establish a general result of existence and uniqueness of a minimizing element in $\overline{\mathcal{U}}$.

Proposition 2.3. *Let $\overline{\mathcal{U}}$ be the convex hull defined in (2.4). There exists one realization of a minimum in $\overline{\mathcal{U}}$ and this minimal-norm element is unique.*

Thanks to the characterization of $\boldsymbol{\omega}$ as the minimal-norm element in $\overline{\mathcal{U}}$ the following result holds:

Proposition 2.4. *For all $\mathbf{u} \in \overline{\mathcal{U}} \ : \ (\mathbf{u}, \boldsymbol{\omega}) \geq \|\boldsymbol{\omega}\|^2$.*

Thus a dichotomy result is established in theorem 2.5 stating whether a given design point is Pareto-stationary or there exists a descent direction common to all criteria.

Theorem 2.5. *Under the assumptions made in propositions 2.2 and 2.4, two cases are possible:*

(i) either $\boldsymbol{\omega} = \mathbf{0}$ and the design point $\mathbf{Y}^{(0)}$ is Pareto-stationary;

(ii) or $\boldsymbol{\omega} \neq \mathbf{0}$ and in correspondance of the design point $\mathbf{Y} = \mathbf{Y}^{(0)}$ the vector $-\boldsymbol{\omega}$ defines a descent direction common to all criteria.

Moreover, if (ii) holds and $\boldsymbol{\omega}$ belongs to the interior $\mathcal{U}$ of $\overline{\mathcal{U}}$ then the Fréchet derivatives are all equal

$$(\mathbf{u}_i, \boldsymbol{\omega}) = \|\boldsymbol{\omega}\|^2 \qquad , \qquad \forall i = 1, \ldots, n$$

and more generally for the scalar product it holds

$$(\mathbf{u}, \boldsymbol{\omega}) = \|\boldsymbol{\omega}\|^2 \qquad , \qquad \forall \mathbf{u} \in \overline{\mathcal{U}}.$$

2.1.3 Convergence of MGDA

Multiple-Gradient Descent Algorithm can either stop after a finite number of iterations achieving a Pareto-stationary design point or generate an infinite sequence of iterates. Here we present a general convergence result for that case:

Theorem 2.6. *If the sequence of iterates $\{\mathbf{Y}^{(k)}\}$ generated by the* Multiple-Gradient Descent Algorithm *is infinite, then it admits a subsequence that converges to a Pareto-stationary design point.*

Remark 2.7. The statement of theorem 2.6 is extremely important when considering the overall two-step optimization strategy introduced in section 1.2. Within this framework, the existence of a Pareto-stationary limit point $\mathbf{Y}^*$ such that $\mathbf{Y}^{(k)} \to \mathbf{Y}^*$ when $k \to \infty$ guarantees the existence of an optimal starting point for the competitive optimization phase. Hence, we expect that using final *MGDA* iterate as design point to construct a Nash game as described in appendix A leads to a robust global strategy to deal with multiobjective optimization problems. Additional information is available in appendix A and in [Dés11].

2.2 Practical implementation of MGDA

Starting from theorem 2.5, we can extract the standard formulation of *Multiple-Gradient Descent Algorithm* as extension of the classical single objective *Steepest Descent Method* to the case of multiobjective optimization. Here we propose the general procedure for *MGDA* optimization algorithm:

Listing 2.1: *Multiple-Gradient Descent Algorithm*

```
1. Initialize the design point $\mathbf{Y} = \mathbf{Y}^{(0)}$;
2. Evaluate $n$ objective functionals $J_i(\mathbf{Y})$ , $\forall i = 1, \dots, n$;
3. Compute the gradient vectors $\mathbf{u}_i = \nabla J_i(\mathbf{Y})/S_i$ , $\forall i = 1, \dots, n$;
4. Determine the minimal-norm element $\boldsymbol{\omega}$ in the convex hull $\overline{\mathcal{U}}$;
5. If $\boldsymbol{\omega} = \mathbf{0}$ (or under a given tolerance), stop;
   Else perform line search to determine the optimal step size $\widetilde{\rho}$;
6. Update design point $\mathbf{Y}$ to $\mathbf{Y} - \widetilde{\rho}\boldsymbol{\omega}$.
```

From script 2.1, we notice that at each iteration of *MGDA* all the criteria decrease thus we refer to this process as a cooperative optimization method; hence, as stated in the introduction of this section, *Multiple-Gradient Descent Algorithm* is a gradient-based method to perform the coooperative phase of a multiobjective optimization procedure.

In the following subsections we present some details about the computational strategies to determine the descent direction $\boldsymbol{\omega}$ and the line search algorithm to identify the optimal step size that $\forall i = 1, \dots, n$ guarantees the best improvement of the objective functionals during the transition from $J_i(\mathbf{Y})$ to $J_i(\mathbf{Y} - \widetilde{\rho}\boldsymbol{\omega})$.

2.2.1 Computing the descent direction

We recall that $\boldsymbol{\omega}$ is the minimal-norm element in the convex hull $\overline{\mathcal{U}}$, that is

$$\boldsymbol{\omega} = \underset{\mathbf{u} \,\in\, \overline{\mathcal{U}}}{\operatorname{argmin}} \; \|\mathbf{u}\|$$

and that every element in this set can be written as a convex combination of the scaled gradient vectors $\mathbf{u}_i$, $i = 1, \dots, n$. Thus the problem reads as the numerical minimization of the quadratic form that expresses $\|\boldsymbol{\omega}\|^2$ in terms of the coefficients $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^T$ of the convex combination subject to two constraints:

$$\min_{\boldsymbol{\alpha} \in A_{ad}} \left\| \sum_{i=1}^{n} \alpha_i \mathbf{u}_i \right\|^2 \tag{2.5}$$

where $A_{ad} \subset \mathbb{R}^n$ is the set of the admissible vectors $\boldsymbol{\alpha}$ such that their components are non-negative and they constitute a partition of the unity:

$$A = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^n \;\middle|\; \alpha_i \geq 0 \quad \forall i = 1, \dots, n \quad , \quad \sum_{i=1}^{n} \alpha_i = 1 \right\} \tag{2.6}$$

From a practical point of view, the determination of the α_i's is performed by means of a change of variables. To satify the positivity constraint $\alpha_i \geq 0 \; \forall i = 1, \dots, n$, we consider $\alpha_i = \sigma_i^2$; thus the coefficients σ_i have to be found on the unit sphere $\mathbb{R}^n$ which is parametrized using $n - 1$ spherical coordinates:

$$\sigma_i = \sin \phi_{i-1} \prod_{j=1}^{n-1} \cos \phi_j$$

Thus the parameters α_i's result

$$\alpha_i = (1 - c_{i-1}) \cdot \prod_{j=1}^{n-1} c_j \quad \forall i = 1, \ldots, n$$

where $c_0 = 0$ and $c_j = (\cos \phi_j)^2$ and $\boldsymbol{\omega}$ is computed by means of a search algorithm in $[0,1]^{n-1}$.

Remark 2.8. In [ZDD11] the authors remark that as long as the gradient vectors $\mathbf{u}_i$, $i = 1, \ldots, n$ are known, problem (2.5) can be solved by means of classical evolution strategies but some issues due to ill-conditioning may appear when dealing with high-dimensional problems.

Basic MGDA strategy ($n = 2$)

In this section we analyze the specific case of two criteria ($n = 2$) since $\boldsymbol{\omega}$ can be explicitly expressed and our application to isogeometric structural shape optimization will focus on the concurrent minimization of two objective functionals.

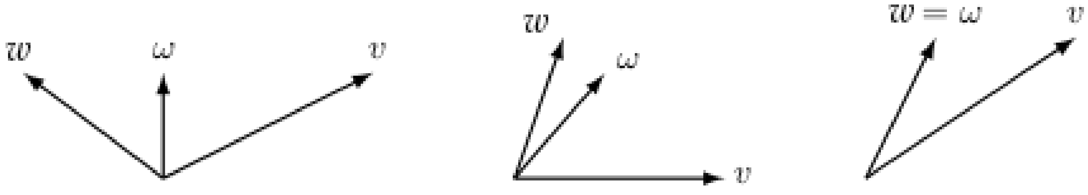


Figure 2.1: Different configurations of the gradient vectors $\mathbf{v}$ and $\mathbf{w}$ to determine $\boldsymbol{\omega} \in \overline{\mathcal{U}}$. On the left, the angle θ is obtuse, in the center it is acute but superior to the limit angle and on the right inferior to the limit angle. Figure from [Dés09].

To simplify the notation let the gradient vectors be $\mathbf{u}_1 = \mathbf{v}$ and $\mathbf{u}_2 = \mathbf{w}$ and the coefficients for convex combination $\alpha_1 = \gamma$ and $\alpha_2 = 1 - \gamma$. We consider the following quadratic form

$$\ell(\gamma) = \|\gamma \mathbf{v} + (1-\gamma)\mathbf{w}\|^2 = (\gamma \mathbf{v} + (1-\gamma)\mathbf{w}, \gamma \mathbf{v} + (1-\gamma)\mathbf{w}) \tag{2.7}$$

so that the first-order derivative results

$$\ell'(\gamma) = 2(\mathbf{v} - \mathbf{w}, \gamma(\mathbf{v} - \mathbf{w}) + \mathbf{w}) \tag{2.8}$$

Thus

$$\ell'(\gamma) = 0 \quad \Leftrightarrow \quad \gamma \|\mathbf{v} - \mathbf{w}\|^2 + \mathbf{w} \cdot (\mathbf{v} - \mathbf{w}) = 0$$

and beyond the trivial case where $\mathbf{v} = \mathbf{w}$, the minimum is achieved for

$$\gamma = \frac{\mathbf{w} \cdot (\mathbf{v} - \mathbf{w})}{\|\mathbf{v} - \mathbf{w}\|^2} = \frac{\|\mathbf{w}\|^2 - \mathbf{w} \cdot \mathbf{v}}{\|\mathbf{v}\|^2 + \|\mathbf{w}\|^2 - 2\mathbf{v} \cdot \mathbf{w}} \tag{2.9}$$

Remark 2.9. From equation (2.9) we notice that if the vectors $\mathbf{v}$ and $\mathbf{w}$ are normalized, that is $\|\mathbf{v}\| = \|\mathbf{w}\| = 1$ then the optimal values for the coefficients are $\gamma = 1 - \gamma = \frac{1}{2}$. On the contrary, if the vectors are not normalized, then it is not sure that γ lies within the interval $[0, 1]$.

Starting from equation (2.9) a condition on the angle $\theta = \widehat{(\mathbf{v},\mathbf{w})}$ can be established:

$$\begin{aligned} 0<\gamma<1 \quad &\Leftrightarrow \quad 0 < \|\mathbf{w}\|^2 - \mathbf{w}\cdot\mathbf{v} < \|\mathbf{v}\|^2 + \|\mathbf{w}\|^2 - 2\mathbf{v}\cdot\mathbf{w} \\ &\Leftrightarrow \quad \mathbf{v}\cdot\mathbf{w} < \min\{\|\mathbf{v}\|^2, \|\mathbf{w}\|^2\} = \Big(\min\{\|\mathbf{v}\|, \|\mathbf{w}\|\}\Big)^2 \\ &\Leftrightarrow \quad \cos\theta < \frac{\min\{\|\mathbf{v}\|, \|\mathbf{w}\|\}}{\max\{\|\mathbf{v}\|, \|\mathbf{w}\|\}} \\ &\Leftrightarrow \quad \theta > \arccos\left(\frac{\min\{\|\mathbf{v}\|, \|\mathbf{w}\|\}}{\max\{\|\mathbf{v}\|, \|\mathbf{w}\|\}}\right) \end{aligned} \tag{2.10}$$

Thus the angle θ between the two gradient vectors has to be at least equal to a certain limit angle which is function of their norms. If the norms of the gradient vectors are very different, then the limit angle is close to $\frac{\pi}{2}$; if they are close to one another, the limit angle is small but the condition is still satisfied except if the directions of $\mathbf{v}$ and $\mathbf{w}$ are too close (Fig. 2.1). Therefore a sufficient condition is for the angle θ to be obtuse, that is $\mathbf{v}\cdot\mathbf{w} < 0$.

In figure 2.1 we observe three different scenarios due to different values of the angle θ. In the first two cases, $\boldsymbol{\omega} \in \mathcal{U}$ and not simply to $\overline{\mathcal{U}}$ and the descent direction points strictly in between $\mathbf{v}$ and $\mathbf{w}$. Thus $\boldsymbol{\omega}$ is orthogonal to the line that connects the extremities of $\mathbf{v}$ and $\mathbf{w}$ (which is the convex hull $\overline{\mathcal{U}}$) and $\forall \mathbf{u} \in \overline{\mathcal{U}}$ the inner product is constant

$$(\mathbf{u}, \boldsymbol{\omega}) = \|\boldsymbol{\omega}\|^2$$

Hence, for the two-dimensional case the descent direction common to all criteria is $\boldsymbol{\omega} = \gamma\mathbf{v} + (1-\gamma)\mathbf{w}$, where the coefficients for the convex combination are $1-\gamma$ and γ as in expression (2.11).

$$\gamma = \begin{cases} \dfrac{\mathbf{w}\cdot(\mathbf{w}-\mathbf{v})}{\|\mathbf{v}-\mathbf{w}\|^2} & , \quad \mathbf{v}\cdot\mathbf{w} < \Big(\min\{\|\mathbf{v}\|, \|\mathbf{w}\|\}\Big)^2 \\ 0 \text{ or } 1 & , \quad \text{otherwise, depending whether } \min\{\|\mathbf{v}\|, \|\mathbf{w}\|\} = \|\mathbf{v}\| \text{ or } \|\mathbf{w}\| \end{cases} \tag{2.11}$$

Remark 2.10. If the gradients $\mathbf{u}_i$'s are not normalized, that is $S_i = 1 \ \forall i = 1, \dots, n$, the direction of the minimial-norm element $\boldsymbol{\omega}$ is expected to be mostly influenced by the elements $\mathbf{u}_i$, $i = 1, \dots, n$ which have smaller norms. The two-dimensional case in figure 2.1 provides a simple visualization of that phenomenon.

Since during the optimization procedure, the gradient vectors with small values of the norm are usually associated with criteria that have already achieved a fair degree of convergence, the utility of considering these directions for a well-balanced multiobjective optimization is questionable. This observation points out the necessity of properly setting the scaling factors S_i's in order to construct a balanced strategy without excessively increasing the computational cost.

2.2.2 Line search for the optimal step size

In multiobjective optimization, computing a step that improves all criteria and gives significant evolution to the problem is not a trivial task. In general, an adaptive method to compute the best step at every iteration would be convenient.

If the design point $\mathbf{Y}$ is not Pareto-stationary, the Fréchet derivatives of all criteria are strictly negative (and equal if $\boldsymbol{\omega} \in \mathcal{U}$). We proceed by defining $\forall i = 1, \dots, n$ the functions $j_i : \mathbb{R} \to \mathbb{R}$ such that

$$j_i(\rho) = J_i(\mathbf{Y} - \rho\boldsymbol{\omega})$$

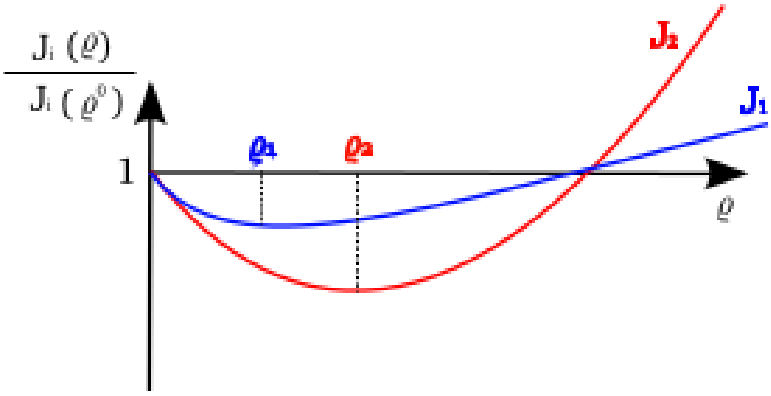


Figure 2.2: Line search to identify the best step size ρ along the common descent direction $-\boldsymbol{\omega}$ such that every criterion improves. Figure from [ZDD11].

Basic idea is to identify the optimal step size ρ such that ρ is the largest strictly positive real number for which all functions $j_i(\rho)$, $i = 1, \ldots, n$ are monotone descreasing over the interval $[0, \rho]$.
For this purpose each function $j_i(\rho)$ is evaluated at three different values of ρ and a surrogate quadratic model is constructed. The optimal step size ρ_i related to $j_i(\rho)$ assumes the value corresponding to the location of the minimum in the surrogate model. Then the smallest ρ_i is set as the global optimal step $\widetilde{\rho}$ as illustrated in figure 2.2:

$$\widetilde{\rho} = \min_{1 \leq i \leq n} \rho_i$$

Thanks to the definition of $\boldsymbol{\omega}$, the positivity of the steps ρ_i and $\widetilde{\rho}$ is guaranteed.

2.3 Alternative formulations of MGDA

In last years J.-A. Désidéri [Dés12b] proposed several variants of *Multiple-Gradient Descent Algorithm* focusing on the improvement of different aspects. He mainly focused on optimizing the research of the descent direction common to all criteria and on studying the sensitivity of the method to different initializations. In particular, main goal was to identify the best basis to span the optimization space starting from the gradient of the objective functionals.
Here we only report main ideas that could lead to improved formulations of the described algorithm. First of all, the author observed that in the construction of the set $\overline{\mathcal{U}}$ and in the identification of the descent direction, smaller gradient vectors have major influence so the idea is to rescale the gradients by means of appropriate factors in order to normalize them.
The construction of an orthogonal basis starting from the $\mathbf{u}_i$'s represents a major issue in the performance of the algorithm; moreover the sensitivity of the method to the ordering of the vectors before the execution of the Gram-Schmidt procedure is still an open question. Here we present a brief overview of the major results and we refer to [Dés12a] and [Dés12b] for further details.

2.3.1 A direct method for computing a descent direction

Starting from remark 2.10, we consider a given set of strictly positive scaling factors S_i $i = 1, \dots, n$ to properly normalize the gradient vectors. Thus we arbitrarly choose a gradient vector as starting point, for example $\mathbf{u}_1$ and we scale it as follows

$$\mathbf{u}_1^{new} = \frac{\mathbf{u}_1}{A_1} \quad , \quad A_1 = S_1$$

and we recursively construct the remaining elements by performing a Gram-Schmidt orthogonalization process

$$\mathbf{u}_i^{new} = \frac{\mathbf{u}_i - \sum_{k<i} c_{i,k} \mathbf{u}_k^{new}}{A_i} \tag{2.12}$$

where $\forall k < i$ the coefficients $c_{i,k}$ are such that

$$c_{i,k} = \frac{(\mathbf{u}_i \ , \ \mathbf{u}_k^{new})}{(\mathbf{u}_k^{new} \ , \ \mathbf{u}_k^{new})}$$

and the scaling factors A_i have the form

$$A_i = \begin{cases} S_i - \sum_{k<i} c_{i,k} & \text{, if non-zero} \\ \epsilon_i S_i & \text{, otherwise} \end{cases}$$

being $\epsilon_i > 0$, $0 < |\epsilon_i| \ll 1$ an arbitrary small parameter.
Thus the computation of the minimal-norm element $\boldsymbol{\omega}$ in the convex hull defined by the orthogonal vectors $\mathbf{u}_i^{new}$, $i = 1, \dots, n$ is straightforward. Thanks to the orthogonality property of the new gradient vectors, we get

$$\|\boldsymbol{\omega}\|^2 = \sum_{i=1}^{n} \alpha_i^2 \|\mathbf{u}_i^{new}\|^2 \tag{2.13}$$

We assume that $\boldsymbol{\omega}$ belongs to the interior of the convex hull; hence by ignoring the inequality constraints over the α_i's which have to be non-negative $\forall i = 1, \dots, n$, the Lagrangian function arising from the constrained optimization problem reads as

$$L(\boldsymbol{\alpha}, \lambda) = \|\boldsymbol{\omega}\|^2 - \lambda \left(\sum_{i=1}^{n} \alpha_i - 1 \right) = \sum_{i=1}^{n} \alpha_i^2 \|\mathbf{u}_i^{new}\|^2 - \lambda \left(\sum_{i=1}^{n} \alpha_i - 1 \right) \tag{2.14}$$

The first-order optimality conditions are

$$\begin{aligned} 0 &= \frac{\partial L}{\partial \alpha_i} = 2\alpha_i \|\mathbf{u}_i^{new}\|^2 - \lambda \quad \forall i = 1, \dots, n \\ 0 &= \frac{\partial L}{\partial \lambda} = \sum_{i=1}^{n} \alpha_i - 1 \end{aligned} \tag{2.15}$$

From the first line of (2.15) we get

$$\alpha_i = \frac{\lambda}{2\|\mathbf{u}_i^{new}\|^2} \tag{2.16}$$

and from the equality constraint in the second line:

$$1 = \sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \frac{\lambda}{2\|\mathbf{u}_i^{new}\|^2} \quad \Rightarrow \quad \frac{\lambda}{2} = \frac{1}{\sum_{i=1}^{n} \frac{1}{\|\mathbf{u}_i^{new}\|^2}} \tag{2.17}$$

thus $\forall i = 1, \ldots, n$

$$\alpha_i = \frac{1}{\|\mathbf{u}_i^{new}\|^2 \sum_{j=1}^{n} \frac{1}{\|\mathbf{u}_j^{new}\|^2}} = \frac{1}{1 + \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\|\mathbf{u}_i^{new}\|^2}{\|\mathbf{u}_j^{new}\|^2}} < 1 \tag{2.18}$$

The result in equation (2.18) confirms our initial conjecture on $\boldsymbol{\omega}$ belonging to the interior of the convex hull. Thanks to theorem 2.5, $\forall \mathbf{u} \in \overline{\mathcal{U}}$ the scalar product $(\mathbf{u}\ ,\ \boldsymbol{\omega})$ is proved to be equal to the constant value $\|\boldsymbol{\omega}\|^2$; moreover, by inserting (2.16) and (2.17) in (2.13) we get

$$\|\boldsymbol{\omega}\|^2 = \frac{\lambda}{2}$$

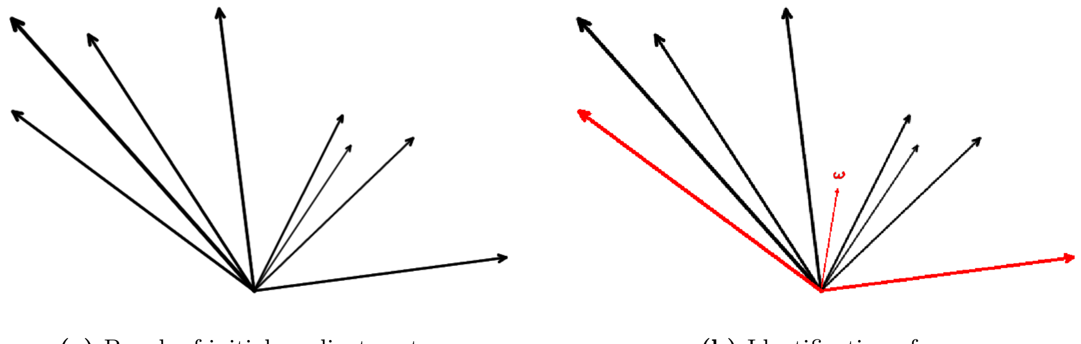
Thus for all functionals J_i's the Fréchet derivatives read as

$$(\mathbf{u}_i\ ,\ \boldsymbol{\omega}) = \left(A_i \mathbf{u}_i^{new} + \sum_{k<i} c_{i,k} \mathbf{u}_k\ ,\ \boldsymbol{\omega} \right) = \left(A_i + \sum_{k<i} c_{i,k} \right) \frac{\lambda}{2} = S_i \frac{\lambda}{2} \tag{2.19}$$

Remark 2.11. The proposed Gram-Schmidt orthogonalization process yields to a different characterization of the subspace spanned by the original gradient vectors but allows an explicit computation of the minimal-norm element $\boldsymbol{\omega}$.
Nevertheless from a numerical point of view, this variant - known as *MGDA II* - may present some stability issues, due either to sign switch of some scaling factors A_i's or to strong correlation or linear dependence among the gradient vectors. Some details are available in [Dés12b] but in these scenarios classical *MGDA* approach seems more robust and reliable.

2.3.2 A Gram-Schmidt procedure to order the vectors

In presence of high-dimensional problems, trends may emerge within the directions of several gradient vectors as shown in figure 2.3. In this scenario, we aim at taking into account the leading direction of a subgroup of vectors by means of a unique element of the orthogonal basis previously constructed by using the Gram-Schmidt orthogonalization process (see remark 2.10). Thus we use only $\widetilde{n} < n$ gradient vectors in order to determine a descent direction common to all criteria. To achieve this purpose the computation of a new orthogonal gradient vector is performed by choosing among those not yet accounted



(a) Bunch of initial gradient vectors. **(b)** Identification of $\boldsymbol{\omega}$.

Figure 2.3: Identification of $\boldsymbol{\omega}$ within a bunch of initial unordered gradient vectors starting from the extreme elements.

for, the gradient for which the scalar product with the current value of $\boldsymbol{\omega}$ is algebrically smallest.
In this way, we select the vector for which the current configuration is the least satisfactory. Hence we expect the overall computational cost to decrease since the order in which the gradient vectors are used for the Gram-Schmidt procedure should lead to a quick interruption of the process.

We consider the usual family of the gradient vectors $\nabla J_i(\mathbf{Y})$ $i = 1, \ldots, n$. Let us introduce a new family $\mathbf{g}_i$ $i = 1, \ldots, n$ and the family of the scaled gradient vectors $\mathbf{J}'_i$ $i = 1, \ldots, n$ defined as

$$\mathbf{J}'_i = \frac{1}{S_i} \nabla J_i(\mathbf{Y}^{(0)})$$

where S_i $i = 1, \ldots, n$ are user-supplied scaling factors.
We set

$$\mathbf{u}_1 = \mathbf{g}_1 = \underset{i=1,\ldots,n}{\operatorname{argmax}} \min_{\substack{j=1,\ldots,n \\ j \neq i}} \frac{(\mathbf{J}'_j, \mathbf{J}'_i)}{(\mathbf{J}'_i, \mathbf{J}'_i)} \tag{2.20}$$

and we build a $n \times n$ lower triangular matrix $C = [c_{i,j}]$ such that the main diagonal contains the cumulative row sums and $c_{i,j} = 0$, $\forall i \geq j$. By looping over the index of the columns, we compute the coefficients of the columns

$$c_{j,i-1} = \frac{(\mathbf{g}_j, \mathbf{u}_{i-1})}{(\mathbf{u}_{i-1}, \mathbf{u}_{i-1})} \qquad \forall j = i, \ldots, n$$

and we update the cumulative row-sums

$$c_{j,j} = c_{j,j} + c_{j,j-1} = \sum_{k=1}^{i-1} c_{j,k} \qquad \forall j = i, \ldots, n$$

In order to compute the new orthogonal vector $\mathbf{u}_i$, first we have to identify the element whose scalar product with current $\boldsymbol{\omega}$ is smallest:

$$\ell = \underset{j=i,\ldots,n}{\operatorname{argmin}}\; c_{j,j}$$

Then we swap the information contained in rows i and ℓ of the matrix C and in the corresponding elements $\mathbf{g}_i$'s and we compute

$$\mathbf{u}_i = \frac{\mathbf{g}_i - \sum_{k=1}^{i-1} c_{i,k} \mathbf{u}_k}{A_i} \quad , \quad A_i = 1 - c_{i,i} \tag{2.21}$$

Remark 2.12. From a numerical point of view, we have to verify that the elements A_i's are non-zero $\forall i = 1, \ldots, n$. For this purpose we fix a cut-off constant $\overline{a}$ such that $c_{j,j} > \overline{a}$, $\forall j = i, \ldots, n$ thus we get $A_i \geq 1 - \overline{a} > 0$.

If the new gradient vector $\mathbf{u}_i$ computed in (2.21) is non-zero, we can proceed by computing the next element of the family of the $\mathbf{u}_i$, $i = 1, \ldots, n$, otherwise we generate the corresponding element $\mathbf{g}_i$ such that

$$\mathbf{g}_i = \sum_{k=1}^{i-1} c_{i,k} \mathbf{u}_k = \sum_{k=1}^{i-1} c'_{i,k} \mathbf{g}_k \tag{2.22}$$

where the coefficients $c'_{i,k}$ are computed by backward substitution.
Thus the computation of the descent direction $\boldsymbol{\omega}$ is reduced to the computation of the

minimal-norm element in the convex hull generated by the gradient vectors $\mathbf{u}_i$, $i = 1, \dots, \widetilde{n}$ where $\widetilde{n} < n$. In particular, we consider all the $\mathbf{u}_i$'s of the form (2.21) such that they are non-zero and the corresponding coefficients α_i's belong to the open interval $(0, 1)$:

$$\boldsymbol{\omega} = \sum_{i=1}^{\widetilde{n}} \alpha_i \mathbf{u}_i \neq 0 \qquad , \qquad \alpha_i = \frac{1}{\|\mathbf{u}_i\|^2 \sum_{j=1}^{\widetilde{n}} \frac{1}{\|\mathbf{u}_j\|^2}} \cdot \frac{1}{1 + \sum_{\substack{j=1 \\ j \neq i}}^{\widetilde{n}} \frac{\|\mathbf{u}_i\|^2}{\|\mathbf{u}_j\|^2}} \tag{2.23}$$

The algorithm stops when a Pareto-stationary design point is achieved, that is if the $c'_{i,k}$'s are non-positive $\forall k = 1, \dots, i-1$. Nevertheless, this variant of *MGDA* - the so-called *MGDA III* - may present some issues if this condition on the coefficients is not fulfilled: if there exists at least one index $\overline{k} \in [1, i-1]$ such that $c'_{i,\overline{k}} > 0$, an ambiguity arises and the descent direction common to all criteria has to be determined by means of classical *Multiple-Gradient Descent Algorithm* as described in section 2.2.1.

Remark 2.13. If $\widetilde{n} = n$ the Gram-Schmidt process is performed completely and the algorithm is equivalent to the second variant of *MGDA* discussed in section 2.3.1. We remark that in the case of the incomplete Gram-Schmidt procedure, that is $\widetilde{n} \neq n$, the Fréchet derivatives of the functionals satisfy different bounds from the ones previously computed.

- The first $\widetilde{n}$ Fréchet derivatives are such that $(\mathbf{g}_i, \boldsymbol{\omega}) = (\mathbf{u}_i, \boldsymbol{\omega}) = \|\boldsymbol{\omega}\|^2 > 0$.
- For the estimate of the remaining elements $\mathbf{g}_i$, $i = \widetilde{n}+1, \dots, n$, we recall that $\boldsymbol{\omega}$ is given by (2.23) thus we get

  $$\mathbf{g}_i = \sum_{k=1}^{\widetilde{n}} c_{i,k} \mathbf{u}_k + \mathbf{v}_i$$

  where the $\mathbf{v}_i$'s are orthogonal to the space $\langle \mathbf{u}_1, \dots, \mathbf{u}_{\widetilde{n}} \rangle$ and consequently the following bound holds

  $$(\mathbf{g}_i, \boldsymbol{\omega}) = \sum_{k=1}^{\widetilde{n}} c_{i,k} (\mathbf{u}_k, \boldsymbol{\omega}) = \sum_{k=1}^{\widetilde{n}} c_{i,k} \|\boldsymbol{\omega}\|^2 = c_{i,i} \|\boldsymbol{\omega}\|^2 > \overline{a} \|\boldsymbol{\omega}\|^2 > 0$$

2.3.3 An Hessian-based approach to properly scale the gradients

Let us assume that the objective functionals $J_1(\mathbf{Y}), \dots, J_n(\mathbf{Y})$ are of class $\mathcal{C}^2$ in order to be able to compute the associated Hessian matrices $H_i(\mathbf{Y})$'s. Starting from the idea of the Newton method for single objective optimization, we aim at extending the concept of preconditioning to realize an optimal scaling of the gradient vectors as follows

$$H_i(\mathbf{Y}) \mathbf{p}_i = \nabla J_i(\mathbf{Y}^{(0)}) \quad , \quad \forall i = 1, \dots, n$$

and then project the resulting vector $\mathbf{p}_i$ over the plane identified by the gradient direction itself:

$$\mathbf{q}_i = \frac{(\mathbf{p}_i, \nabla J_i(\mathbf{Y}^{(0)}))}{\|\nabla J_i(\mathbf{Y}^{(0)})\|^2} \nabla J_i(\mathbf{Y}^{(0)}) \tag{2.24}$$

The final so-called *MGDA IV* strategy corresponds to the previous *MGDA III* procedure where the scaled gradients are defined as in equation (2.24):

$$\mathbf{J}'_i = \mathbf{q}_i \tag{2.25}$$

This is equivalent to defining the scaling constants S_i's as

$$S_i = \frac{\|\nabla J_i(\mathbf{Y}^{(0)})\|^2}{(\mathbf{p}_i, \nabla J_i(\mathbf{Y}^{(0)}))}$$

Remark 2.14. If the Hessian matrices are not known exactly, an approximation may be computed by using an iterative procedure, for example a *Broyden-Fletcher-Goldfarb-Shanno* (*BFGS*) inspired algorithm $\forall i = 1, \dots, n$:

$$\begin{aligned} &\widetilde{H}_i^{(0)} = Id \\ &\widetilde{H}_i^{(k+1)} = \widetilde{H}_i^{(k)} - \frac{1}{\mathbf{s}^{(k)^T} \widetilde{H}_i^{(k)^T} \mathbf{s}^{(k)}} \widetilde{H}_i^{(k)} \mathbf{s}^{(k)} \mathbf{s}^{(k)^T} \widetilde{H}_i^{(k)^T} + \frac{1}{\mathbf{z}_i^{(k)^T} \mathbf{s}^{(k)}} \mathbf{z}_i^{(k)} \mathbf{z}_i^{(k)^T} \end{aligned} \tag{2.26}$$

where k is the index of the *MGDA* iterate and the variables that appear in (2.26) are

$$\mathbf{s}^{(k)} = \mathbf{Y}^{(k+1)} - \mathbf{Y}^{(k)} \quad , \quad \mathbf{z}_i^{(k)} = \nabla J_i(\mathbf{Y}^{(k+1)}) - \nabla J_i(\mathbf{Y}^{(k)})$$

2.4 Metamodel-assisted MGDA optimization

A major drawback of the optimization algorithms described so far is the high computational cost due to the evaluation of the functionals and their gradients in correspondance of a large number of different configurations. In complex problems such the ones arising in Computational Fluid Dynamics or computational mechanics this may result in an excessive demand of computational resources. For this reason in [ZDD12] A. Zerbinati et al. propose to couple *MGDA* with a surrogate model to estimate the objective functionals rather than actually computing them. Basic idea of the global optimization procedure is reported in figure 2.4.

During an initial phase, some simulations are performed for different design points in order to construct a data set of values and consequently formulate a metamodel starting from it. After a training of the data set, the statistical model is used to predict the values of the objective functionals corresponding to a new configuration point that lies in the region where the improvement of the criteria is expected to be the most significant.

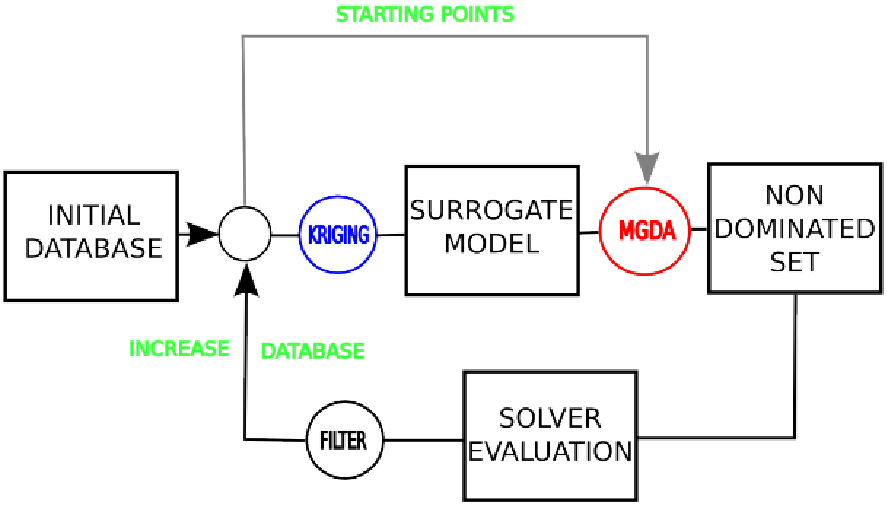


Figure 2.4: Global optimization procedure for metamodel-assisted *MGDA* optimization. Figure from [ZDD12].

The entries of the data set are also used as initial starting points for *MGDA* iterations. The execution of the algorithm is performed until convergence is achieved and the metamodel-predicted values are used as objective functionals in the optimization strategy. Final

MGDA points are non-dominated, thus they belong to the first Pareto front associated with the problem under analysis. Eventually each configuration is reevaluated using the numerical solver of the problem and the resulting entry is added to the data set. In order to mantain the dimension of the problem as small as possible, a filter is applied and new design points are discarded if too close to already existing entries. At completion of the enrichment process, the metamodel is updated and training is performed again in order to refine the information carried by the surrogate model.

Remark 2.15. In our implementation, the control loop for metamodel-assisted *Multiple-Gradient Descent Algorithm* is repeated an *a priori* fixed number of times but more accurate strategies could be considered. For example, introducing an error estimate in a feasible norm could help evaluating the changement of the data set among subsequent iterations thus allowing to stop the refinement procedure when a given tolerance is fulfilled.

2.4.1 Kriging-based metamodel

Here we provide a general overview of kriging-based metamodel for the prediction of the objective functionals in correspondance of a given design point. For more details on this topic and on the application of kriging models to optimization procedures we refer to [DC12]. To improve the readibility, given a functional $J_i(\Omega)$ we define $j = J_i(\mathbf{Y})\ ,\ \mathbf{Y} \in \Omega$ as the function that assumes the values of one of the criteria to optimize.

Basic idea of kriging models relies on treating the response of an experiment (in our case, the value of the objective functional $J_N = \{j_1, \dots, j_N\}$ computed in correspondance of a set of design points $\{\mathbf{Y}^{(1)}, \dots, \mathbf{Y}^{(N)}\}$) as if it were a realization of a multivariate Gaussian stochastic process with joint probability density

$$\mathbb{P}(J_N) = \frac{1}{\sqrt{(2\pi)^N |\Sigma_N|}} exp\left\{ -\frac{1}{2} J_N^T \Sigma_N^{-1} J_N \right\} \tag{2.27}$$

where Σ_N is the $N \times N$ covariance matrix that expresses the correlation among the realizations associated to different design points. We suppose that the value J_{N+1} of the objective functional obtained when adding a new design point $\mathbf{Y}^{(N+1)}$ to the model is itself a realization of a $(N+1)$-dimensional Gaussian process with joint probability density equal to the one arising from substituting $N+1$ to N in equation (2.27).
Thus we can write the probability density for the unknown function value j_{N+1} given the data J_N as

$$\mathbb{P}(j_{N+1} \mid J_N) = \frac{\mathbb{P}(j_{N+1} \cap J_N)}{\mathbb{P}(J_N)} = \frac{\mathbb{P}(J_{N+1})}{\mathbb{P}(J_N)}$$

We perform an appropriate splitting of the covariance matrix Σ_{N+1} as follows

$$\Sigma_{N+1} = \begin{bmatrix} \Sigma_N & \mathbf{k} \\ \mathbf{k}^T & \kappa \end{bmatrix}$$

where

$$\mathbf{k} = [\sigma(\mathbf{Y}^{(1)}, \mathbf{Y}^{(N+1)}), \dots, \sigma(\mathbf{Y}^{(N)}, \mathbf{Y}^{(N+1)})] \quad , \quad \kappa = \sigma(\mathbf{Y}^{(N+1)}, \mathbf{Y}^{(N+1)}) \tag{2.28}$$

Then we introduce the concept of prediction of the kriging model at the new point $\mathbf{Y}^{(N+1)}$ as

$$\widehat{j}_{N+1} = \mathbf{k}^T \Sigma_N^{-1} J_N$$

whereas to measure the uncertainty of the prediction we define the variance

$$\sigma^2_{N+1} = \kappa - \mathbf{k}^T \Sigma_N^{-1} \mathbf{k}$$

We obtain that the probability density for the new functional value is given by

$$\mathbb{P}(j_{N+1} \mid J_N) \propto exp\left\{ -\frac{(j_{N+1} - \widehat{j}_{N+1})^2}{2\sigma^2_{N+1}} \right\} \tag{2.29}$$

Thus the probability density for the objective functional j at the new design point $\mathbf{Y}^{(N+1)}$ is also Gaussian with mean $\widehat{j}_{N+1}$ and standard deviation σ_{N+1}. We remark that this statistical approach introduces an additional error in the evaluation of the functional and this uncertainty is taken into account by the estimation of the standard deviation σ_{N+1}.

From a practical point of view, the construction of the kriging model counts two major steps. First, a function that describes the dependency among the data has to be determined by estimating the spatial correlation with respect to a particular stochastic process. In the literature, all the proposed models depend on a set of parameters that have to be properly calibrated in order to make the description the most consistent with the observed data. In particular, this results in the minimization of the log-likelihood function

$$\mathcal{L} = J_N^T \Sigma_N^{-1} J_N + \log |\Sigma_N|$$

by means of a *Particle Swarm Optimization* technique ([VSS05]) that is robust and not sensitive to local minima which could represent an issue since $\mathcal{L}$ is a multi-modal function. Then the iterative prediction model is constructed as previously described in this section and further details are available in [Cre93]. The predicted value is constructed from the linear combination of the previously known functional values; the parameters arising in this formulation are obtained by minimizing the mean square error under the constraint of unbiasedness. Several variants of kriging models have been proposed during the years and we refer to [Wil97] for further details.

Remark 2.16. Kriging models may gain in accuracy if additional information is introduced by means of a set of adjoint variables, leading to the so-called co-kriging models. In particular, in the literature several solutions have been proposed in order to enclose the information carried by the gradients of the functionals into the original kriging model to make the description even more consistent with the observed data.
A general introduction to co-kriging is available in [Mye82] whereas we refer to [AC02] for a specific description of the use of gradients to build co-kriging strategies.

Eventually, in table 2.1 we present a comparative summary of the *MGDA*-based optimization algorithms described so far, highlighting their specifics and their properties.

2.5 MGDA for constrained optimization

The formal theory presented so far is true under the assumption of dealing with an unconstrained optimization problem. In presence of constraints, previous results can be extended by means of minor changes. Basic idea is projecting $\forall i = 1, \dots, n$ the gradients $\mathbf{u}_i$ of the functionals J_i's onto the subspace tangent to the constraint surfaces.

We consider a constrained optimization problem of the following form

$$\min_{\mathbf{Y} \in \mathcal{U}_{ad}} \mathbb{J}(\mathbf{Y}) \qquad , \qquad \mathbb{J}(\mathbf{Y}) = (J_1(\mathbf{Y}), \dots, J_n(\mathbf{Y}))^T \tag{2.30}$$

Algorithm	Specifics	Properties
MGDA	- Construction based on the identification of the minimal-norm element in the convex hull of the gradients	- Convergence to Pareto-stationary points - Insensitivity to Pareto front convexity
Kriging-assisted *MGDA*	- Coupling with a statistical model to reduce computational costs	- Works on a set of design points - Iterative enrichements of the data set - More efficient (Lower CPU time)
MGDA II	- Direct computation of a descent direction (*GSP*) - Possibility of automatic rescaling if $\exists i \ : \ S_i < 0$	- Applies only to linearly independent gradients - More efficient (Larger Fréchet derivatives) - $n!$ possible orderings
MGDA III	- Specific ordering in *GSP* - Incomplete *GSP* - Fallback to *MGDA* if ambiguties arise	- Not limited to linearly independent gradients - More efficient (Even larger Fréchet derivatives)
MGDA IV	- Scaling driven by Hessian matrices - *BFGS* to approximate Hessian matrices when analytical computation is not possible	- Higher rate of convergence

Table 2.1: Comparison of several variants of *Multiple-Gradient Descent Algorithm* presented so far.

where the admissible domain $\mathcal{U}_{ad}$ is given by the restriction to the subspace of $\mathbb{R}^N$ where the equations of the constraints are satisfied:

$$\mathcal{U}_{ad} = \left\{ \Omega \subset \mathbb{R}^N \;\middle|\; g_k(\mathbf{Y}) = 0 \quad \forall k = 1, \dots, \widetilde{K} \quad \forall \mathbf{Y} \in \Omega \right\} \tag{2.31}$$

Suppose that the active scalar constraints at $\mathbf{Y} = \mathbf{Y}^{(0)}$ are

$$g_k(\mathbf{Y}) = 0 \quad , \quad k = 1, \dots, K \leq \widetilde{K}$$

thus we identify the direction orthogonal to the surface $\mathcal{P}$ defined by equation $g_k(\mathbf{Y}) = 0$ as

$$\mathbf{v}_k = \nabla g_k(\mathbf{Y}) \quad , \quad k = 1, \dots, K \tag{2.32}$$

By applying the Gram-Schmidt orthogonalization process, we get the family of the $\mathbf{w}_k$, $k = 1, \dots, K$ of orthogonal vectors that collectively span the same subspace. After a proper normalization procedure we obtain that the projection onto the orthogonal space is given by the operator

$$Q : \mathbb{R}^N \to \mathbb{R}^N \quad , \quad Q = \sum_{k=1}^{K} [\mathbf{w}_k][\mathbf{w}_k]^T$$

where the bracketed vector stands for the column vector of its components viewed as a $N \times 1$ matrix. Thus we can define the projection matrix from $\mathbb{R}^N$ to the subspace $\mathcal{U}_{ad}^{\mathbf{Y}}$ of the admissible solutions considering the active constraints at $\mathbf{Y} = \mathbf{Y}^{(0)}$ as follows

$$P : \mathbb{R}^N \to \mathbb{R}^N \quad , \quad P = I_N - \sum_{k=1}^{K} [\mathbf{w}_k][\mathbf{w}_k]^T \tag{2.33}$$

Hence, previous *MGDA* procedures are modified by replacing the original scaled gradients $\mathbf{u}_i$'s with their projections onto the subspace tangent to the constraint surfaces, that is $P\mathbf{u}_i$'s. Then the projected gradients $P\mathbf{u}_i$, $i = 1, \ldots, n$ are used to find the minimal-norm element $\boldsymbol{\omega} \in \overline{\mathcal{U}}$ which is now a descent direction for all criteria $J_i(\mathbf{Y})$, $i = 1, \ldots, n$ subject to the active constraints $g_k(\mathbf{Y}) = 0$, $k = 1, \ldots, K$.

Chapter 3

Shape optimization using IsoGeometric Analysis

In this chapter we introduce the equations governing a problem of linear elasticity and we present the *IsoGeometric Analysis* paradigm to provide an accurate parametrization of the geometry of the domain and an efficient approximation of the differential problem. After a brief introduction to *Non-Uniform Rational B-Splines* we provide the discrete formulation for the approximation of the linear elasticity problem within this framework. We refer to [PT97] as general reference in the literature for *NURBS* whereas more details on the numerical approximation of PDEs within that framework are available in [HCB05].

Moreover we present a shape optimization problem arising in structural engineering and we describe the analytical and functional framework in which the analysis has to be set. Several approaches to the numerical solution of the problem are proposed, arising both from classical methods in optimization and from gradient-descent algorithms: in particular, first we present the approaches based on penalty formulations and Lagrange multipliers ([NW99]); then we focus on descent methods arising from shape derivatives, that is techniques of differentiation with respect to the domain ([All06] and [BDVS12]).

Eventually we propose to set the shape optimization problem within a multiobjective optimization framework. Thanks to the *NURBS*-based approach, we get the same characterization for both the space of parameters Ω of the optimization problem and the domain D of the differential problem in continuum mechanics. In particular, the domain D can be represented as a *NURBS* surface whose control points are the design variables for the shape optimization problem under analysis. Thus from now on, we refer to Ω as both the physical domain of the linear elasticity problem and the variable of the shape optimization problem: we provide the general formulation of the overall resulting problem and we analyze it by using *Multiple-Gradient Descent Algorithm* (Chapter 4).

3.1 Linear elasticity problem in structural engineering

We consider an open domain $\Omega \subset \mathbb{R}^d$ $d = 2, 3$ which describes the solid object we are analyzing whose boundary $\partial\Omega$ is composed by three disjoint parts Γ_D, Γ_N and Γ such that $\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$, $\Gamma_N \cap \Gamma = \emptyset$ and $\Gamma \cap \Gamma_D = \emptyset$.
This object is deformable and subject to external forces: in particular, a Dirichlet boundary condition representing imposed displacements is prescribed on Γ_D; on Γ_N we impose the value for the stress tensor by means of a Neumann condition whereas Γ is a free boundary where we prescribe a homogeneous Neumann boundary condition. Thus Γ is a moving

boundary within our domain and from now on we will treat it as the optimization variable for the shape optimization problem.

We introduce the second-order Cauchy stress tensor $\boldsymbol{\sigma}(\mathbf{u})$ which assumes the following formulation based on Hooke's elasticity law

$$\boldsymbol{\sigma}(\mathbf{u}) = 2\mu\boldsymbol{\epsilon}(\mathbf{u}) + \lambda tr(\boldsymbol{\epsilon}(\mathbf{u}))\mathbf{I}d \tag{3.1}$$

In this context, $\mathbf{u}$ represents the displacement field, $\boldsymbol{\epsilon}(\mathbf{u})$ the Green-Lagrange strain tensor and μ and λ are the Lamé parameters of the material.
The general form of the strain tensor is

$$\boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2}\Big(\nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}^T \cdot \nabla\mathbf{u}\Big)$$

but under the assumption of small deformations a linearization is possible:

$$\boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2}\Big(\nabla\mathbf{u} + \nabla\mathbf{u}^T\Big)$$

The governing equations for our problem are the classical linear elasticity equations based on the assumption of small deformations ([All06]). For the sake of simplicity we assume zero distributed body forces, focusing our study on the static equilibrium of an isolated system:

$$\begin{cases} -div\boldsymbol{\sigma}(\mathbf{u}) = \mathbf{0} & , \ \Omega \\ \mathbf{u} = \mathbf{0} & , \ \Gamma_D \\ \boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} = \mathbf{g} & , \ \Gamma_N \\ \boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} = \mathbf{0} & , \ \Gamma \end{cases} \tag{3.2}$$

where $\mathbf{n}$ is the outward unit normal vector.

3.1.1 Variational formulation

In this section we introduce the variational formulation of the linear elasticity problem (3.2) and we state a general result for existence and uniqueness of the weak solution.
Let us introduce the functional space

$$V = \{\boldsymbol{\varphi} \in (H^1(\Omega))^d \ , \ \boldsymbol{\varphi} = \mathbf{0} \ on \ \Gamma_D\}$$

The variational form of problem (3.2) reads as follows: we seek a displacement field $\mathbf{u} \in V$ such that

$$\int_\Omega (2\mu\boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{v}) + \lambda div(\mathbf{u}) div(\mathbf{v}))d\omega = \int_{\Gamma_N} \mathbf{g}\cdot\mathbf{v}d\sigma \qquad \forall\mathbf{v} \in V \tag{3.3}$$

A result of existence and uniqueness of the solution for the variational linear elasticity problem can be established and we report here the statement:

Theorem 3.1. *Let* $\mathbf{g} \in L^2(\Gamma_N)$. *If the measure of* Γ_D *is positive, then there exists a unique solution* $\mathbf{u} \in V$ *for the variational form of the linear elasticity problem. Moreover, it holds*

$$\|\nabla\mathbf{u}\|_{L^2(\Omega)} \leq \frac{C}{\mu}\|\mathbf{g}\|_{L^2(\Gamma_N)} \tag{3.4}$$

The proof is based on the classical framework of Lax-Milgram theorem and to verify the coercivity we use Korn's inequality. Further details are available at [Hor95] and [Sal08].

Remark 3.2. The variational formulation of the linear elasticity problem corresponds to the virtual work principles where the test function $\mathbf{v}$ in equation (3.3) is a virtual displacement. The weak solution $\mathbf{u}$ is also solution of the minimization problem

$$\min_{\mathbf{v}\in V} E(\mathbf{v}) \quad , \quad E(\mathbf{v}) = \frac{1}{2}\int_{\Omega} (2\mu|\boldsymbol{\epsilon}(\mathbf{v})|^2 + \lambda|div(\mathbf{v})|^2) d\omega$$

where the functional E represents the deformation energy. Thus the variational formulation of the linear elasticity problem corresponds to the Euler equation for the functional E and the weak solution $\mathbf{u}$ is equal to the equilibrium of the deformation energy among all possible displacement fields.

3.2 Introduction to Non-Uniform Rational B-Splines

Let Ω_0 be a parametric domain. *NURBS* basis functions are defined in Ω_0 as functions of the variable $\boldsymbol{\xi}$ and can be represented in the physical domain Ω by introducing a transformation F that maps Ω_0 to Ω (Fig. 3.1).

$$F : \Omega_0 \to \Omega \qquad , \qquad F(\boldsymbol{\xi}) = \mathbf{y}(\boldsymbol{\xi}) \tag{3.5}$$

Here we describe the formal derivation of the basis functions in one dimension and the extension to the d-dimensional case is straightforward by means of d-variate tensor products.

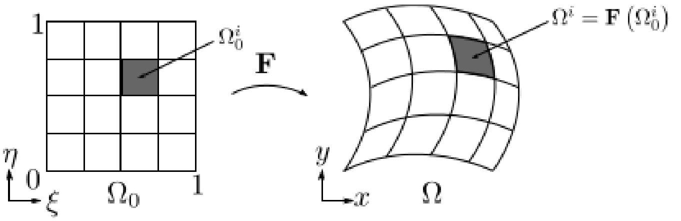


Figure 3.1: Transformation map F from the parametric domain Ω_0 to the physical domain Ω. Figure from [BDVS12].

3.2.1 B-Spline basis functions and geometry representation

Let us define a *knot vector* $\Xi = (\xi_0, \ldots, \xi_a)^T \in \mathbb{R}^a$, $a = n + p + 1$: it consists of non-decreasing real numbers which describe the general geometrical structure of the curve. Using this notation, p is the polynomial *order* of the basis functions and n is the number of functions considered.

In the literature ([PT97]) $p = 0$ stands for piecewise constant polynomials, $p = 1$ for linear ones and so on. To make a parallelism with Finite Element Method, B-Splines order corresponds to Lagrangian basis degree whereas *patches* play the role of subdomains like *FEM* elements.

Definition 3.3 (*Uniform knots* and *non-uniform knots*)**.** Equally-spaced knots in the domain Ω_0 are said to be *uniform.* On the contrary we name *non-uniform* the knots that are unequally-spaced in the domain.

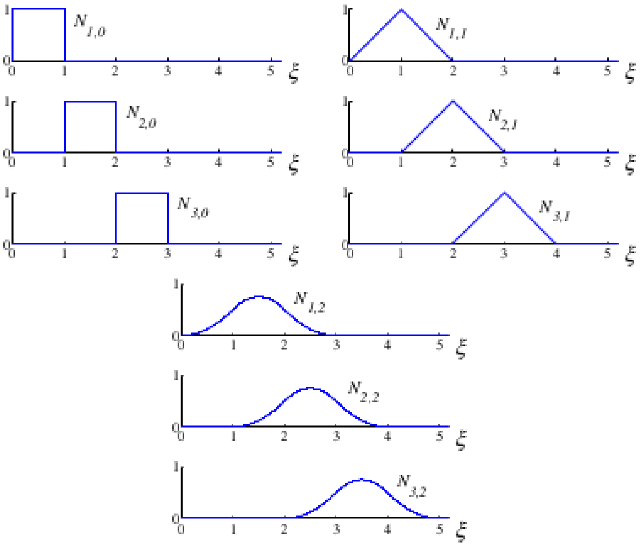


Figure 3.2: *NURBS* basis functions of order $0, 1, 2$ for uniform knot vector $\Xi = \{0, 1, 2, 3, 4, \ldots\}$. Figure from [HCB05].

Definition 3.4 (*Repeated knots* and *open knot vector*)**.** Knots located at the same coordinates in the parametric space are known as *repeated* knots. A knot vector is said to be *open* if its first and last knots are repeated $p+1$ times.

Remark 3.5. Open knot vectors are tipically used in *CAD* applications. Main advantage consists in basis functions being interpolatory at the ends of the parametric space interval (in several dimension this property equals to being interpolatory at the corners of the patches). We remark the importance of this property since in general basis functions are not interpolatory at interior nodes which is one of the main differences among approaches based on *IsoGeometric Analysis* and Finite Element Method.

We can introduce B-Spline basis functions starting from the following expression for the piecewise constant one ($p = 0$)

$$N_{i,0}(\xi) = \begin{cases} 1 \ , & \xi_i \leq \xi < \xi_{i+1} \\ 0 \ , & \text{otherwise} \end{cases} \tag{3.6}$$

and then we recursively define them for p as

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{3.7}$$

where the quotient $0/0$ is assumed to be zero.

In figure 3.2 we observe that for $p = 0$ and $p = 1$ B-Spline basis functions are the same as for standard piecewise constant and linear Finite Element. Increasing the order we notice that all basis functions are identical and they only are shifted along the knot vector giving a homogeneous pattern to the spatial representation of the domain.

In general, basis functions of order p have $p - 1$ continuous derivatives and if a knot is repeated k times, the regularity of the function decreases generating a function of class $\mathcal{C}^{p-k}$ in correspondance of that specific knot. When the multiplicity of a knot is exactly p the basis function is interpolatory: in figure 3.3 knot number 4 has multiplicity $k = 2$

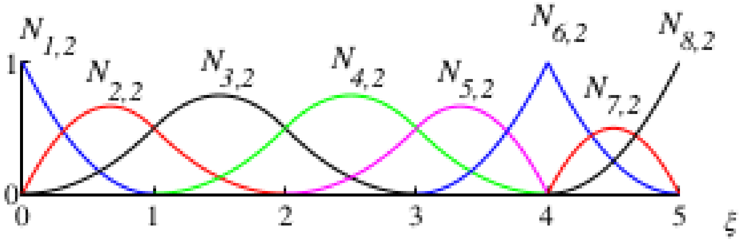


Figure 3.3: Quadratic *NURBS* basis functions for open non-uniform knot vector $\Xi = \{0,0,0,1,2,3,4,4,5,5,5\}$. Figure from [HCB05].

s and the resulting curve parametrized using quadratic B-splines is of class $\mathcal{C}^0$ whereas in the other internal knots the regularity is $\mathcal{C}^1$.
Eventually we outline some major properties of B-Spline basis functions:

1. They constitute a partition of unity: $\forall \xi \ , \ \sum_{i=1}^{n} N_{i,p}(\xi) = 1$.
2. The support of $N_{i,p}$ is compact and contained in the interval $[\xi_i, \xi_{i+p+1}]$.
3. Each basis function is non-negative, meaning $N_{i,p}(\xi) \geq 0 \ \forall \xi$; thanks to this property all coefficients of a mass matrix computed from a B-Spline basis are non-negative.

Remark 3.6. The partition of unity represents the most important property for the B-splines since it allows to construct a basis for the functions of this parametrization. Moreover the compact supports of the basis functions allow to assemble sparse matrices in the algebraic formulation of the differential problem. However we notice that the matrices have a less sparse pattern than the ones arising from classical Finite Element Method. From a computational point of view, this results in lower efficiency of the numerical solver but this drawback is compensated by the lower number of degrees of freedom necessary for a detailed description of the geometry.

3.2.2 Grid generation and refinement strategies

In this section we briefly introduce the concepts of knot insertion and order elevation to refine the representation of B-spline curves. For further details we refer to [HCB05] and to the more extensive introduction to *NURBS* and computational geometry by L. Piegl and W. Tiller ([PT97]).

h-refinement: knot insertion

The process of knot insertion described in figure 3.4 is analogue to Finite Element Method h-refinement. Starting from a knot vector $\Xi = (\xi_0, \ldots, \xi_a)^T$, we want to add a new knot $\overline{\xi}$ within the interval $[\xi_k, \xi_{k+1})$. We remark that knots have to be inserted without geometrically or parametrically changing a curve and here we present the process to correctly update the knot vector, the control points and the basis functions in order to leave the curve intact.

The knot vector resulting from previous insertion reads as

$$\Xi^{new} = (\xi_1, \ldots, \xi_k, \overline{\xi}, \xi_{k+1}, \ldots, \xi_a)^T$$

and by using the geometrical information in Ξ^{new} the new $n+1$ basis functions are recursively constructed by means of (3.6) and (3.7). At the same time, the new $n+1$ control points $\overline{B} = (\overline{B}_1, \ldots, \overline{B}_{n+1})^T$ are generated as linear combination of the previous ones according to expression (3.8):

$$\overline{B_i} = \alpha_i B_i + (1-\alpha_i) B_{i-1} \tag{3.8}$$

where the coefficients α_i's are

$$\alpha_i = \begin{cases} 1 \ , & 1 \le i \le k-p \\ \frac{\overline{\xi}-\xi_i}{\xi_{i+p}-\xi_i} \ , & k-p+1 \le i \le k \\ 0 \ , & k+1 \le i \le a+1 \end{cases}$$

To preserve the continuity of the curve, the control points have to be chosen as in (3.8); repeated knots cause the regularity of the curve to decrease and a unique internal knot cannot appear more than p times if we want to avoid the curve to become discontinuous.

The h-refined curve is geometrically and parametrically identical to the original one but the solution space is enriched with more basis functions: as a matter of fact, the insertion of a knot is responsible for the insertion of a new control point and the creation of a new basis function. Thus the subdivision process generated by the knot insertion is analogous to the classical Finite Element Method h-refinement strategy.

p-refinement: order elevation

As for h-refinement, the geometrical and parametrical properties of B-Spline curves have to be preserved if the polynomial order of the basis functions is increased. Thus the space spanned by the original functions is contained in the span of the basis functions of the order-elevated case. To leave the parametrization of the curve intact, first of all we perform a subdivision of the curve into Bézier curves by inserting new knots; then we elevate every individual segment and we remove the unnecessary knots to combine the segments into one order-elevated B-spline curve. The result of this procedure is presented in figure 3.5.

We remark that for actually preserving the geometry of the curve - that is the regularity in every knot - each unique knot value ξ_i has to be conveniently repeated. For this reason, the number of control points after the refinement depends on the multiplicity of the original knots. The location of the control points changes from the original curve to the order-elevated one and the results are different from the ones obtained from previous h-refinement strategy. Hence, T.J.R. Hughes et al. proposed to properly combine the approaches of knot insertion and order elevation to obtain the so-called k-refinement strategy.

k-refinement

In [HCB05] the authors notice that the processes of knot insertion and order elevation do not commute. If we consider a unique knot value $\overline{\xi}$ to be inserted in a curve of order p, the basis functions at this point will be of class $\mathcal{C}^{p-1}$; elevating the order of the new curve to q, the multiplicity of every knot is increased so that the regularity properties in the curve are preserved, that is in $\overline{\xi}$ the basis remains of class $\mathcal{C}^{p-1}$. On the contrary, by elevating the order of the original curve to q and inserting a unique knot value, then the basis would have $q-1$ continuous derivatives at $\overline{\xi}$. The latter procedure is known as k-refinement and there is no such strategy in Finite Element Method since h-refinement and p-refinement commute in that framework.

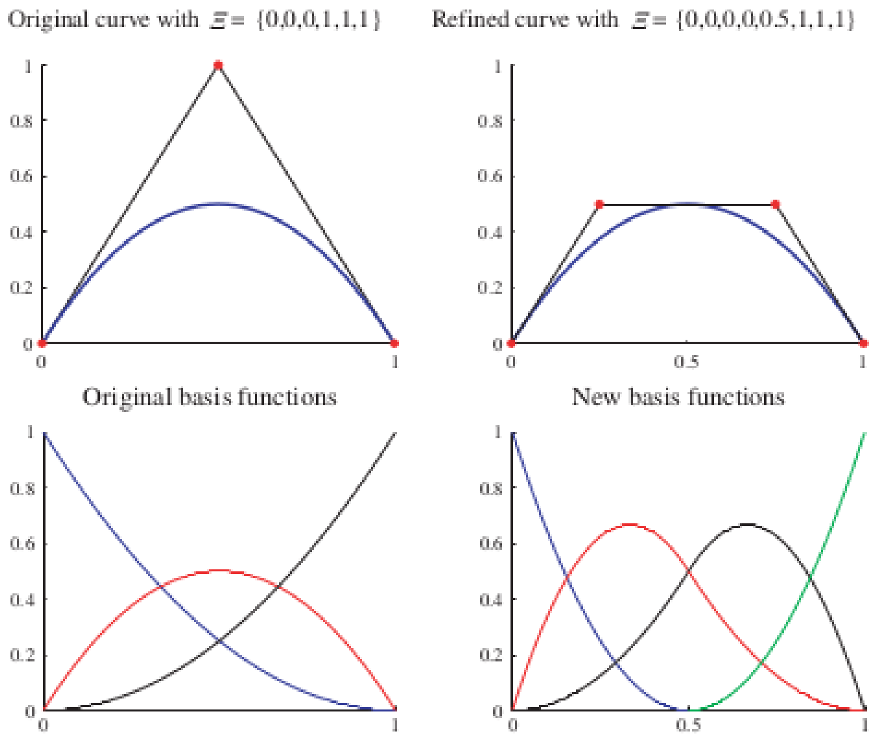


Figure 3.4: Algorithm for h-refinement: knot insertion and resulting changes in the control points. Figure from [HCB05].

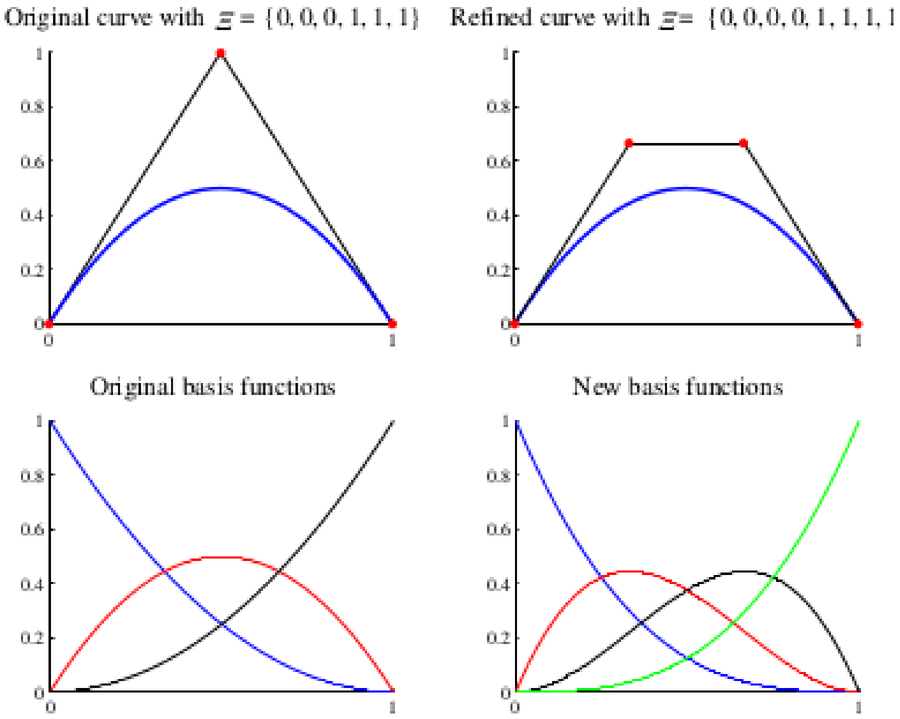


Figure 3.5: Algorithm for p-refinement: order elevation and resulting changes in the control points. Figure from [HCB05].

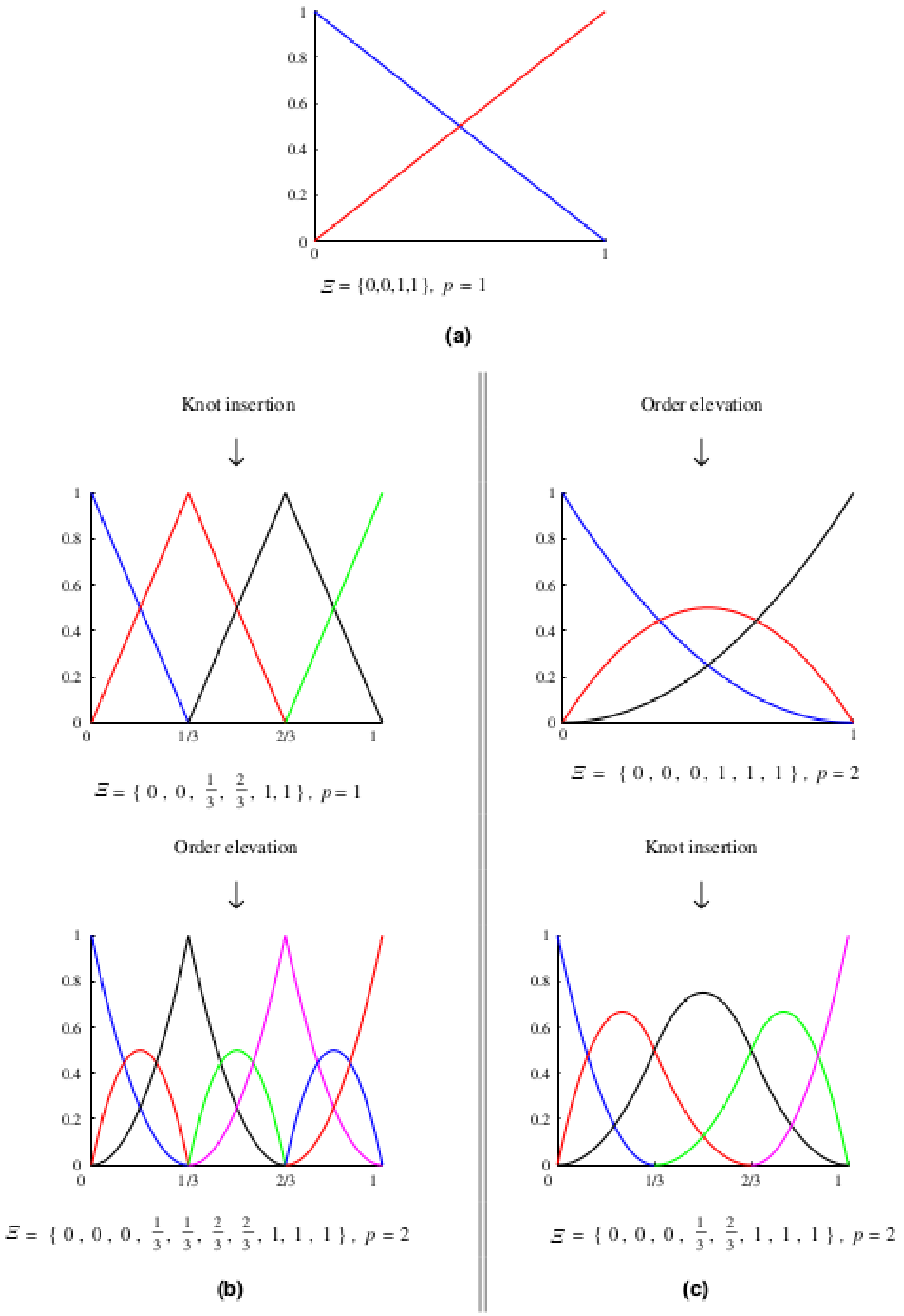


Figure 3.6: Algorithms for refinement strategies: comparison between hp-refinement (b) and k-refinement (c). Figure from [HCB05].

In figure 3.6 we present a comparative study by T.J.R. Hughes et al. of the number and the regularity of the basis functions when either a hp-refinement or a k-refinement is preformed. In particular, we consider an initial curve with one non-zero knot span in an open knot vector and $p+1$ basis functions. We perform h-refinement until we have $n-p$ elements and n basis functions; then we elevate the order mantaining continuity at $p-1$ level: for this purpose, we have to replicate each distinct knot and consequently add a basis function in each element. Hence the resulting number of basis functions is $2n-p$. After a total of r p-refinements we get $(r+1)n-rp$ basis functions of class $\mathcal{C}^p$. Thus the number of functions that have to be considered is very large, especially in practical applications where the number of elements is by far larger than the order of the basis.
Now, we consider the same initial domain and we proceed by k-refinement. We order elevate r times adding one basis function at each refinement and then we perfom h-refinement until we get $n-p$ elements. The resulting scenario counts $n+r$ basis functions, each of class $\mathcal{C}^{r+p-1}$. The overall computational saving in terms of degrees of freedom is substantial since $(r+1)n-rp \ll n+r$. Moreover, when dealing with problems in d dimensions, these quantities are raised to the d power and this difference increases even more.

Hence in k-refinement there is a homogeneous structure within the patches and the number of control variables inserted during the refinement process is limited. Moreover, other advantages may arise due to the higher regularity of the functions involved in the computation: for example, more accurate representations of physical quantities are expected and this results in better descriptions of boundary layers in Computational Fluid Dynamics and strains and stresses in computational mechanics.

3.2.3 Non-Uniform Rational B-Splines representation

In general, a geometric object in $\mathbb{R}^d$ can be obtained by means of a projective transformation of a B-spline entity embedded in a $d+1$ space. As a matter of fact, starting from piecewise quadratic curves we can represent circles and ellipses using projective transformations. From an analytical point of view, by applying a projective transformation to a B-spline curve we get a rational polynomial

$$C_R(\xi) = \frac{f(\xi)}{g(\xi)}$$

where f and g are piecewise polynomial functions.

Let B_i^w's be the control points for a B-spline curve described in $\mathbb{R}^{d+1}$ by the knot vector Ξ. The control points in $\mathbb{R}^d$ are obtained from the following expressions:

$$(B_i)_j = \frac{(B_i^w)_j}{w_i} \qquad w_i = (B_i^w)_{d+1}\ ,\ j = 1, \dots, d \tag{3.9}$$

where $(B_i)_j$ is the j-th component of the vector B_i, $(B_i^w)_j$ is the corresponding component of the projective vector of control points and $w_i \in \mathbb{R}$ is the weight associated to the i-th function.
Considering the set $\mathcal{I} = \{0, \dots, n\}$, the one-dimensional rational basis functions of degree p are given by

$$R_{k,p}(\xi) = \frac{w_k N_{k,p}(\xi)}{\sum_{i\in\mathcal{I}} w_i N_{i,p}(\xi)} \tag{3.10}$$

Two- (respectively three-) dimensional *NURBS* basis functions are defined as the bivariate (respectively trivariate) tensor product of the one-dimensional basis functions presented in (3.10). Here we introduce two sets $\mathcal{I} = \{0, \dots, n\}$ and $\mathcal{J} = \{0, \dots, m\}$, two

B-spline basis functions N and M and we define two knots vectors $\Xi_\xi = (\xi_0, \dots, \xi_a)^T$ and $\Xi_\eta = (\eta_0, \dots, \eta_b)^T$ where $a = n + p + 1$ and $b = m + q + 1$; thus the rational surfaces of degrees p and q can be expressed as

$$R_{kl,pq}(\xi, \eta) = \frac{w_{kl} N_{k,p}(\xi) M_{l,q}(\eta)}{\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} w_{ij} N_{i,p}(\xi) M_{j,q}(\eta)} \tag{3.11}$$

The extension to three dimensional basis functions is straightforward after defining a new knot vector, a new degree and a new B-spline basis function; for the sake of readibility we omit the degrees p and q from the formulations of *NURBS* surfaces and volumes.

Starting from the general form of the transformation (3.5), we provide some additional details for the two-dimensional case: let Ω_0 be the square $(0,1) \times (0,1)$, any point $\mathbf{y} = (x, y)^T$ in the physical domain Ω is mapped back to a point $\boldsymbol{\xi} = (\xi, \eta)^T$ in the parametric domain as described in figure 3.1. Thus we associate a control point to each basis function and we can explicitly describe the relationship in equation (3.5) as follow

$$\mathbf{y}(\xi, \eta) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} R_{ij}(\xi, \eta) \mathbf{Y}_{ij} \tag{3.12}$$

where $\mathbf{Y}_{ij} \in \mathbb{R}^2$ are the coordinates of the control point of indices (i, j) in the parametric domain. Using this representation, we can express the domain Ω as a single *NURBS* patch and the geometrical description is fully provided by the control points, the weights, the knot vectors and the degrees of the basis functions.

As previously done for B-spline functions, we outline some major properties and we recall [PT97] for further details.

1. *NURBS* basis functions constitute a partition of unity: $\forall \xi$, $\sum_{i=1}^{n} R_{i,p}(\xi) = 1$.
2. The continuity and the support of *NURBS* basis functions are the same as for B-splines.
3. *NURBS* are affine-covariant, that is affine transformations in the physical space are obtained by applying the transformation to the control points.
4. If the weights are all equal, then *NURBS* functions are B-spline functions.
5. *NURBS* surfaces and volumes are piecewise polynomial entities arising from the projective transformations of tensor products.

3.3 IsoGeometric Analysis

Classical numerical methods for the approximation of the elasticity problem firstly rely on the construction of a discrete domain Ω_h which is a polygonal approximation of the continuous object we are studying. Major drawbacks of this approach consist in the high computational cost required by the mesh generation process and in the numerical errors introduced by the geometrical approximation.
To avoid this problem, T.J.R. Hughes et al. ([HCB05]) propose a new approach to better integrate *Finite Element Analysis* and *Computer Aided Design* by means of a unique representation suitable for both the geometry and the discrete solution: *Non-Uniform Rational B-Splines* are a *de facto* standard for geometrical modelling in *CAD* thus main idea of *IsoGeometric Analysis* consists of employing this parametric representation both

for exactly describing the computational domain and for solving the governing equations without previously approximating the domain by means of a piecewise linear grid.

We recall the general framework for the treatment of differential problems using *IsoGeometric Analysis* and we refer to [CHB09] for a detailed introduction to *IGA*-based Finite Element Method:

1. The mesh for a *NURBS* patch is defined by the product of knot vectors, that is $\Xi_\xi \times \Xi_\eta$ in two dimensions.

2. Knot spans subdivide the domain into *elements* which correspond to *FEM* elements.

3. The support of each basis funtion is a patch containing a small number of elements.

4. The geometry is defined by the control points associated to the basis functions.

5. The description of the problem is *isoparametric* since the unknown fields are represented using the same parametrization of the geometry and the degrees of freedom are the coefficients of the basis functions.

6. Mesh refinement known as k-refinement is obtained from a combination of order elevation and knot insertion.

7. Global matrices for the resolution of the algebraic problem associated to the isogeometric formulation can be assembled starting from local matrices constructed from *NURBS* patches. Compatibility conditions among different patches are imposed by employing the same *NURBS* representations for *NURBS* edges and surfaces on both sides of the interfaces. Hence, the refinement propagates from patch to patch and a continuous Galerkin method for *IsoGeometric Analysis* arises ([CHR07]). The scenario involving non-conforming patches can be handled by either formulating a discontinuous Galerkin method or imposing additional constraint equations to attain pointwise compatibility at patch interfaces as described by P. Kagan et al. in [KFBY03].

8. Direct methods for local refinement are not available using B-splines and *Non-Uniform Rational B-Splines.* For a general overview of this topic we refer to [VGJS11]. Moreover, to account for local refinements, M. Scott et al. introduce the concept of *Analysis-suitable T-splines* and we refer to [DJS10] and [SLSH12] for further details.

9. Dirichlet boundary conditions are imposed by applying them to the control variables. In the case of homogeneous Dirichlet conditions, this procedure results in exact pointwise satisfaction. In the case of inhomogeneous Dirichlet conditions, the boundary values have to be approximated by functions lying in the *NURBS* space.

10. Neumann boundary conditions are naturally satisfied by performing the integration over the boundary of the domain in the same way as in standard Finite Element formulations.

Eventually, we present a summary of similarities and differences between Finite Element Method and *IsoGeometric Analysis* by means of tables 3.1 and 3.2.

Finite Element Method IsoGeometric Analysis
Partition of unity
Compact support
Affine covariance
Patch tests satisfied
Isoparametric concept

Table 3.1: List of the main similarities between the Finite Element Method paradigm and the *IsoGeometric Analysis* frameowrk based on *NURBS*.

Finite Element Method	IsoGeometric Analysis
Nodal points	Control points
Nodal variables	Control variables
Mesh	Knot vectors
Basis interpolates nodal points and variables	Basis does not interpolate control points and variables
Approximated geometry	Exact geometry
Polynomial basis	*NURBS* basis
Subdomains	Patches

Table 3.2: Comparison of the main differences between the Finite Element Method paradigm and the *IsoGeometric Analysis* frameowrk based on *NURBS*.

3.3.1 Galerkin formulation

The starting point to build an *IsoGeometric Analysis* Galerkin method is the variational formulation (3.3) of the problem whose strong form is defined by equation (3.2).
Let us introduce a family of two-dimensional *NURBS* basis functions $\widehat{R}_{ij}$'s defined in the physical domain such that

$$\widehat{R}_{ij}(\mathbf{y}) = \widehat{R}_{ij}(x, y) = \widehat{R}_{ij} \circ F(\xi, \eta) = R_{ij}(\boldsymbol{\xi}) \tag{3.13}$$

Thus the discretized unknown displacement field $\mathbf{u}_h(\mathbf{y})$ is constructed as convex combination of the *NURBS* functions that describe the geometry

$$\mathbf{u}_h(\mathbf{y}) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} R_{ij}(\boldsymbol{\xi}) \mathbf{U}_{ij} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \widehat{R}_{ij}(\mathbf{y}) \mathbf{U}_{ij} \tag{3.14}$$

The unknown values $\mathbf{U}_{ij}$'s are two-dimensional vectors and are comparable to control points. In general, they do not stand for diplacements in specific nodes since *NURBS* functions are not interpolatory in the knots; however in our problem we employed open knot vectors thus forcing *NURBS* functions to be interpolatory on the boundary. Thanks to this property, we can easily impose zero Dirichlet boundary conditions on Γ_D by setting to zero the coefficients $\mathbf{U}_{ij}$ that belong to the corresponding knots on the boundary Γ_D.

Thinking of boundary conditions as a constraint over the degrees of freedom of the problem, we can write the unknown field eliminating the basis functions that are required to enforce the zero Dirichlet boundary conditions; renumbering the remaining basis functions and unknowns with an index ℓ spanning 1 to L, equation (3.14) reads as

$$\mathbf{u}_h(\mathbf{y}) = \sum_{\ell=1}^{L} \mathbf{N}_\ell(\mathbf{y}) U_\ell = \mathbf{N}(\mathbf{y}) \mathbf{U} \tag{3.15}$$

where $\mathbf{U} \in \mathbb{R}^L$ contains the unknown displacement coefficients $\mathbf{U}_{ij}$'s corresponding to the unconstrained control points, that is, the ones where we do not impose Dirichlet boundary conditions. The assembled matrix $\mathbf{N}(\mathbf{y}) \in \mathbb{R}^{2\times L}$ is constructed from the *NURBS* basis functions R_{ij}'s in a way that each column $\mathbf{N}_\ell(\mathbf{y}) : \mathbb{R}^2 \to \mathbb{R}^2$ splits U_ℓ in its components along the xy-directions.
Thus the finite-dimensional subspace $V_h \subset V$ where we set the isogeometric paradigm for the numerical approximation of the elasticity problem is given by the span of the *NURBS* basis functions previously described:

$$V_h = \langle \mathbf{N}_1, \ldots, \mathbf{N}_L \rangle \tag{3.16}$$

Imposition of the boundary conditions

As observed in point 9 and 10 of section 3.3 on the properties of *IsoGeometric Analysis*, a critical aspect is represented by the correct imposition of the boundary conditions. From a modelling point of view, Dirichlet conditions correspond to fixing the displacement over the boundary: this is trivial when dealing with homogeneous conditions; otherwise the situation is more complex because we have to properly approximate the data by means of *NURBS* basis functions, thus introducing additional numerical errors. On the other hand, Neumann boundary conditions naturally arise from the formulation of the linear elasticity problem and have the physical meaning of imposing a stress over the boundary of the structure in analysis.

Moreover, special configurations of the geometry may lead to the imposition of symmetrical boundary conditions in order to decrease the number of degrees of freedom in the problem, thus reducing the overall computational effort. This is the case we analyze in this work when dealing with an application on computational mechanics; additional details on the analytical and numerical treatment of this problem will be provided in chapter 5.

Discrete algebraic problem

Using the space V_h defined in (3.16) we can build a stiffness matrix $K \in \mathbb{R}^{L\times L}$ and a force vector $\mathbf{F} \in \mathbb{R}^L$. The algebraic formulation of the elasticity problem is straightforward:

$$K\mathbf{U} = \mathbf{F} \tag{3.17}$$

The elements of the stiffness matrix and the force vector arising from the discretization of the variational formulation are described in equation (3.18); in particular, these entries are computed using classical quadrature rules and the integration is performed using Gaussian quadrature points in the parametric domain. An overview of more efficient quadrature techniques is available in [HRS10].

$$K_{ij} = \int_\Omega (2\mu\boldsymbol{\epsilon}(\mathbf{N}_i) : \boldsymbol{\epsilon}(\mathbf{N}_j) + \lambda div(\mathbf{N}_i) div(\mathbf{N}_j)) d\omega \qquad , \qquad F_\ell = \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{N}_\ell d\sigma \tag{3.18}$$

Since the support of the *NURBS* basis functions is larger than the one of classical Lagrangian Finite Element basis functions, the pattern of the *IGA* stiffness matrix is generally less sparse. For the solution of the linear system (3.17), several strategies are available. Since K is a symmetric positive definite matrix and the dimension of the problem is moderate, we choose a classical sparse direct solver such as the multi-frontal method implemented in the *UMFPACK Library.*

3.3.2 Convergence properties of IsoGeometric Analysis

First of all, we recall the definition of energy norm for the problem (3.3):

$$\|\mathbf{u}\|_E^2 := \int_\Omega (2\mu\boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{u}) + \lambda div(\mathbf{u}) div(\mathbf{u}))d\omega$$

Existence and uniqueness of a solution for the linear elasticity problem are guaranteed by the results stated in section 3.1.1. Thus the classical best approximation property holds for the discrete solution $\mathbf{u}_h$:

$$|\mathbf{u} - \mathbf{u}_h|_{H^1(\Omega)} \leq C \inf_{\mathbf{v}\in V_h} |\mathbf{u} - \mathbf{v}|_{H^1(\Omega)}$$

Moreover we observe that the semi-norm $|\mathbf{u}|_{H^1(\Omega)}$ is equivalent to the energy norm in $H^1(\Omega)$ and the previous statement also reads as

$$\|\mathbf{u} - \mathbf{u}_h\|_E \leq \|\mathbf{u} - \mathbf{v}\|_E \quad \forall \mathbf{v} \in V_h$$

The proof of the convergence of the solution computed by using *IsoGeometric Analysis* is very complex and here we only report the convergence estimate for our case, where we focus on the H^1-seminorm of the solution. For further details, including the extension to a generic H^k Sobolev space and the proof of these results we refer to the work of Y. Bazilevs et al. [BBdVC$^+$06].

We assume that the geometrical parametrization F which maps an element Q in the parametric space into an element K in the physical space is at least of class $\mathcal{C}^0$: let Q denote the cartesian product of d non-empty knot spans, thus we have $K = F(Q)$. Moreover if the degree of the *NURBS* basis used to describe the problem is p, then we can write the element-wise estimate

$$|\mathbf{u} - \mathbf{u}_h|_{H^1(K)} \leq Ch^p \sum_{i=0}^{p+1} \|\nabla F\|_{L^\infty(\widetilde{Q})}^{i-p-1} |\mathbf{u}|_{H^i(\widetilde{K})} \tag{3.19}$$

where C is a constant that depends on the geometry of Ω and on the size of Q but not on the meshsize parameter h; $\widetilde{Q}$ (respectively $\widetilde{K}$) denote the extended supports of elements Q (respectively K) grouping the supports of all the basis functions that are non-zero over Q (respectively K). Again we can infer the main difference of sparsity between Finite Element Method and *IsoGeometric Analysis* by observing the broader support of *NURBS* basis functions and the consequent less sparse pattern of the stiffness matrix.

By summing up (3.19) over all elements, we can establish a convergence estimate of order p with respect to the H^1-seminorm. We remark that this estimate requires bounds for the i-th order seminorm of the solution $\mathbf{u}$ unlike the Finite Elements case. Also, the parametrization enters the error bound and thus has a clear effect on the quality of the solution. If we measure the error in the H^k-seminorm where $k > 1$ and if the smoothness of the parametrization is less than $\mathcal{C}^{k-1}$, the global error estimate requires the introduction of the so-called *bent Sobolev spaces* to take into account the reduced smoothness of the parametrization along specific knot lines. For more details, we refer again to [BBdVC$^+$06].

Remark 3.7. From a computational point of view, we observe that $\mathbf{u}_h$ approaches the energy norm of the exact solution from below, that is $\|\mathbf{u}_h\|_E \leq \|\mathbf{u}\|_E$. Moreover, when refining a discretization it must hold $\|\mathbf{u}_{h_1}\|_E \leq \|\mathbf{u}_{h_2}\|_E$ where h_1 stands for the fine grid and h_2 for the coarse one. This property is often used to cross-check the numerical solution from a practical point of view.

3.4 Shape optimization problem in structural engineering

A classical problem of shape optimization in structural engineering concerns the optimization of the shape of a hole within a squared plate subject to external traction ([Kir81]). From a mathematical point of view, this corresponds to studying the configuration of the free boundary Γ while minimizing the compliance of the structure subject to a constant mass constraint.

The objective functional $J(\Omega)$ depends on the shape of the domain Ω and represents the compliance of the structure which by intuition is the *inverse* of the stiffness and describes its deformability. In the literature it is defined as

$$J(\Omega) = \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{u} d\sigma \tag{3.20}$$

thus minimizing the compliance corresponds to making the structure more rigid.

Let us introduce the functional $G(\Omega)$ as the integral form of the volume of the object

$$G(\Omega) = \int_{\Omega} d\omega \tag{3.21}$$

Therefore we can write the constrained shape optimization problem in the following form:

$$\min_{\Omega \in \mathcal{U}_{ad}} J(\Omega) \qquad , \qquad \mathcal{U}_{ad} = \left\{ \Omega \subset \mathbb{R}^d \ , \ G(\Omega) = V_0 \right\} \tag{3.22}$$

where $\mathcal{U}_{ad}$ is the set of admissible shapes for the domain Ω and V_0 is the initial volume which represents the reference value for $G(\Omega)$.

3.4.1 Existence and uniqueness of the optimal shape

We recall the definition in equation (1.5) of the space of admissible shapes arising as deformations of a domain Ω. All the shapes in $\mathcal{O}_{\mathcal{T}}(\Omega)$ have the same topology as Ω thus this approach does not allow the topological optimization of the domain, that is no change in the number of connected components of the boundary is possible. A general reference for the treatment of shape and topological optimization is [dGAJ08].

Let us introduce a pseudo-distance in $\mathcal{O}_{\mathcal{T}}(\Omega)$ as follows

$$d(\Omega_1, \Omega_2) = \inf_{\substack{T \in \mathcal{T} \\ T(\Omega_1) = \Omega_2}} \left[\|T - I\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} + \|T^{-1} - I\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} \right] \tag{3.23}$$

The following existence result may be stated and a complete proof is available in [MS76]:

Theorem 3.8. *Let us restrict the set of admissible shapes $\mathcal{U}_{ad}$ to small variations of the reference shape Ω_0 according to the distance (3.23). Thus the shape optimization problem (3.22) reads as*

$$\min_{\Omega \in \mathcal{U}^d_{ad}} J(\Omega) \qquad , \qquad \mathcal{U}^d_{ad} = \left\{ \Omega \in \mathcal{U}_{ad} \ \middle| \ d(\Omega, \Omega_0) \leq R \ , \ R > 0 \right\} \tag{3.24}$$

and has at least one minimum point, that is an optimal shape exists.

Remark 3.9. Uniqueness of the optimal shape may only be conjectured in a general case and additional assumptions and restrictions have to be made to prove it in specific cases. As a matter of fact, many topological and geometrical issues arise and we refer to [Cha03], [Che75] and [Pir84] for further details on these topics.

3.4.2 Penalty formulation

Classically, problems in constrained optimization have been treated by constructing a new functional with an additional penalty term such that the value of the functional worsens in the region where the constraint is not satisfied. Thus we can write the optimization problem in the form

$$\min_{\Omega \in \mathbb{R}^d} F(\Omega, \beta) \qquad , \qquad F(\Omega, \beta) = J(\Omega) + \beta\Big(G(\Omega) - V_0\Big)^2 \tag{3.25}$$

where β is a penalty parameter as big as the user wants (usually $\beta \simeq \mathcal{O}(10^6)$) that allows to enforce the value of the functional F to worsen where the constraint is not satisfied.

Since this work focuses on gradient-based optimization methods, we require that the functionals have sufficient regularity in order to ensure the existence of point-wise gradients within the region of admissible solutions. For this reason we construct the penalty term as a quadratic power to get at least $\mathcal{C}^1$ regularity for F. Under this assumption, for every point of the domain we are able to compute the first-order partial derivatives with respect to all the variables.

In the literature several works deal with a comparative study of penalty methods and other classical optimization approaches such as barrier or Lagrangian formulations. A major drawback of the approaches based on penalty parameters is the difficulty of correctly calibrating the value of β to ensure an effective penalization of the objective functional: if β is too small, then the algorithm will consider also inadmissible configurations; on the contrary, if β is too big this method will discard feasible solutions even in presence of minor variations of the volume which could be *a priori* admissible.

The critical role of the penalty parameter and the sensitivity of the optimization methods to its value increase when dealing with multiobjective optimization. As a matter of fact, the presence of the same penalty parameter in the formulation of different functionals $F_i(\Omega, \beta)$'s - for example in the case of functionals arising from different configurations of the boundary conditions - may generate conflicts during the calibration process if the span of the value of the F_i's is very broad for different realizations of i. Moreover, the mathematical problem could become even more complex if different penalty parameters associated to several constraints appear: this is the case of local strain constraints, where possible correlations among the constraints could lead to non-trivial dependencies of the parameters β_j's $\ j = 1, \ldots, M$ where M represents the total number of restrictions to be imposed.

In this scenario, identifying the real relationships among mathematical and physical quantities and constructing a robust algorithm is not affordable because of the builtin uncertainties which are not easy to estimate and control.

3.4.3 Lagrangian formulation

A finer approach to the treatment of the constraints is based on the formulation of an unconstrained optimization problem by means of the Lagrangian functional $L(\Omega, \lambda)$

$$\min_{\Omega \in \mathbb{R}^d} L(\Omega, \lambda) \qquad , \qquad L(\Omega, \lambda) = J(\Omega) + \lambda\Big(G(\Omega) - V_0\Big) \tag{3.26}$$

where $\lambda > 0$ is a positive Lagrange multiplier.

Main drawback of the formulation of the optimization problem using the Lagrangian $L(\Omega, \lambda)$ is due to the nature of the optimal solution (Ω^*, λ^*) which is a saddle point. As a matter of fact, Ω^* represents the geometrical configuration that minimizes $L(\Omega, \lambda)$ whereas the optimal Lagrange multiplier λ^* is a maximum for the Lagrangian functional.

Moreover if we extend the Lagrangian formulation to the case of multiobjective optimization, other issues arise because of the nature of the pair (Ω^*, λ^*). Let $\mathbf{g}_1$ and $\mathbf{g}_2$ be two different configurations of the Neumann boundary conditions, then we introduce the Lagrangian functionals $L_1(\Omega, \lambda)$ and $L_2(\Omega, \lambda)$ for the minimization of the compliances $J_1(\Omega)$ and $J_2(\Omega)$ subject to the volume constraint. Within the multiobjective optimization paradigm associated to *Multiple-Gradient Descent Algorithm*, this results in a two-steps procedure where we first minimize L_i , $i = 1, 2$ with respect to the state variable $\mathbf{Y} \in \Omega$

Listing 3.1: *MGDA* for Lagrangian multiobjective optimization - Step 1

```
1. Fix $\lambda$;
2. Compute the gradient vectors $\nabla_{\mathbf{Y}} L_i$ , $i = 1, 2$;
3. Identify $\boldsymbol{\omega}$ as a convex combination of $\nabla_{\mathbf{Y}} L_i$;
4. Determine an optimal step size $\widetilde{\rho}$;
5. Update the design point $\mathbf{Y}$ to $\mathbf{Y} - \widetilde{\rho}\boldsymbol{\omega}$;
6. Perform MGDA until convergence.
```

then we maximize them with respect to the adjoint variable λ

Listing 3.2: *MGDA* for Lagrangian multiobjective optimization - Step 2

```
1. Fix $\mathbf{Y} \in \Omega$;
2. Compute the gradient vectors $\nabla_\lambda L_1$;
3. Identify a descent direction $\boldsymbol{\omega}'$ for $\nabla_\lambda L_1$;
4. Determine an optimal step size $\overline{\rho}$;
5. Update the design point $\lambda$ to $\lambda + \overline{\rho}\boldsymbol{\omega}'$;
6. Perform Steepest Descent Method until convergence.
```

We remark that $\nabla_\lambda L_1(\Omega, \lambda) = \nabla_\lambda L_2(\Omega, \lambda)$ since these terms represent the constraint over the volume variation: as a matter of fact, we are seeking a shape Ω such that both the configurations of the boundary conditions are satisfied, thus the domain has to be the same when minimizing $L_1(\Omega, \lambda)$ and $L_2(\Omega, \lambda)$. However this formulation presents several drawbacks, first of all the necessity to perform an hybrid multi-single objective optimization strategy. Moreover it is well known in the literature that this formulation of the optimization algorithm is not stable and convergence issues may arise under some circumstances. A possible solution comes from the coupling of the processes to identify a descent direction at each time step, giving the so-called *Uzawa algorithm*:

$$\begin{cases} \mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} - \widetilde{\rho}^{(k)} \nabla_{\mathbf{Y}} L_1(\cdot, \lambda^{(k)}) \\ \mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} - \widetilde{\rho}^{(k)} \nabla_{\mathbf{Y}} L_2(\cdot, \lambda^{(k)}) \\ \lambda^{(k+1)} = \lambda^{(k)} + \overline{\rho}^{(k)} \nabla_\lambda L_1(\mathbf{Y}^{(k)}, \cdot) \end{cases} \tag{3.27}$$

Further details about the theoretical foundations of this approach can be found in [BPV97]. However, the complexity of the mathematical formulation of the problem and the computational cost arising from the optimization procedure based on a system of fully coupled PDEs make this approach not particularly appealing. Hence we choose not to deal with this topic in this work but a detailed analysis could be interesting for future investigations.

3.4.4 Shape gradient for the linear elasticity problem

The problem of shape optimization in structural engineering is widely treated in [All06] where G. Allaire proves some major results about the differentiation of the compliance with respect to the domain. Moreover the author presents a numerical algorithm to perform shape optimization starting from the concept of shape derivative. In particular, the author remarks that the problem is self-adjoint and this restricts the solution of the adjoint problem to the solution of the primal one.

Under the assumptions $\mathbf{g}, \mathbf{u} \in H^2(\Omega)$ we can differentiate equation (3.20) with respect to the domain Ω as described in section 1.4. We remark that the map $T_t : \Omega \mapsto \Omega_t$ in equation (1.7) tracks the deformation of the domain at time t and is different and *a priori* independent from the map $F : \Omega_0 \mapsto \Omega$ in equation (3.5) that describes the mechanical deformation of the system within the isogeometric paradigm.

Remark 3.10. In this context, a major advantage of the isogeometric paradigm relies on the use of a unique basis for the representation of the geometry of the domain and for the computational procedure: as a matter of fact, both the structural displacement field arising from the linear elasticity problem and the deformation field for the shape optimization algorithm may be written by means of *NURBS* basis functions leading to a global procedure that uses only one parametrization.

Here we only report the main result of the form of the shape derivative of the compliance for a given deformation $\mathbf{v}$:

$$dJ(\Omega; \mathbf{v}) = -\int_\Gamma \Big(2\mu|\boldsymbol{\epsilon}(\mathbf{u})|^2 + \lambda|div(\mathbf{u})|^2\Big)\mathbf{v} \cdot \mathbf{n} d\sigma \tag{3.28}$$

In next sections we will formulate different optimization procedures based on the information carried by (3.28).

3.5 Shape optimization procedures

First we present the algorithm proposed by L. Blanchard et al. in [BDVS12] for the construction of an optimization procedure starting from the concepts of descent direction and shape derivative. Then we describe the idea of setting the optimization problem (3.22) in a multiobjective optimization framework in order to apply *Multiple-Gradient Descent Algorithm* to compute the Pareto-optimal solutions as discussed in 2.

3.5.1 Shape optimization using shape-based Steepest Descent Method

As for the differentiation of the compliance functional in equation (3.28), we can compute the shape derivative for the Lagrangian functional (3.26) as follows:

$$dL(\Omega; \mathbf{v}) = \int_\Gamma \Big(\lambda - \big(2\mu|\boldsymbol{\epsilon}(\mathbf{u})|^2 + \lambda|div(\mathbf{u})|^2\big)\Big)\mathbf{v} \cdot \mathbf{n} d\sigma \tag{3.29}$$

Thus an extension of the Steepest Descent Method in terms of shape deformations is straightforward

$$\Omega_{k+1} = (I + \mathbf{v}_k)(\Omega_k)$$

where $\mathbf{v}_k$ describes a deformation such that

$$\mathbf{v}_k = \begin{cases} (2\mu|\boldsymbol{\epsilon}(\mathbf{u}_k)|^2 + \lambda|div(\mathbf{u}_k)|^2 - \lambda_k)\mathbf{n}_k & , \ \Gamma \\ 0 & , \ \Gamma_D \cup \Gamma_N \end{cases} \tag{3.30}$$

As L. Blanchard et al. point out in [BDVS12], equation (3.30) defines a shape deformation of the boundaries $\partial\Omega = \Gamma \cup \Gamma_D \cup \Gamma_N$ but to define a map that is a perturbation of the identity (Section 1.4.1) we are interested in a deformation field involving the whole domain. Classical approaches in the literature rely either on transfinite interpolation ([BF10]) or on regularization procedures that extend the function defined over the boundary inside the domain by solving an additional differential problem. Following the intuition of G. Allaire et al. in [AJT04], a displacement field $\mathbf{v}_k$ defined over the whole domain Ω_k is sought by means of an armonic extension formulated using a Laplacian operator. For the sake of simplicity, in our problem we choose to use the same isotropic operator previously introduced for the linear elasticity problem, thus we seek $\mathbf{v}_k$ such that

$$\begin{cases} -div\boldsymbol{\sigma}(\mathbf{v}_k) = \mathbf{0} & , \ \Omega \\ \mathbf{v}_k = \mathbf{0} & , \ \Gamma_D \cup \Gamma_N \\ \boldsymbol{\sigma}(\mathbf{v}_k) \cdot \mathbf{n}_k = (2\mu|\boldsymbol{\epsilon}(\mathbf{u}_k)|^2 + \lambda|div(\mathbf{u}_k)|^2 - \lambda_k)\mathbf{n}_k & , \ \Gamma \end{cases} \tag{3.31}$$

We notice that the deformation field $\mathbf{v}_k$ on the boundary Γ is not imposed as a Dirichlet condition but instead we consider a Neumann condition. It is well known in the literature that the algorithms based on shape derivatives may suffer from a loss of regularity thus the use of a Neumann boundary condition works as a smoothing operator that enables to increase the regularity of the deformation field $\mathbf{v}_k$. Moreover, from the numerical point of view, the field $\mathbf{v}_k$ is defined by using *NURBS* basis functions but the deformation in equation (3.30) usually does not belong to the *NURBS* space whereas the one arising from equation (3.31) does.

Eventually, we use the basis functions of the discrete space V_h in equation (3.16) to introduce a definition of the deformation field in terms of displacements of the control points:

$$\mathbf{v}_k(\mathbf{y}) = \sum_{\ell=1}^{L} \mathbf{N}_\ell(\mathbf{y}) V_\ell$$

where the components V_ℓ represent the motion of the control points. By recalling the form of the stiffness matrix in equation (3.18), a discrete formulation similar to the one in (3.17) is obtained also for the deformation field:

$$K\mathbf{V} = \mathbf{C}^1 - \lambda\mathbf{C}^2$$

$$C^1_\ell = \int_\Gamma (2\mu|\boldsymbol{\epsilon}(\mathbf{u})|^2 + \lambda|div(\mathbf{u})|^2)\mathbf{N}_\ell \cdot \mathbf{n} d\sigma \quad , \quad C^2_\ell = \int_{\Gamma_N} \mathbf{N}_\ell \cdot \mathbf{n} d\sigma \tag{3.32}$$

The vectors $\mathbf{C}^1$ and $\mathbf{C}^2$ include the information of the shape derivative of the compliance and the constant volume constraint. From a practical point of view, the system above is solved twice to compute $K^{-1}\mathbf{C}^1$ and $K^{-1}\mathbf{C}^2$; then for any Lagrange multiplier λ, the deformation can be easily computed without any extra cost:

$$\mathbf{V} = K^{-1}\mathbf{C}^1 - \lambda K^{-1}\mathbf{C}^2$$

In script 3.3 we report the workflow of the *Steepest Descent Method* for the shape optimization procedure based on the shape derivative.

Listing 3.3: Shape optimization procedure using *Steepst-Descent Method* with shape derivative

```
1.  Read data file:
 (a) Detect knot vectors and degree of the basis functions to define the
```

```
    isogeometric paradigm;
 (b) Detect the control points and the weights to define the exact geometry;
 (c) Set refinement options;
2.  Generate initial geometry;
3.  Run IGA solver to compute the structural displacement defined by
    the control points $\mathbf{U}^{(k)}$;
4.  Run IGA solver to compute the shape deformation defined by the control
    points $\mathbf{V}^{(k)}$;
5.  Initialize step length $t^{(k)}$
6.  Compute the Lagrange multiplier $\lambda^{(k)}$ such that the domain
    $\mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + t^{(k)}\mathbf{V}^{(k)}$ is admissible;
7.  Run IGA solver for the domain described by control points $\mathbf{Y}^{(k+1)}$ yielding
    to structural displacement defined by control points $\mathbf{U}^{(k+1)}$;
8.  Compute the compliance for the domain defined by $\mathbf{Y}^{(k+1)}$;
9.  If the compliance does not decrease, $t^{(k)}$ is reduced and go to step 6;
    otherwise the step length is retained;
10. If $\|J^{(k+1)} - J^{(k)}\| \geq tol$, go to step 3 and update $k = k+1$; otherwise the
    procedure has converged and the optimization loop may be arrested.
```

3.5.2 Shape optimization within a multiobjective optimization paradigm

An alternative method is based on the idea of formulating the optimization problem in (3.22) as a multiobjective optimization problem and applying *Multiple-Gradient Descent Algorithm* to identify the first Pareto front.

First of all, we have to identify two antagonistic criteria in order to describe a proper functional space where the competition for optimization takes place. From equation (3.28) we notice that $dJ(\Omega; \mathbf{v}) \leq 0$ if $\mathbf{v} \cdot \mathbf{n} > 0$, that is, the compliance can always be decreased by enlarging the domain.

In details, since (3.20) and (3.21) define two quantities whose improvement/worsening respectively depend on the worsening/improvement of the other, we can employ the functionals $J(\Omega)$ and $G(\Omega)$ as the criteria to identify the Pareto-optimal solutions of the problem. Thus the information carried by the shape derivative allow us to choose the compliance and the volume constraint as criteria for formulating the problem within the paradigm of multiobjective optimization as follows

$$\min_{\Omega \subset \mathbb{R}^d} \mathbb{J}(\Omega) \qquad , \qquad \mathbb{J}(\Omega) = (J(\Omega), G(\Omega) - V_0)^T \tag{3.33}$$

In the following chapter we will focus on the study of problem (3.33) by applying several variants of *Multiple-Gradient Descent Algorithm* as discussed in chapter 2. We aim at implementing a strategy for concurrent optimization, that is, finding a descent direction common to all criteria such that at every iteration of *MGDA* both the compliance and the volume constraint functional decrease. This procedure allows to identify a set of Pareto-optimal solutions starting from which competitive optimization algorithms can proceed to identify a subset of optimal solutions based on the nature of the problem.
Competitive optimization has not been treated in this thesis but we refer to appendix A for a general introduction to the topic since it represents a very promising field of investigation for MultiDisciplinary Optimization.

Remark 3.11. A more complex and general case of shape optimization problem treated within the framework of multiobjective optimization arises from aeronautical engineering: in this context, we want to minimize both the drag and the compliance of a given object - for example a wing - subject to a lift constraint. Basic idea is to employ a variation of the

classical procedure named *Projected Multiple-Gradient Descent Algorithm* as presented in section 2.5: the gradients of the drag functional $D(\Omega)$ and the compliance $J(\Omega)$ are projected on the subspace of $\mathbb{R}^d$ where the constraint due to the lift coefficient is satisfied; then we seek a descent direction using the modified gradients.
Some applications to more complex problems are discussed in [Dés11] where the two-step global procedure of cooperative and competitive optimization is applied to a shape optimization problem of compressible fluid dynamics. Again, the author highlights the interest of *Multiple-Gradient Descent Algorithm* both as standalone gradient-based optimization technique and as initial deterministic procedure to determine Pareto-optimal configurations to be used as starting points in a dynamic Nash game.

Chapter 4

Multiple-Gradient Descent Algorithm for isogeometric structural shape optimization

In this chapter we focus on merging the optimization procedures described in chapter 2 with the parametrization techniques presented in chapter 3 to construct an efficient framework for the resolution of optimization problems using exact geometry description.
Our main application is related to the linear elasticity problem in equation (3.2) where the optimization variable is the free boundary Γ. The geometry of the problem is treated by means of the *NURBS* parametrization previously introduced whereas the optimization algorithm aims at finding a descent direction that concurrently minimizes the compliance and the variation over the volume constraint.

In the following sections we present the different algorithms implemented for the study of the optimization procedures focusing on their strengths and weaknesses and highlighting feasible improvements. Table 4.1 summarizes the strategies investigated: on one hand, we compare the methods used to compute the objective functionals and their gradients; on the other hand, we compare the dimension of the space of the optimization variables in order to quantify the computational cost of the implemented variants of *MGDA*.

Algorithm	Computation of J	Computation of ∇J	Variables	Applies to
MGDA	Exact through *IGA* solver	Approximated by FD on fine surrogate mesh	Control points on the coarse grid (Low number)	Single design point
Kriging-assisted *MGDA*	Predicted using kriging model	Approximated by FD on fine surrogate mesh	Control points on the coarse grid (Low number)	Set of design points
MGDA using analytical gradients	Exact through IGA solver	Extracted from *NURBS* parametrization in *IGA* solver	Control points after k-refinement (High number)	Single design point

Table 4.1: Comparison of three main strategies for shape optimization within a multiobjective optimization framework using *IsoGeometric Analysis*.

4.1 MGDA using numerical approximated gradients

The standard procedure for *Multiple-Gradient Descent Algorithm* has been presented in script 2.1. Concerning the linear elasticity problem in equation (3.2), the boundary is parametrized by means of *Non-Uniform Rational B-Splines* thus the optimization variables are the control points describing the *NURBS* curve Γ.

In particular, the algorithm starts by evaluating the objective functionals $J(\Omega)$ and $G(\Omega) - V_0$ invoking the *IsoGeometric Analysis* solver once. Then the optimization loop starts: a surrogate conforming mesh is built using a very fine spatial step and the gradients of the criteria are computed by means of a second order centered Finite Difference scheme. We remark that to perform this operation the values of the functionals $J(\Omega)$ and $G(\Omega)$ have to be computed in correspondance of two adjacent spatial nodes, thus the *IGA* solver have to be invoked $2q$ times where q is the global number of the control points used to describe Γ.

Since we are working on a two-dimensional optimization problem, the values of the coefficients for the convex combination can be computed analitically as in (2.11) thus leading to the identification of the descent direction $-\boldsymbol{\omega}$ common to all criteria. Then a bisection-based line search procedure is performed to identify the optimal step size $\widetilde{\rho}$ that guarantees the minimization of both the compliance and the volume variation.

Eventually the design point is updated using the step size $\widetilde{\rho}$ along the direction $-\boldsymbol{\omega}$ and the iterative loop ends when the point under analysis is Pareto-stationary, that is when $\|\boldsymbol{\omega}\|$ is smaller than an *a priori* fixed tolerance.

Optimization algorithm

Here we present the overall strategy for shape optimization based on classical *Multiple-Gradient Descent Algorithm.* The routine listed in script 4.1 describes the evolution of a single design point towards a Pareto-stationary configuration; thus it has to be repeated for an adequate set of design points in order to obtain a good description of the Pareto front.

Listing 4.1: Shape optimization procedure using classical *MGDA*

```
1. Read data file:
 (a) Detect the knot vectors and the order of the basis functions to define
     the isogeometric paradigm;
 (b) Detect the control points and the weights to define the exact geometry;
 (c) Set the refinement options;
2. Generate the initial geometry;
3. Run the IGA solver to compute the objective functionals:
 (a) Solve the displacement field;
 (b) Compute the compliance and the volume;
 (c) Export the objective functionals;
4. Compute the gradients of the criteria:
 (a) Generate a surrogate FD mesh with stepsize $s_h = 10^{-6}$;
 (b) Generate a new geometry by vertically and horizontally perturbing each
     component of the design vector alone adding $\pm s_h$;
 (c) Run IGA solver as in step 3;
 (d) $\forall i = 1,\dots,2q$ compute $\frac{\partial}{\partial Y_i}J(\mathbf{Y}) = \frac{1}{2h}(J(\mathbf{Y}_i^+) - J(\mathbf{Y}_i^-))$ and $\frac{\partial}{\partial Y_i}G(\mathbf{Y}) = \frac{1}{2h}(G(\mathbf{Y}_i^+) - G(\mathbf{Y}_i^-))$
     where $\mathbf{Y}_i^+ = [Y_1,\dots,Y_{i-1},Y_i + s_h,Y_{i+1},\dots,Y_q]$ and $\mathbf{Y}_i^- = [Y_1,\dots,Y_{i-1},Y_i - s_h,Y_{i+1},\dots,Y_q]$;
5. Identify the coefficients $\alpha$ and $1-\alpha$ as in equation (2.11) and identify
   the minimal-norm element $\boldsymbol{\omega}$ as convex combinations of the gradient
   vectors $\nabla J(\mathbf{Y})$ and $\nabla G(\mathbf{Y})$;
6. Perform line search by means of a bisection algorithm to determine $\widetilde{\rho}$ as
```

```
   described in section 2.2.2;
7. Update the design vector $\mathbf{Y}$ to $\mathbf{Y} - \widetilde{\rho}\boldsymbol{\omega}$;
8. If $\|\boldsymbol{\omega}\| \geq tol$, go to step 2; otherwise a Pareto-stationary design point has
   been achieved and the optimization procedure is stopped.
```

Remark 4.1. A satisfactory description of the Pareto front requires 20 to 40 design points, each of which is obtained after 30 to 50 iterations of *MGDA* procedure. Since for every run of the optimization algorithm, the solver is invoked $2q + 1$ times to perform the evaluation of the functionals and compute the components of the gradients, the resulting computational cost is very demanding. For this reason in the following section classical *Multiple-Gradient Descent Algorithm* is enhanced by the use of a kriging-based metamodel that strongly reduces the computational cost of the overall procedure by substituting the exact computation of the functionals with their prediction using statistical techniques for spatial data.

4.2 Kriging-assisted MGDA optimization

Metamodel-assisted *MGDA* procedure relies on the idea of skipping the exact computation of the functionals and their gradients by means of the numerical solver related to the problem and estimating these quantities using a statistical approach.

An initial data set of design points is generated using *Latin Hypercube Sampling* to uniformly map the whole domain of computation. A surrogate model based on this database is trained and is used to predict the values of the objective functionals in correpondance of new points as described in section 2.4.1. Initial design points are also used as starting points for the application of classical *Multiple-Gradient Descent Algorithm* until a Pareto-stationary point is achieved; in this case, the objective functionals are not computed by using the *IGA* solver but they are predicted by means of the surrogate model we just trained.

From previous theoretical results we know that each *MGDA* final point belongs to the Pareto front arising from the concurrent minimization of the two criteria in analysis. A set of non-dominated design points is obtained from the coupling of *MGDA* strategy with the surrogate kriging model; then each configuration is re-evaluated by means of the *IGA* solver and is added to the data set coming from the previous iteration.

We remark that a filter has to be applied in order to avoid the excessive increase of design points too close to one another. In particular, if the new entries of the database are too close to already existent elements, these new elements are neglected since the additional information they provide is quite poor with respect to what is already known.

After the enrichment process is over, the metamodel is updated and a more accurate surrogate model is obtained. This allows to increase the precision of the information and consequently to better predict the following entries of the data set, that is the objective functionals corresponding to the new design points.

Remark 4.2. In [ZDD12] the authors highlight the importance of an adimensionalization process in order to rescale the objective functionals and reduce the antagonism due to the different nature of the quantities in analysis. Same idea has to be applied to the design points in order to map them back to the interval $[-1, 1]$; under this assumption the *LHS* procedure initially performed to map the domain with uniformly distributed random points finds an *a posteriori* justification.

Optimization algorithm

As previously pointed out, metamodel-assisted optimization strategies are based on the joint use of deterministic and statistical approaches. A simple visualization of the workflow of the kriging-based *Multiple-Gradient Descent Algorithm* is available in figure 2.4 and the overall shape optimization procedure is detailed in scripts 4.2 and 4.3.
Again, the computation of the gradients of the functionals is performed by means of second order centered Finite Difference schemes on a surrogate conforming mesh. However the values of the objective functionals in correspondance of adjacent spatial nodes are not computed invoking the *IGA* solver but instead are predicted using the kriging metamodel. Thus the *IGA* solver is invoked only when a Pareto-stationary design point is achieved whereas the intermediate steps are handled by performing a Maximum Likelihood prediction.

A major role is played by the definition of the initial data set (Script 4.2) that will be later used both for the prediction of the values of the functionals by means of the kriging model and as starting point for the computation of *MGDA* iterates.

Listing 4.2: Shape optimization procedure using Kriging-assisted *MGDA* - Construction of the initial data set

```
1. Build an initial data set using Latin Hypercube Sampling in $\mathbb{R}^{2q}$ where $2q$
   is the number of components in the design vector;
2. For each entry of the data set build a data file for the linear
   elasticity problem;
3. Read the data file:
 (a) Detect the knot vectors and the order of the basis functions to define
     the isogeometric paradigm;
 (b) Detect the control points and the weights to define the exact geometry;
 (c) Set the refinement options;
4. Generate the initial geometry;
5. Run the IGA solver to compute the objective functionals:
 (a) Solve the displacement field;
 (b) Compute the compliance and the volume;
 (c) Export the objective functionals;
6. Store the values of the functionals in the data set.
```

Remark 4.3. A critical step in the construction of the initial data set is represented by the compatibility that has to be enforced between the physical problem and its discrete numerical representation since initial control points are generated using statistical techniques in order to uniformly map the whole domain. In particular, repeated control points or control points very close to one another may generate singularities in the control polygon: for this reason, kinks and singularities may appear in the representation of curves and surfaces using the exact geometry paradigm. Hence inaccurate initial control points can lead to the definition of singular geometries, thus causing non-physical scenarios in the analysis of stresses and displacements in a linear elasticity problem to appear.

In script 4.3 we focus on the optimization procedure performed starting from the previously defined data set. We remark that for this process the number of initial entries of the data set has to be fixed *a prori* and for each element the routines of the *IGA* solver have to be invoked. Moreover the metamodel-assisted optimization strategy is performed an *a priori* fixed number of times that have to be properly calibrated in order to obtain a robust method.

Listing 4.3: Shape optimization procedure using Kriging-assisted *MGDA* - Metamodel-assisted optimization

```
1.  Acquire the data set;
2.  Perform the training of the data set:
 (a) Compute the log-likelihood associated with the model;
 (b) Calibrate the parameters that describe the correlation among the data
     in order to minimize the log-likelihood;
3.  Perform kriging-based prediction of the functional values
 (a) Compute the correlation among the data in the data set;
 (b) Compute the prediction of the values;
 (c) Compute the variance of the predicted values;
4.  Compute the gradients of the criteria:
 (a) Generate a surrogate FD mesh with stepsize $s_h = 10^{-6}$;
 (b) Generate a new geometry by vertically and horizontally perturbing each
     component of the design vector alone adding $\pm s_h$;
 (c) Perform kriging-based prediction as in step 3;
 (d) $\forall i = 1, \dots, 2q$ compute $\frac{\partial}{\partial Y_i} J(\mathbf{Y}) = \frac{1}{2h}(J(\mathbf{Y}_i^+) - J(\mathbf{Y}_i^-))$ and $\frac{\partial}{\partial Y_i} G(\mathbf{Y}) = \frac{1}{2h}(G(\mathbf{Y}_i^+) - G(\mathbf{Y}_i^-))$
     where $\mathbf{Y}_i^+ = [Y_1, \dots, Y_{i-1}, Y_i + s_h, Y_{i+1}, \dots, Y_q]$ and $\mathbf{Y}_i^- = [Y_1, \dots, Y_{i-1}, Y_i - s_h, Y_{i+1}, \dots, Y_q]$;
5.  Identify the coefficients $\alpha$ and $1 - \alpha$ as in equation (2.11) and identify
    the minimal-norm element $\boldsymbol{\omega}$ as convex combinations of the gradient
    vectors $\nabla J(\mathbf{Y})$ and $\nabla G(\mathbf{Y})$;
6.  Perform line search by means of a bisection algorithm to determine $\widetilde{\rho}$ as
    described in section 2.2.2;
7.  Update the design vector $\mathbf{Y}$ to $\mathbf{Y} - \widetilde{\rho}\boldsymbol{\omega}$;
8.  If $\|\boldsymbol{\omega}\| \geq tol$, go to step 3; otherwise a Pareto-stationary design point has
    been achieved and the optimization procedure is stopped.
9. Run the IGA solver to compute the objective functionals:
 (a) Solve the displacement field;
 (b) Compute the compliance and the volume;
 (c) Export the objective functionals;
10. Enrich the data set with the computed final MGDA points and the
    functional values corresponding to the new configurations;
11. Filter the data set to eliminate the design configurations similar to
    entries already present in the database in order to keep only additional
    information;
12. If the number of iterates of metamodel-assisted MGDA is not over, go to
    step 2; otherwise a set of Pareto-stationary solutions have been found
    and the overall procedure is stopped.
```

From a computational point of view, main advantage of metamodel-assisted *MGDA* is the reduction of the computational effort. As a matter of fact, the method applies to a set of design points at the same time but invokes the *IGA* solver only to evaluate last *MGDA* iterate. This results in the parallel evolution of several design points, thus leading to a complete description of the Pareto front in 10 to 15 runs of the global optimization strategy. Moreover, the iterative enrichment of the data set and its filtering are responsible for the improvement of the information at every iteration. Hence this procedure may be considered a sort of *predictor-corrector* optimization strategy.

Remark 4.4. The application of a statistical-based prediction strategy introduces additional error in the approximation of the problem. However the evaluation of the variance associated with the model allow to monitor the uncertainties due to the kriging strategy and to control them.

Remark 4.5. The strategy listed in 4.3 can be extended in order to consider a more general surrogate model that takes account for both the objective functionals themselves and a set of adjoint variables, namely their derivatives. The overall procedure remains the same and just steps 2 and 3 are rivisited: the information contained by the gradients is used to force the model to better fit the data. Once the *MGDA* procedure achieves a Pareto-stationary point, the solver is invoked (step 9) to evaluate the new configuration and again the objective functionals and their gradients are exported and used to enrich the original data set. In our application this approach seems very promising since we can directly extract the information related to the gradients from the numerical solver thanks to the isogeometric parametrization previously introduced. However this topic has not been treated in this work but the extension is straightforward and we expect good results from this improvement of the overall methodology.

4.3 Obtaining analytical gradients in the parametric space

The aim of this section is to extract the information on the gradients ∇G and ∇J in the parametric space from the *IGA* solver. In particular, we know that the isogeometric parametrization from section 3.2.3 allows to conveniently express the domain boundaries in terms of control points and *NURBS* basis functions. This approach has been widely treated in the literature and we refer to these works [WFC08], [HTM11] and [CH09] for further details.

We recall the form of the transformation F (3.5) that maps the parametric domain Ω_0 into the physical one Ω:

$$F(\boldsymbol{\xi}) = \mathbf{y}(\boldsymbol{\xi}) = \mathbf{y}(\xi, \eta) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} R_{ij}(\xi, \eta) \mathbf{Y}_{ij}$$

where the components of the control points in the parametric and physical domains are respectively $\mathbf{Y}_{ij} = (X_{ij}, Y_{ij})$ and $\mathbf{y}_{ij} = (x_{ij}, y_{ij})$.
Thus the Jacobian matrix arising from the transformation F is given by

$$\mathrm{J} = \frac{\partial}{\partial \boldsymbol{\xi}} F(\boldsymbol{\xi}) = \frac{\partial}{\partial \boldsymbol{\xi}} \mathbf{y}(\boldsymbol{\xi}) = \left[\begin{array}{cc} x_{,\xi} & y_{,\xi} \\ x_{,\eta} & y_{,\eta} \end{array} \right] \tag{4.1}$$

and thanks to equation (3.12) the Jacobian reads as

$$\mathrm{J} = \left[\begin{array}{cc} \sum_{i,j} R_{ij,\xi} X_{ij} & \sum_{i,j} R_{ij,\xi} Y_{ij} \\ \sum_{i,j} R_{ij,\eta} X_{ij} & \sum_{i,j} R_{ij,\eta} Y_{ij} \end{array} \right] \tag{4.2}$$

where $i \in \mathcal{I},\ \ j \in \mathcal{J}$ and the *NURBS* basis functions are $R_{ij} = R_{ij}(\xi, \eta)$.
In the following subsections we derive the gradients of the functionals $G(\Omega)$ and $J(\Omega)$ with respect to the control points $\mathbf{Y}_{kl}$ used for the geometrical parametrization of the domain by means of *Non-Uniform Rational B-Splines*. For a general reference about differentiation with respect to knots and control points of *NURBS*, we refer to [dB72] and [PT98].

4.3.1 Gradient of the volume in the parametric space

We notice that the volume functional $G(\Omega)$ only depends on the geometrical information of the domain. Thus it is straightforward to differentiate with respect to the control variables and the computation can be executed starting from the integral formulation in equation (3.21). In particular, we employ the transformation F and its Jacobian (4.2) to map Ω back to Ω_0 and perform the integration over the parametric domain by means of Gaussian quadrature points. The volume functional can be expressed as follows:

$$
\begin{aligned}
G(\Omega) &= \int_\Omega d\omega = \iint_{\xi,\eta} |\mathrm{J}| d\xi d\eta = \\
&= \iint_{\xi,\eta} \Big(\sum_{i,j} R_{ij,\xi} X_{ij} * \sum_{i,j} R_{ij,\eta} Y_{ij} - \sum_{i,j} R_{ij,\xi} Y_{ij} * \sum_{i,j} R_{ij,\eta} X_{ij} \Big) d\xi d\eta
\end{aligned}
\tag{4.3}
$$

Hence, to compute the partial derivatives of (4.3) with respect to the control points $\mathbf{Y}_{kl}$'s, we use the linearity of the operator $\mathbf{Y}_{kl} \mapsto G(\Omega)$ that allows us to write

$$
\frac{\partial}{\partial X_{kl}} G = \iint_{\xi,\eta} \Big(\sum_{i,j} R_{ij,\xi} \delta_{ik} \delta_{jl} * \sum_{i,j} R_{ij,\eta} Y_{ij} - \sum_{i,j} R_{ij,\xi} Y_{ij} * \sum_{i,j} R_{ij,\eta} \delta_{ik} \delta_{jl} \Big) d\xi d\eta \tag{4.4}
$$

where δ_{ik} is the classical Kronecher delta, that is

$$
\delta_{ik} = \begin{cases} 1 \ , \ i = k \\ 0 \ , \ \text{otherwise} \end{cases}
$$

Eventually the gradient ∇G with respect to control points $\mathbf{Y}_{kl} = (X_{kl}, Y_{kl})$ reads as

$$
(\nabla G)_{kl} = \left[\frac{\partial}{\partial X_{kl}} G \ , \ \frac{\partial}{\partial Y_{kl}} G \right]^T
$$

$$
\frac{\partial}{\partial X_{kl}} G = \iint_{\xi,\eta} \Big(R_{kl,\xi} * \sum_{i,j} R_{ij,\eta} Y_{ij} - \sum_{i,j} R_{ij,\xi} Y_{ij} * R_{kl,\eta} \Big) d\xi d\eta \tag{4.5}
$$

$$
\frac{\partial}{\partial Y_{kl}} G = \iint_{\xi,\eta} \Big(\sum_{i,j} R_{ij,\xi} X_{ij} * R_{kl,\eta} - R_{kl,\xi} * \sum_{i,j} R_{ij,\eta} X_{ij} \Big) d\xi d\eta \tag{4.6}
$$

Remark 4.6. We notice that the volume functional $G(\Omega)$ only depends on the geometry of the domain. From a computational point of view, this results in a routine that can be extracted from the *IGA* solver and performed independently from the calculation of the displacement field.

4.3.2 Gradient of the compliance in the parametric space

The computation of the gradient of the compliance functional with respect to the control points $\mathbf{Y}_{kl}$ relies on the ability to express the shape derivative $dJ(\Omega; \mathbf{v})$ as a duality product between the operator $\nabla J(\Omega)$ and the vector $\mathbf{v}$, that is $dJ(\Omega; \mathbf{v}) = \langle \nabla J, \mathbf{v} \rangle$.

The formulation of the shape derivative of the compliance is known from equation (3.28) and can be expressed in the following general form

$$
dJ(\Omega; \mathbf{v}) = \int_\Gamma B \mathbf{v} \cdot \mathbf{n} d\gamma \qquad , \qquad B = -\Big(2\mu |\boldsymbol{\epsilon}(\mathbf{u})|^2 + \lambda |div(\mathbf{u})|^2 \Big) \tag{4.7}
$$

Starting from the works of J.-P. Zolésio ([Zol82]) and J. Cea ([Cea86]), a procedure to compute the gradient $\nabla J(\mathbf{Y})$ from the above expression of the shape derivatives $dJ(\Omega; \mathbf{v})$ is implemented.

Thanks to the parametrization introduced in section 3.2.3, the domain Ω can be expressed as a *NURBS* surface whose control points are of the following form

$$
\mathbf{y} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} R_{ij}(\xi, \eta) \mathbf{Y}_{ij}
$$

The admissible deformation $\mathbf{v}$ applied to Ω is a *NURBS* surface itself thus we can write it as

$$\mathbf{v} = \sum_{k\in\mathcal{I}}\sum_{l\in\mathcal{J}} R_{kl}(\xi,\eta)\mathbf{V}_{kl} = \sum_{k\in\mathcal{I}}\sum_{l\in\mathcal{J}} R_{kl}(\xi,\eta)\left(V_{kl}^{(1)}\ ,\ V_{kl}^{(2)}\right)^T$$

The vector $\mathbf{n}$ that appears in equation (4.7) represents the normal direction to the *NURBS* surface $\mathbf{y}$ and is identified by the expression $\mathbf{n} = \mathbf{y}_{,\xi}\times\mathbf{y}_{,\eta}$ which in the two-dimensional case reads as

$$\mathbf{n} = \frac{1}{|\mathrm{J}|}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} R_{ij,\xi}(\xi,\eta)\left(Y_{ij}\ ,\ -X_{ij}\right)^T$$

where $|\mathrm{J}|$ is the determinant of the Jacobian of the transformation F and assumes the same formulation as in equation (4.3).

Moreover we notice that the boundary over which the integration is performed can be expressed as a *NURBS* curve, thus Γ is a set of points whose characterization is again the one previously presented. Hence the term $\mathbf{v}\cdot\mathbf{n}$ can be expressed as follows

$$\mathbf{v}\cdot\mathbf{n} = \frac{1}{|\mathrm{J}|}\sum_{k,l}\sum_{i,j}\left(R_{kl}V_{kl}^{(1)}R_{ij,\xi}Y_{ij} - R_{kl}V_{kl}^{(2)}R_{ij,\xi}X_{ij}\right) \tag{4.8}$$

where the indices i and k belong to the set $\mathcal{I}$ whereas j and l belong to $\mathcal{J}$. For the sake of readibility, the dependency of the basis functions R_{kl} and their derivatives $R_{ij,\xi}$ from the parametric variables (ξ,η) is omitted.

From equation (4.7) we know that

$$dJ(\Omega;\mathbf{v}) = \int_\Gamma B\mathbf{v}\cdot\mathbf{n}d\gamma = \iint_{\xi,\eta} B\mathbf{v}\cdot\mathbf{n}|\mathrm{J}|d\xi d\eta$$

and thanks to the expression (4.8), the following result is derived:

$$dJ(\Omega;\mathbf{v}) = \sum_{k,l}\sum_{i,j} a_{ij}^{kl}\left(Y_{ij}\ ,-X_{ij}\ \right)\left(V_{kl}^{(1)}\ ,\ V_{kl}^{(2)}\right)^T \quad , \quad a_{ij}^{kl} = \iint_{\xi,\eta} BR_{kl}R_{ij,\xi}d\xi d\eta \tag{4.9}$$

In equation (4.9) it is straightforward to isolate the term which represents the gradient of the compliance with respect to the control points, that is

$$dJ(\Omega;\mathbf{v}) = \langle\nabla J\ ,\ \mathbf{V}\rangle$$

Hence the components of the gradients of the functional $J(\Omega)$ read as

$$(\nabla J)_{kl} = \left[\frac{\partial}{\partial X_{kl}}J\ ,\ \frac{\partial}{\partial Y_{kl}}J\right]^T \tag{4.10}$$

where

$$\frac{\partial}{\partial X_{kl}}J = \sum_{i,j} a_{ij}^{kl}Y_{ij} \quad , \quad \frac{\partial}{\partial Y_{kl}}J = -\sum_{i,j} a_{ij}^{kl}X_{ij} \tag{4.11}$$

4.4 MGDA using analytical gradients

Thanks to the boundary parametrization and to the expressions of the gradients introduced in the previous section, the algorithm for the concurrent optimization of several criteria can be modified. In particular, we propose to use the information contained in the analytical gradients extracted from the *IGA* solver in order to avoid their numerical approximation.

Thus classical *MGDA* procedure is performed by replacing the numerical computation of the gradients within the optimization process with the extraction of the same information from the *IGA* solver: in this way, the computation of the gradients by means of Finite Difference routines can be avoided and the partial derivatives of the objective functionals with respect to the control points are given by equations (4.5), (4.6) and (4.11).

For this purpose, *Multiple-Gradient Descent Algorithm* routines are extended in order to be able to process the information about user-supplied gradient vectors. Moreover, the global shape optimization procedure is modified to properly deal with spaces of design points of different dimensions. This is due to some theoretical and technical issues concerning the projection of the design variables of the enriched space back to the original set of control points.

Remark 4.7. The k-refinement procedure to enrich *NURBS* space as described in section 3.2.2 modifies the location of the control points in order to mantain the geometrical and parametrical representation of the surface after oder elevation and knot insertion. If a linear relationship between the original control points vector $\mathcal{B}^{in}$ and the enriched one $\mathcal{B}^*$ holds, a projection onto the subspace of original control points would be straightforward:

$$\nabla J_i(\Omega)\Big|_{\mathcal{B}^{in}} = \mathcal{A}^T \nabla J_i(\Omega)\Big|_{\mathcal{B}^*} \quad , \quad \mathcal{B}^* = \mathcal{A}\mathcal{B}^{in} \tag{4.12}$$

We remark that a linear dependency of this type usually stands between the original knot vector and the enriched one whereas *a priori* this is neither sure nor trivial to prove for control points.

Thus the problem of correctly retrieving the gradient of the functionals with respect to the control points is solved by applying the optimization algorithm on the control points in the enriched space rather than in the original one. Main drawback of this strategy is the resulting higher dimension of the optimization problem that causes the computational cost to increase. Nevertheless this approach avoid the definition of the projection operator in equation (4.12) and leads to good results that can actually compete with the one given by the variants of *MGDA* presented so far.

Optimization algorithm

The overall strategy for shape optimization based on the enhanced variant of *Multiple-Gradient Descent Algorithm* is presented. During the execution of the isogeometric solver an enrichement of the initial grid is performed by means of k-refinement. For this reason, the numerical simulations are executed using a different set of control variables than the initial one. As previously pointed out, the definition of a projection operator that maps the control points in the enriched space back to the initial ones is not trivial: to overcome this difficulty, the refined geometry as well as the analytical gradients is exported and the optimization routines are executed in a higher dimensional space.

Thus the shape optimization procedure has to properly manage the information related to the description of the geometry (Step 2) in order to correctly update the computational domain by using the deformation driven by Pareto-stationary design points.

Listing 4.4: Shape optimization procedure using *MGDA* with user-supplied gradients

```
1. Read data file:
 (a) Detect the knot vectors and the order of the basis functions to define
     the isogeometric paradigm;
 (b) Detect the control points and the weights to define the exact geometry;
 (c) Set the refinement options;
2. Generate the initial geometry:
 (a) Construct the coarse grid;
 (b) If the routine is running within the optimization loop, import the
     updated position of the control points; otherwise perform k-refinement;
3. Export the information about the geometry:
 (a) Export the control points in the coarse space;
 (b) Export the control points in the refined space;
4. Run the IGA solver to compute the objective functionals:
 (a) Solve the displacement field;
 (b) Compute the compliance and the volume;
 (c) Export the objective functionals;
 (d) Extract the gradients from NURBS parametrization and export them;
5. Identify the coefficients $\alpha$ and $1-\alpha$ as in equation (2.11) and identify
   the minimal-norm element $\boldsymbol{\omega}$ as convex combinations of the gradient
   vectors $\nabla J(\mathbf{Y})$ and $\nabla G(\mathbf{Y})$;
6. Perform line search by means of a bisection algorithm to determine $\widetilde{\rho}$ as
   described in section 2.2.2;
7. Update the design vector $\mathbf{Y}$ to $\mathbf{Y}-\widetilde{\rho}\boldsymbol{\omega}$;
8. If $\|\boldsymbol{\omega}\| \geq tol$, go to step 2; otherwise a Pareto-stationary design point has
   been achieved and the optimization procedure is stopped.
```

Remark 4.8. As the algorithm presented in section 4.1, this variant of *MGDA* applies to single design points. Thus, a complete description of the first Pareto front requires several runs of the optimization algorithm. However, in this case the *IGA* solver is invoked only once per run because the objective functionals have to be evaluated only in the current configuration whereas the extraction of the components of the gradients is straightforward in the *NURBS*-based code.

Chapter 5

Numerical simulations

In this chapter we study a problem of computational mechanics using the tools of shape optimization and multiobjective optimization described so far. In particular, we focus on a two-dimensional problem well known in the literature and we use it to validate our optimization strategies.

After presenting the analytical solution of the problem, we solve it by means of a single objective *Steepest Descent Method* based on shape derivatives. The obtained final configuration is used to check the results given by the different variants of *MGDA* tested on this problem: as a matter of fact, the shapes associated with final *MGDA* iterates should represent perturbations of the analytical optimal shape.
Eventually, to cross-validate the ability of *Multiple-Gradient Descent Algorithm* to describe the first Pareto front, we use the so-called *Pareto Archived Evolutionary Strategy* to simulate the same problem and we compare the resulting Pareto sets.

5.1 Setup of the problem

We consider a two-dimensional squared flat plate with a hole located at its center, subject to external normal forces (Fig. 5.2b). The problem can be decomposed in a simpler one by using a symmetry property and considering only a quarter of the geometry (Fig. 5.2a). Let us assume that the traction applied on the boundary of the domain is constant. Thus the goal is to determine the shape of the boundary that minimizes the compliance for a constant plate area. For this case the solution is known analitically and is represented by

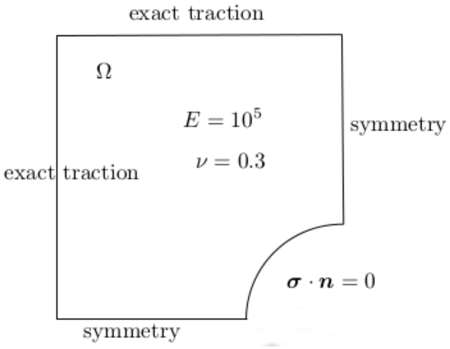
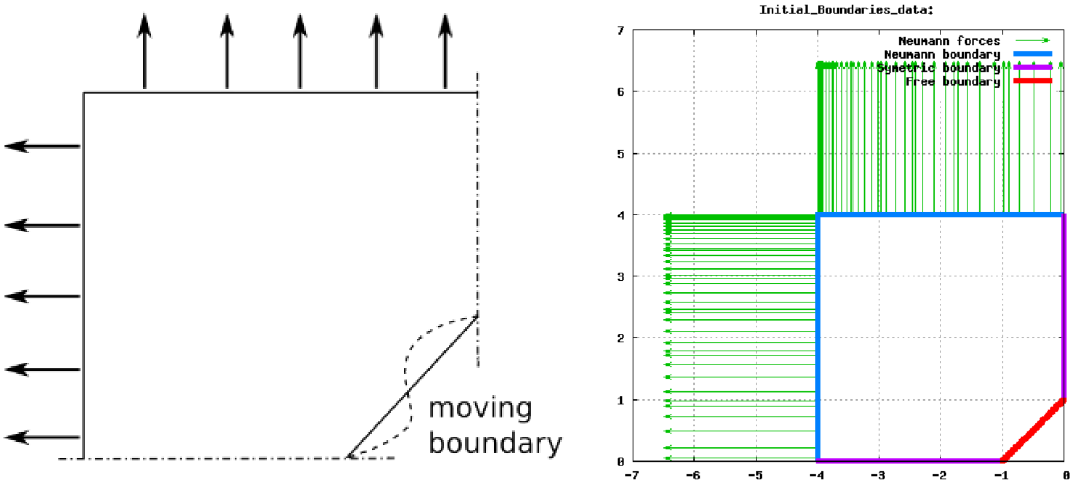


Figure 5.1: Analytical optimal shape of a two-dimensional flat squared plate with a central hole subject to uniform traction.



(a) Configuration of the problem.

(b) Imposed stress on the boundary.

Figure 5.2: Configuration of the free boundary problem and initial configuration of external stresses imposed as boundary conditions.

an arc of circumference as illustrated in figure 5.1.

In the following sections we provide several numerical approximation of the shape optimization problem arising from structural engineering and we analyze the optimal configurations generated by the variants of the algorithms described so far.

5.2 Steepest Descent Method using shape derivatives

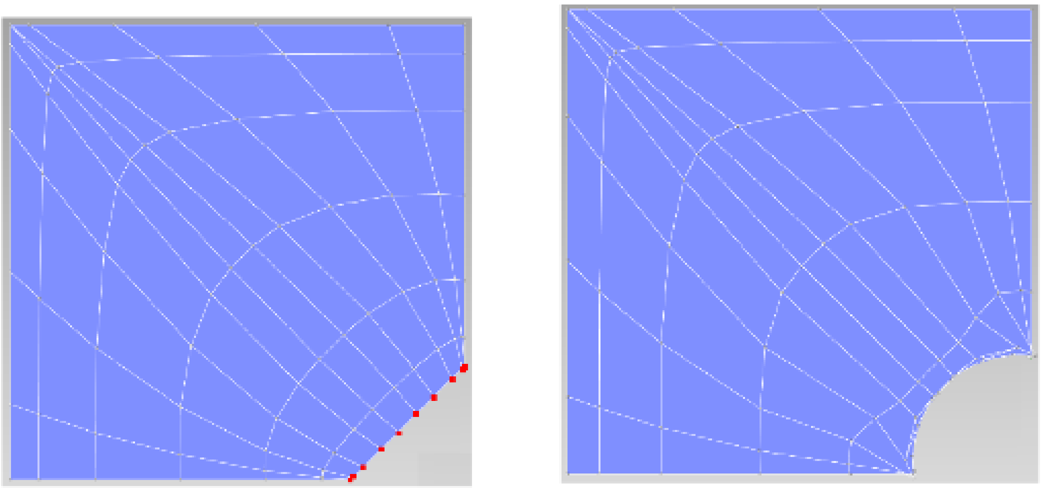
We define the computational domain by means of a single bi-quadratic patch which exhibits a singular point at the top-left corner. For the *NURBS* representation we choose quadratic basis functions, thus the singularity is obtained by introducing a control point of multiplicity two. Additional geometrical constraints are introduced to force the extremities of the moving boundary to stay on the symmetry axes.

In figure 5.3a we can observe the distribution pattern for the control points noticing that they do not look like a common Finite Element mesh nor a linear polygonal grid. As a matter of fact this depends on the use of *Non-Uniform Rational B-Splines* and the transfinite map generated by this basis functions.

The linear elasticity problem is approximated by means of a 12×7 net of control points; the optimization problem counts 24 variables that correspond to the coordinates of the 12 control points defining the moving boundary.

In details, in figure 5.3a the highlighted points on the free boundary are the control points that define the *NURBS* parametrization of Γ and will act as design variables for the shape optimization problem. To prevent the optimization algorithm to investigate non-physical solution some precautions are necessary: the first and last point on Γ can only slide respectively horizontally or vertically and a slight penalization is imposed to keep design points inside the domain, that is, when they excessively approach the symmetry lines the values of the functionals tend to worsen.

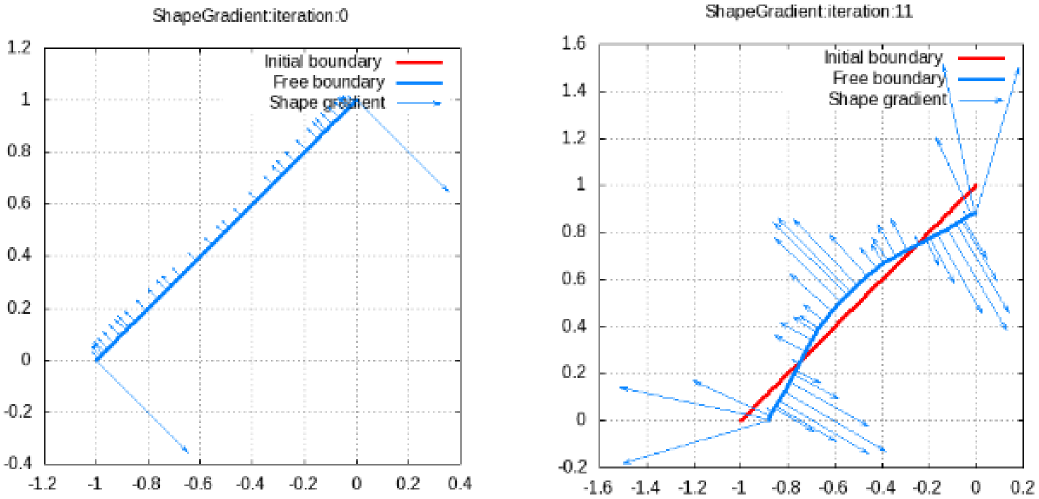
We recall that the analytical solution of this problem is an arc of circumference, as verified in the literature ([BDVS12]) thus we expect the configurations arising from Pareto-optimal



(a) Initial configuration of the structure.

(b) Final configuration of the optimal shape.

Figure 5.3: Comparison between the initial and the final shape of the structure.



(a) Initial shape gradient.

(b) Final shape gradient.

Figure 5.4: Comparison between the initial and the final shape gradient on the moving boundary.

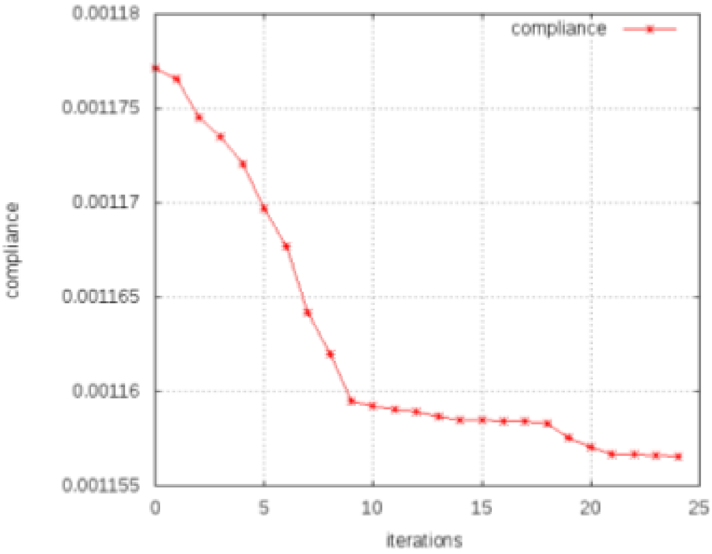


Figure 5.5: Convergence of the algorithm based on the shape gradient.

points to generate shapes which are perturbations of the exact one.

In [BDVS12] some information about the shape gradient for the two-dimensional squared plate are available. In figure 5.4 we observe the changes in the shape gradient while the moving boundary evolves towards a circle which is the final configuration depicted in figure 5.3b. Eventually in figure 5.5 we report the graphic of the convergence of the optimization procedure and we observe that in a small number of iterations the *a priori* fixed tolerance is fulfilled.

5.3 MGDA optimization

In this section we present the results of the multiobjective shape optimization problem obtained using the previously introduced variants of *MGDA* coupled with the *IGA* solver as discussed in chapter 4.

We recall that within a multiobjective optimization paradigm we seek the Pareto-optimal solutions thus we expect to find final configurations *similar but not identical* to the analytical optimal one from section 5.2.
This consideration is actually reasonable since the new problem as formulated in section 3.5.2 aims at concurrently minimizing both the compliance and the variation of the volume whereas in the original problem the latter was treated as a constraint not as an objective functional. Moreover, within the framework of multiobjective optimization a global optimal solution does not exist and we seek configurations representing a trade-off between the minimization of two antagonistic criteria. For this reason, in the following subsections we present the graphics of the Pareto fronts identified using several variants of *MGDA* and we highlight the non-dominance of the final configurations. Eventually, we propose some snapshots of the resulting optimal shapes and we verify that they can be interpreted as perturbations of the optimal arc determined in the analytical case.

5.3.1 Multiple-Gradient Descent Algorithm

Main goal of *MGDA* is to identify a complete characterization of the Pareto front that for this kind of problems requires 20 to 30 points. We consider the previously introduced *NURBS* parametrization and we choose a design point in $\mathbb{R}^6$ that accounts for the coordinates of the control points of the boundary Γ as described in table 5.1. Starting from an admissible design point, *MGDA* evolution assures that all criteria concurrently improve. In figure 5.6 we report *MGDA* evolution from a given design point randomly chosen among a set of admissible values (top-right) and we observe that both the compliance and the volume variation decrease at each iteration until a final Pareto-stationary point is achieved (bottom-left). Moreover we notice that after a small number of iterations, the variations over the functional values are very small thus we conjecture that the algorithm almost converged. As a matter of fact, figure 5.7 shows the evolution of $\boldsymbol{\omega}$ with respect to the iterations and it is straightforward to observe that after few iterations its value fulfills a sufficiently small tolerance. This respects the scenario from theorem 2.5 since we can identify a descent direction common to all criteria until a Pareto-stationary point is achieved.

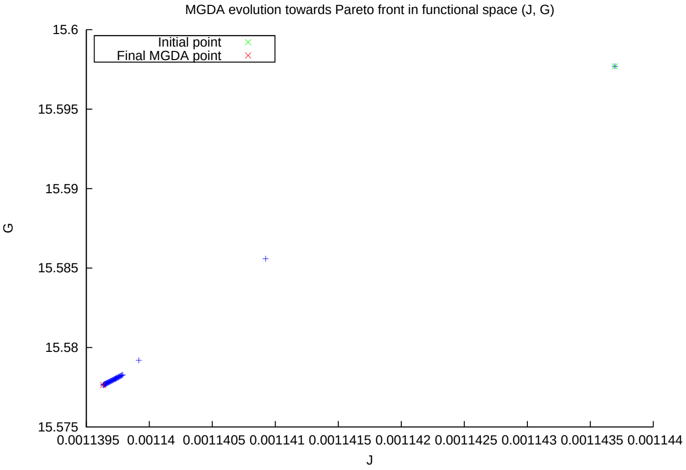


Figure 5.6: Evolution of the functional values in the space $J-G$ using classical *Multiple-Gradient Descent Algorithm*. Starting point on the top-right corner and final point on the bottom-left one.

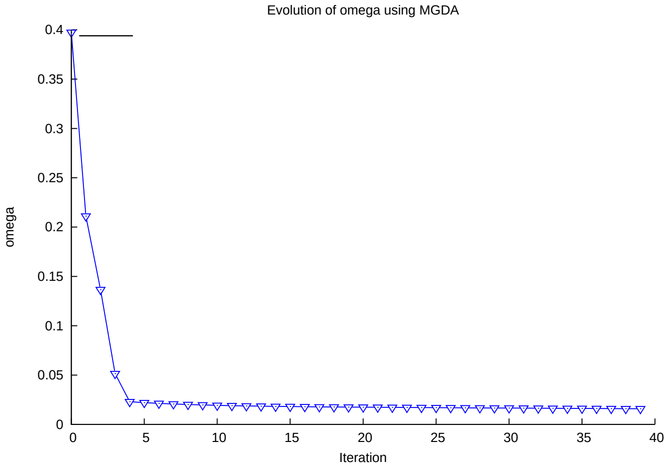


Figure 5.7: Evolution of values of $\boldsymbol{\omega}$ using classical *Multiple-Gradient Descent Algorithm*.

The analysis discussed so far has to be performed for every configuration, thus we apply *Multiple-Gradient Descent Aalgorithm* strategy as many times as the number of initial design points we are considering for our optimization problem. In particular, in figure 5.8 we show the convergence of *MGDA* iterates towards some Pareto-stationary points starting from several feasible initializations. Moreover figure 5.9 describes the corresponding evolution of the minimal-norm element $\boldsymbol{\omega}$ starting from different initial design points. We notice that in every scenario the minimal-norm element decreases at each iteration and eventually converges to a Pareto-stationary design point as expected from the theory and from the single case presented in figure 5.7.

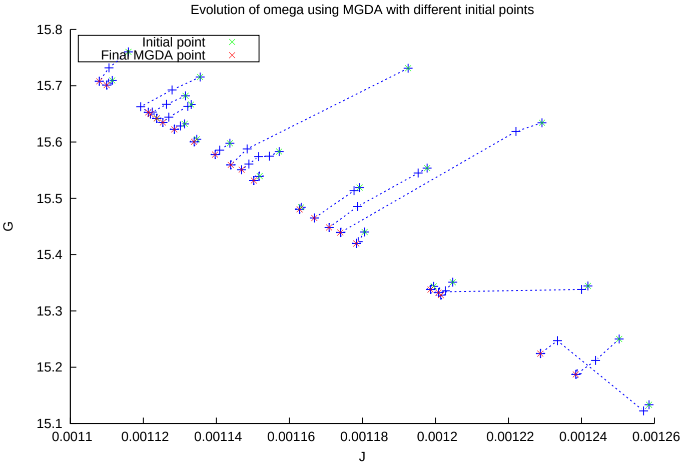


Figure 5.8: Evolution of the functional values in the space $J - G$ starting from several initial design points and using classical *Multiple-Gradient Descent Algorithm.*

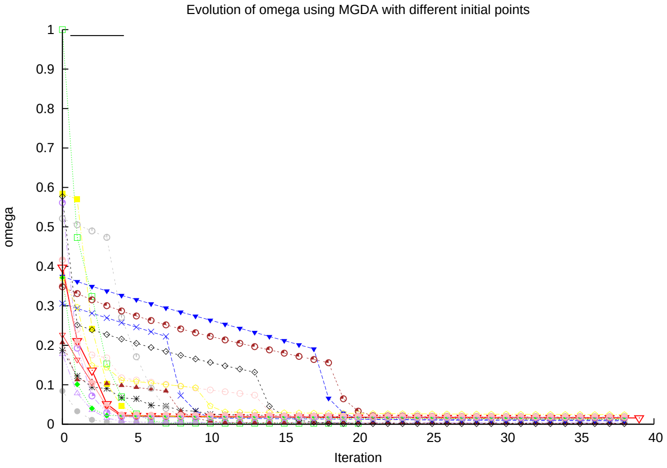


Figure 5.9: Evolution of the values of $\boldsymbol{\omega}$ starting from several initial design points and using classical *Multiple-Gradient Descent Algorithm.*

Remark 5.1. It is important to notice that final *MGDA* points do not depend only on the initial data configuration under analysis. A major role within the optimization algorithm is played by the optimal step size $\widetilde{\rho}$. As a matter of fact, even if the step size is adaptively updated by means of the computational strategy presented in section 2.2.2, the initial value of ρ could strongly influence the value of the iterates at early steps driving the evolution

of the design points away from the closest available Pareto point.
Moreover, if the problem requires the imposition of compatibility conditions (such as the symmetry condition introduced in section 5.1), additional issues may arise. As a matter of fact, the abrut change of direction in the evolution of the green design point on the bottom-right of figure 5.8 can be explained in this sense: along the previously analyzed direction the risk of violating the symmetry condition is particularly high, thus a penalty term causes the optimization procedure to identify an alternative descent direction that fulfills the constraint. Hence these scenarios highlight a sensitivity of the method to initial parameter calibration and have to be carefully managed in order to avoid unwanted spurious oscillations of $\boldsymbol{\omega}$.

Eventually we present the complete Pareto front generated by using the classical version of *Multiple-Gradient Descent Algorithm* (Fig. 5.10) whereas the analysis of the shapes associated with the final configurations is postponed to the end of this section when we will compare them with the ones generated by the other variants of the shape optimization procedure discussed so far.

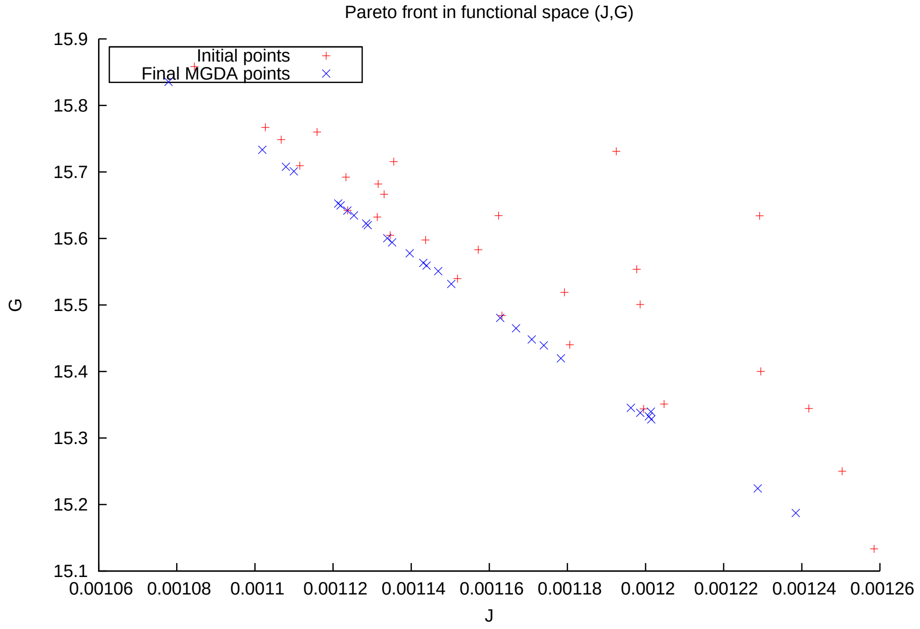


Figure 5.10: Pareto front in the space $J - G$ using classical *MGDA* procedure.

5.3.2 Kriging-assisted MGDA

As pointed out during the description of the algorithm in section 2.4.1, kriging-assited *MGDA* procedure differs from the classical *Multiple-Gradient Descent Algorithm* because it couples the descent method with a surrogate statistical-based prediction model. In this way, the overall strategy performs less computations by directly invoking the *IGA* solver and estimates the values of the objective functionals by means of the kriging model.

An initial data set of 20 design points constituting a Latin Hypercube in $\mathbb{R}^6$ (see table 5.1) is considered. Figure 5.11 depicts the initial data set and the final enriched version after the application of the kriging-based prediction strategy. A remarkable improvement of both criteria is achieved and a Pareto front begins to form. From a computational point of view, kriging-assisted *MGDA* is responsible for a significant reduction in computing efforts

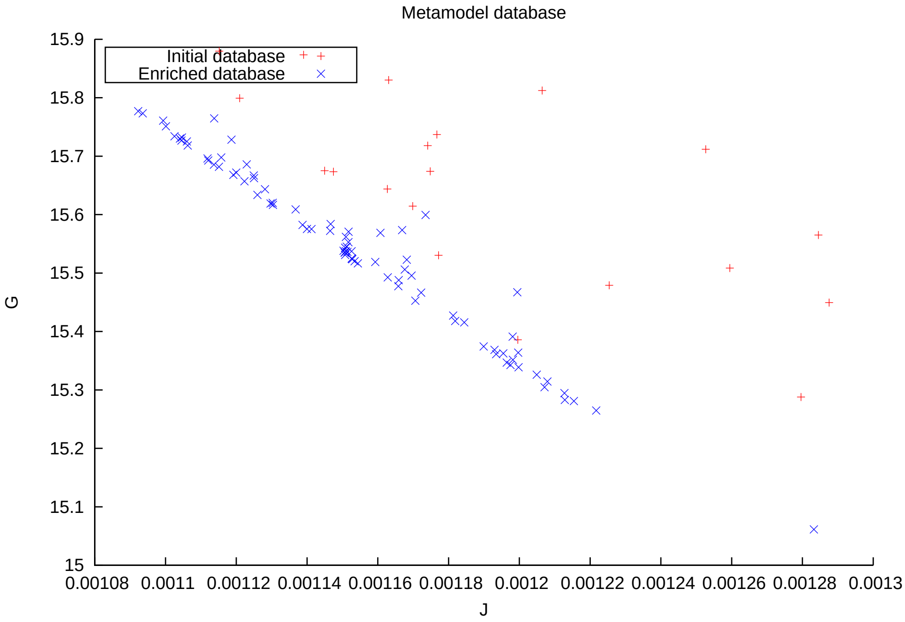


Figure 5.11: Database enrichment as consequence of the combined application of *Multiple-Gradient Descent Algorithm* and kriging-based prediction strategy.

since the *IGA* solver has to be invoked only when *MGDA* iterates already converged, that is once for every blue cross in figure 5.11. Moreover, a proper description of the Pareto front can be achieved by means of subsequent applications of the metamodel strategy as summarized in table 4.1.

In figure 5.12 we observe the Pareto front generated by means of this version of the algorithm and we notice that it is less extended than the one in figure 5.10. In particular, it seems that the extreme values of the front are not achieved by this version of the algorithm. This issue is not likely related to the initial distribution of the design points, since a strategy to generate uniformly distributed random points is performed: design points associated with *Latin Hypercube Sampling* are thought to map the whole domain thus we conjecture that the limited extension of the Pareto front described by final *MGDA* iterates is due to a limitation in the kriging-based algorithm. However further investigation should be performed on the parameters involved in the model in order to obtain a better tuning able to account for the builtin sensitivities.

Even if we conjecture that kriging-based metamodels tend to underestimate the tail values of the Pareto front, the information contained in final *MGDA* points is sufficient to provide a trend of the first Pareto set.
Moreover the reduced computational effort is significant since this strategy applies to a set of design points thus is able to provide a complete description of the Pareto front after 10 to 15 runs of the algorithm. In particular, for this problem in computational mechanics, the computing time for the kriging-assisted procedure is comparable with the one required for the evolution of two single points in classical *MGDA*. Hence the overall strategy is still competitive with respect to computing times.

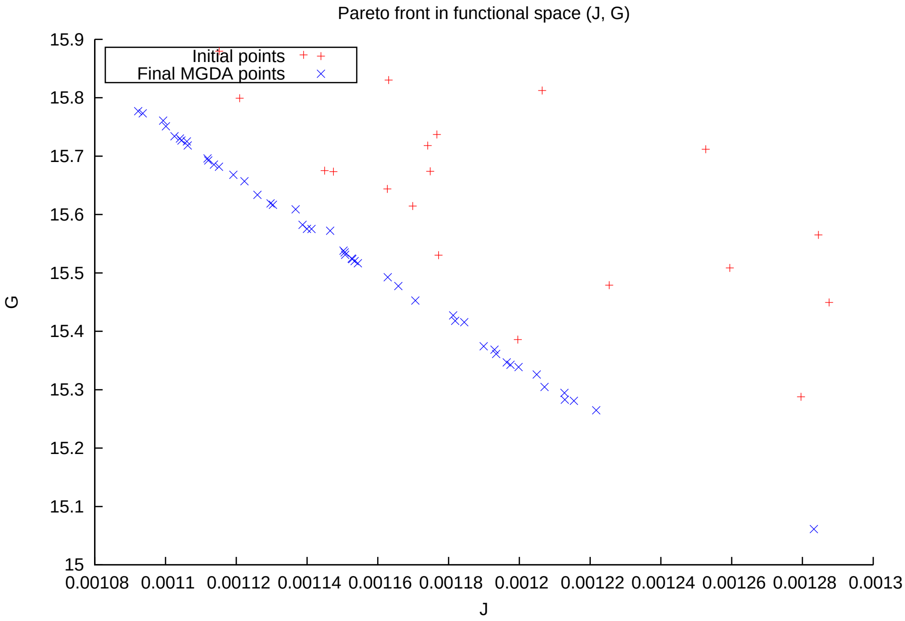


Figure 5.12: Pareto front in the space $J - G$ using kriging-based metamodel-assisted *MGDA* procedure.

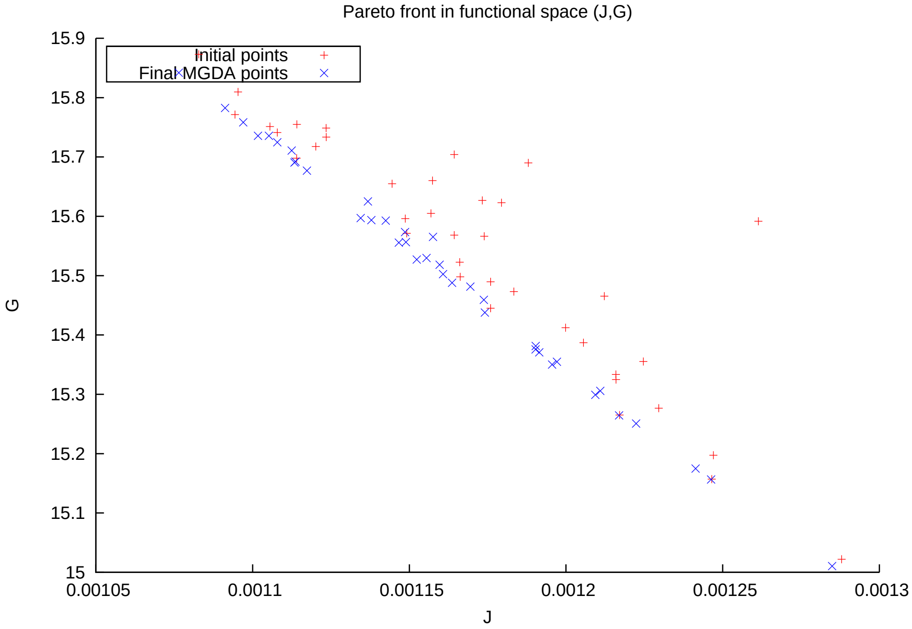


Figure 5.13: Pareto front in the space $J - G$ using *MGDA* combined with the information on the gradients extracted from the *IGA* linear elasticity solver.

5.3.3 MGDA using analytical gradients

As in previous sections, first of all we present the Pareto front generated by classical *Multiple-Gradient Descent Algorithm* enhanced by the use of the information provided by the analytical gradients in the parametric space extracted from the *IGA* solver.

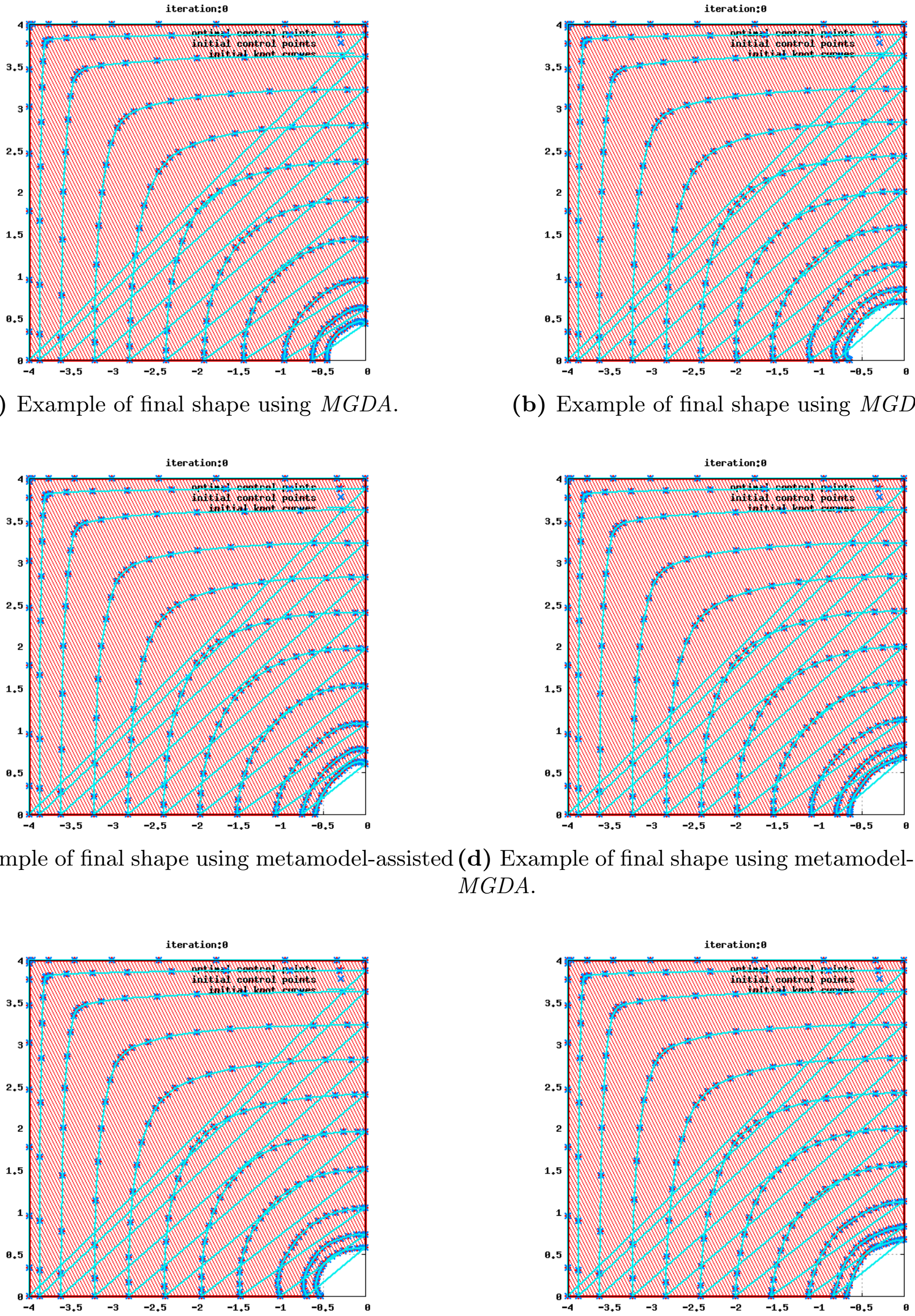
We observe that the general distribution pattern of the Pareto points in figure 5.13 is the same as in the case of classical *MGDA*, meaning that the approximation of the gradients using a second order centered Finite Difference scheme is sufficiently accurate. Thus we can infer that classical *MGDA* procedure is an accurate tool for the treatment of multiobjective optimization problems.

On one hand the classical algorithm requires more computational power to approximate the gradients but on the other hand, it considers a lower number of optimization variables. This is due to the difficulties faced in the definition of a proper projection operator as in equation (4.12). To avoid this problem, the optimization routines are executed directly in the enriched space whose dimension is higher than the original one (see table 5.1). Thus, even if the global optimization algorithm presented in section 4.4 reduces the computational cost of the single iteration of *MGDA*, the additional number of control points increases the computing time. Hence the final procedures are comparable with respect to the CPU time required for the execution. Eventually in figure 5.14 and 5.15 we present four final configurations for each described algorithm. We highlight that the final shapes of the free boundary Γ of the plate are physically consistent with the imposed external forces; moreover, the shapes can be interpreted as perturbations of the arc of circumference expected from the analytical results in the literature.

Most of the differences among the final configurations in figure 5.14 and 5.15 are due to the choice of fostering a better optimization for one criterion rather than the other, that is minimizing first the compliance or the variation over the volume constraint. In particular, in the first figure we report the final shapes corresponding to some design points where the minimization of the compliance is fostered and consequently the domain is larger. On the contrary, the second set describes the case where the minimization of the volume is fostered thus we expect larger value of the compliance functional. Nevertheless, the recurrent arc shape and the limited variations of the volume make all the optimal shape acceptable.

Algorithm	Applies to	N. optimization variables	N. execution	CPU time
MGDA	30 single design points	6 (2 moving points 1 horizontally sliding 1 vertically sliding)	40 *MGDA* runs per design point	8 hours per design point
Kriging-assisted *MGDA*	Data set of 20 design points	6 (2 moving points 1 horizontally sliding 1 vertically sliding)	10 kriging-assisted runs each of which performs 60 *MGDA* iterations	15 hours per data set
MGDA using analytical gradients	30 single design points	34 (16 moving points 1 horizontally sliding 1 vertically sliding)	40 *MGDA* runs per design point	6 hours per design point

Table 5.1: Comparison of the computational cost of the different strategies based on *Multiple-Gradient Descent Algorithm* for shape optimization using *IsoGeometric Analysis.*



(a) Example of final shape using *MGDA*.

(b) Example of final shape using *MGDA*.

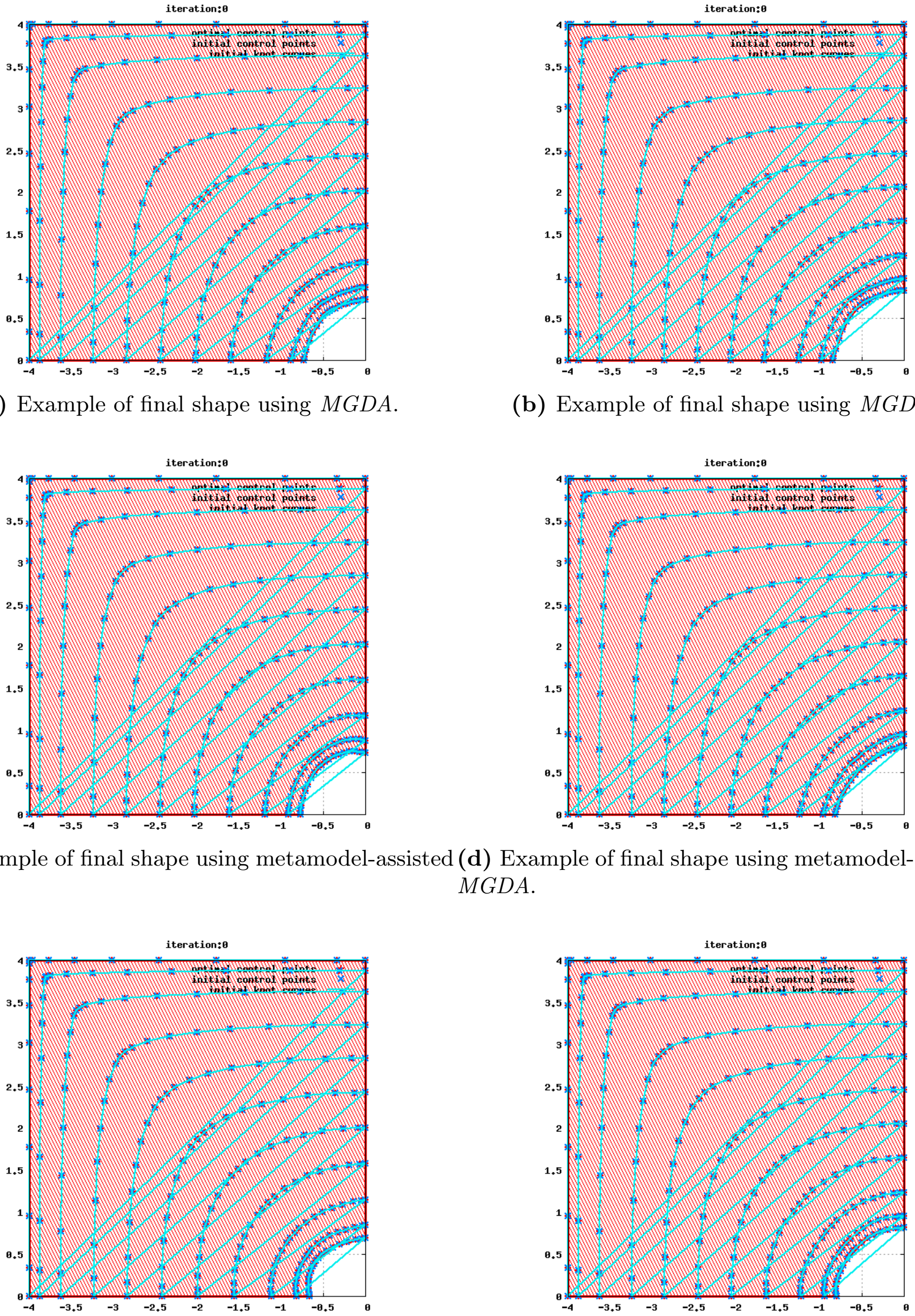
(c) Example of final shape using metamodel-assisted *MGDA*.

(d) Example of final shape using metamodel-assisted *MGDA*.

(e) Example of final shape using exact gradients-enhanced *MGDA*.

(f) Example of final shape using exact gradients-enhanced *MGDA*.

Figure 5.14: Final configurations generated by the described variants of *Multiple-Gradient Descent Algorithm*. Optimal shapes corresponding to design points in the top part of the Pareto front.



(a) Example of final shape using *MGDA*.

(b) Example of final shape using *MGDA*.

(c) Example of final shape using metamodel-assisted *MGDA*.

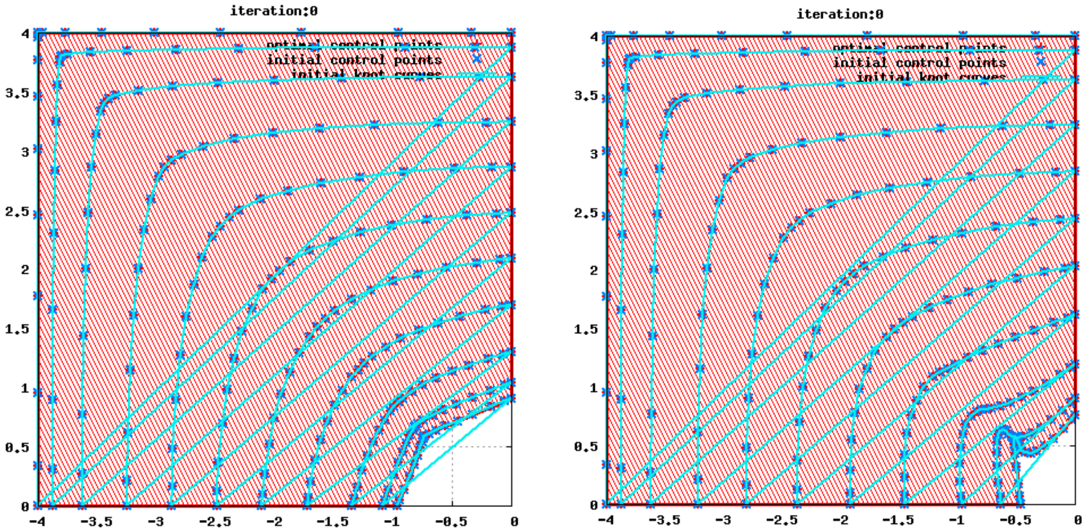
(d) Example of final shape using metamodel-assisted *MGDA*.

(e) Example of final shape using exact gradients-enhanced *MGDA*.

(f) Example of final shape using exact gradients-enhanced *MGDA*.

Figure 5.15: Final configurations generated by the described variants of *Multiple-Gradient Descent Algorithm*. Optimal shapes corresponding to design points in the bottom part of the Pareto front.

Remark 5.2. To maintain physically meaningful domains, overlapping of the patches have to be avoided. In the vicinity of the hole, one can notice a crossing of some control points that could generate issues from a numerical point of view. However, the parameterization remains injective which enables to solve the resulting algebraic systems.
As pointed out in the previous chapter, close adjacency of control points or swapping control points may lead to singularities in the solution constructed using *IsoGeometric Analysis* and in figure 5.16 we report two non-physical solution resulting from these scenarios. This confirms the necessity of applying a filter to the set of design points arising from *MGDA* iterates to avoid non-physical configurations.



(a) Adjacency of control points.

(b) Swapping control points.

Figure 5.16: Non-physical final configurations due to singularities in the control polygon.

5.4 Comparison with the optimization results based on genetic algorithms

A detailed introduction to evolutionary strategies and genetic algorithms for multiobjective optimization is available at [KC00] and [FF98]. As proposed in [ZDD11], here we compare the results produced by the gradient-based approach *Multiple-Gradient Descent Algorithm* with the ones obtained with a classical genetic algorithm widely known in the literature, the *Pareto Archived Evolutionary Strategy*. In particular, we present the description of the Pareto front produced by *Pareto Archived Evolutionary Strategy* and we compare it with the result of the computation performed using classical *MGDA* with numerical approximated gradients, kriging-assisted *MGDA* and *MGDA* using analytical gradients.

In figure 5.17 we report the final population generated by the genetic algorithm and we highlight the elements that belong to the Pareto front. It is straightforward to observe that the number of configurations to be evaluated is extremely large and the actual set we are looking for is just a minor fraction of it.

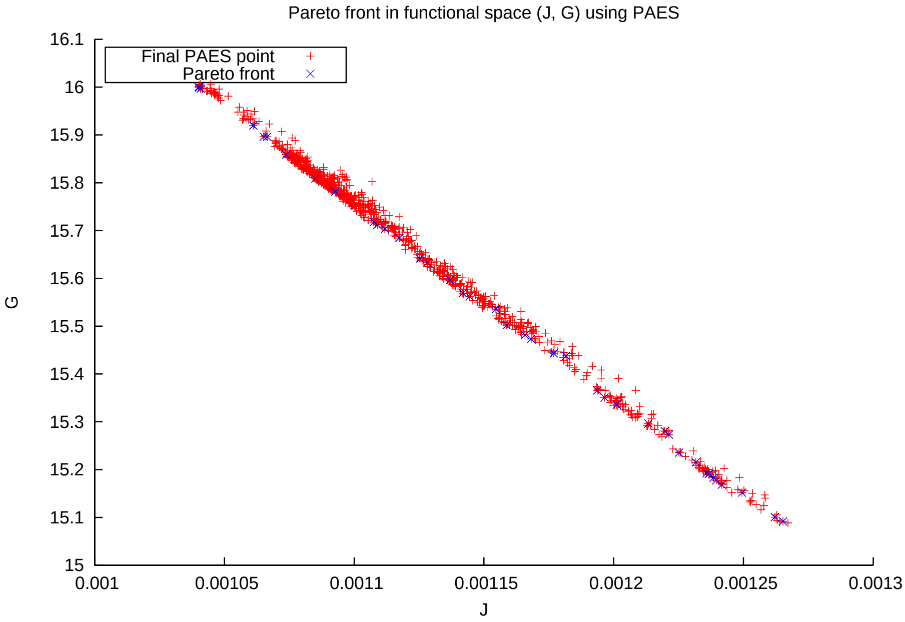


Figure 5.17: Pareto front in the space $J - G$ using *Pareto Archived Evolutionary Strategy*.

Nevertheless, the output of *PAES* computations is a detailed description of the Pareto set and is comparable with the ones provided in figures 5.10, 5.12 and 5.13. As a matter of fact, *PAES* results show a wider range for the Pareto points than the *MGDA*-based outputs thus we conjecture that the evolutionary strategy tends to provide a more complete description of the front than the gradient-based approaches.

Thus we can conclude that genetic algorithms are usually able to detect almost the totality of the minimima even in presence of multimodal functionals to be optimized and their main drawback is the high computational cost required by the simulations.
On the contrary, *Multiple-Gradient Descent Algorithm* and its variants showed themselves to be extremely effective converging to Pareto-optimal design points. Thus a good approximation of the Pareto front is achieved and the overall performances strongly improved. However, the discussed strategies may present some issues when dealing with multimodal functionals or initial design points not uniformly distributed over the domain: in these scenarios, *MGDA* iterates may get trapped within the domain of attraction of a given design point, not being able to provide a full description of the Pareto set anymore.

Properly calibrating the initial population to run the genetic algorithm represents a major issue for evolutionary strategies. Thus, in order to reduce the computational effort required to run full *PAES*, we choose as initial population a set of Pareto-optimal points provided by *MGDA* iterations.
On one hand we use *PAES* as a tool to cross-validate the accuracy of *Multiple-Gradient Descent Algorithm* and figure 5.18 confirms the good results previously highlighted for the variants of *MGDA* tested in section 5.3. On the other hand, *MGDA* is used to speed *Pareto Archived Evolutionary Strategy* up reducing the overall time to run the simulations and thus enhancing its efficiency.
For this reason, the results arising from the gradient-based strategy and the evolutionary algorithm are not independent but it is well known in the literature that the velocity of these approaches is not comparable. Eventually, a summary of the complexity of the implemented strategies and the corresponding computational costs is reported in table 5.1.

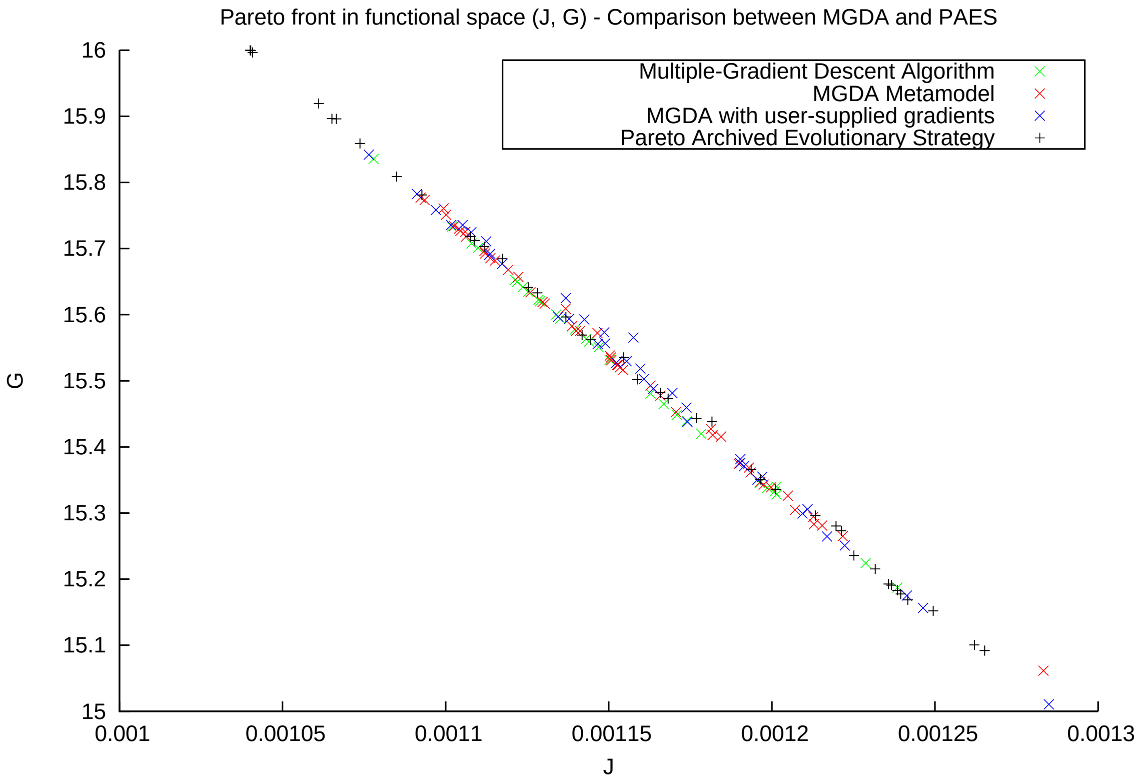


Figure 5.18: Comparison among the Pareto fronts generated by *MGDA*-based methods and the genetic algorithm known as *Pareto Archived Evolutionary Strategy* in the space $J - G$.

From these observations, we infer that an hybrid approach able to combine gradient-based algorithms with evolutionary strategies could represent a good trade-off among the procedures discussed so far. Moreover, we expect this approach to have better performances than other multiobjective optimization strategies commonly used in the literature.

Conclusion

We carried out a methodological work that aims at formulating a shape optimization problem using a multiobjective optimization paradigm; moreover, we treated it by means of deterministic algorithms that employ gradient-based strategies in MultiDisciplinary Optimization. Some theoretical results are stated and a detailed description of the steps to properly formulate the problem is given.

Main achievement of this work is the successful analysis of a shape optimization problem in computational mechanics within a multiobjective optimization framework. In particular, both *Multiple-Gradient Descent Algorithm* and kriging-assisted *MGDA* provided good results in identifying the Pareto front for the problem under analysis. Moreover, the accuracy of classical *MGDA* procedure based on the numerical approximation of the gradients of the objective functionals was confirmed by means of the implementation of a variant of the algorithm that uses user-supplied information for the gradient vectors. In particular, starting from shape derivatives we extracted the information on the gradients in the parametric space and we were able to check our previous results. However this procedure did not result in high computational advantages since the new optimization problem had to be set in a higher dimensional space.

Then we applied the implemented methods to a linear elasticity problem formulated using *IsoGeometric Analysis*. On one hand, *NURBS*-based formulations allowed us to obtain a higher regularity on the domain parametrization and on the solution; nevertheless some issues arose due to critical topics in differential geometry. As a matter of fact, using control points as design variables for global optimization procedures, we noticed that deformations incompatible with the space of admissible solutions could appear. Further analysis should focus on the role of filtering strategies in order to prevent the presence of design points too close to one another or design points with multiplicity higher than 1 which may cause regularity to decrease and singularities to appear within an isogeometric paradigm.

Concerning kriging-assisted optimization, we observed that the range of description of the Pareto front was actually narrower than the one generated by classical *MGDA*. We conjecture that kriging-based methods tend to underestimate the importance of extreme values in Pareto sets but further investigations should be carried on to examine this topic. Knowing the information related to the gradients in the design space could enhance previously presented metamodel: for example, by means of a co-kriging strategy, a surrogate model could be generated by using the information from both the objective functionals and their gradients.

Eventually we checked the results given by *MGDA*-based methods. On one hand, we verified the consistency of the optimal shapes from *MGDA* simulations with the ones given by the single objective shape-based *Steepest Descent Method.* Moreover the Pareto fronts arising from *MGDA* were cross-validated by using *Pareto Archived Evolutionary Strategy.*

Final descriptions of the Pareto fronts were consistent and a further development could face the formulation of an hybrid method able to take advantage of the best properties of the two strategies, that is extending the range of Pareto points generated as evolutionary strategies do and reducing the computational cost as gradient-based methods do.

Future developments

First of all, we remark the importance of deepening the investigation on the topics of competitive optimization since great advantages are expected from the implementation of robust strategies able to meet user specific goals and deal with uncertainties in the data.

Beside this and other possible developments already highlighted, here we propose some more general extensions that could represent leading investigation lines in the field of computational mechanics and optimization.

Within the framework of *IsoGeometric Analysis*, main interests focus on local refinement and integration. The former topic has already been treated in several works and major prospectives focus on the topics of *Analysis-suitable T-splines* and *Hierarchical B-splines.* For the latter, several issues are still open and interesting results are expected to arise from the investigation of *IGA* collocation methods.

Improvements in these fields would have a major impact in every application in computational mechanics, starting from the one we studied in this thesis.

From a modelling point of view, the problem we dealt with was quite elementary. An interesting extension would rely on modelling more realistic and general situations, for example non-linear elasticity problems and non-local interactions in structural mechanics. We expect several points of contact to arise with the research topics related to fracture mechanics and a promising extension could focus on *IGA* XFEM methods.

Concerning the field of optimization, on one hand some issues related to *Multiple-Gradient Descent Algorithm* are still open, especially for the infinite dimensional case. On the other hand, from a computational point of view, a generalization of statistical metamodels in order to take into account adjoint variables could be performed. Thus the implementation of optimization strategies assisted by co-kriging models and massively parallelization are expected to strongly reduce the overall computational cost of the algorithms.

Eventually, further problems could be treated by means of *MGDA* variants, for example in the field of three-dimensional compressible fluid dynamics: in this framework, a structural criterion and an aerodynamic one are set to be the objective functionals and a volume constraint and a lift constraint define the matrix for the projection onto the subspace of admissible solutions. All the results stated in this work hold and a more complex problem arising from industry could be treated within a multiobjective optimization paradigm.

Appendix A

Nash games for competitive optimization

In this appendix, we present a short introduction to competitive optimization. Basic idea of this optimization strategy relies on seeking a configuration that improves one or more criteria flagged as *important* or *critical* without excessively worsening the others.
Starting point is a Pareto-stationary design point $\mathbf{Y}^{(0)}$ where (2.2) is verified. If the performance of the design point is already satisfactory the procedure can be interrupted. Otherwise a Nash game can be established by means of an appropriate split of the variables. The splitting process is a critical step of this strategy: in the case of two disciplines, it may be guided by the analysis of the spectral properties of local Hessian matrices whereas higher dimensional problems may require the development of alternative approaches.

Definition and properties of Nash equilibria

Let us assume that we have two players controlling the subvectors $\mathbf{Y}_1$ and $\mathbf{Y}_2$ such that the global vector of design variables reads as $\mathbf{Y} = (\mathbf{Y}_1 \ , \ \mathbf{Y}_2)^T$.

Definition A.1 (*Nash equilibrium*)**.** The vector $\overline{\mathbf{Y}} = (\overline{\mathbf{Y}}_1, \overline{\mathbf{Y}}_2)^T$ is said to realize a *Nash equilibrium* of the criteria $J_1(\mathbf{Y})$ and $J_2(\mathbf{Y})$ if and only if

$$\begin{cases} \overline{\mathbf{Y}}_1 = \underset{\mathbf{Y}_1}{\text{argmin}} \ J_1(\mathbf{Y}_1, \overline{\mathbf{Y}}_2) \\ \overline{\mathbf{Y}}_2 = \underset{\mathbf{Y}_2}{\text{argmin}} \ J_2(\overline{\mathbf{Y}}_1, \mathbf{Y}_2) \end{cases} \tag{A.1}$$

In script A.1 we present an evolutionary strategy to compute this equilibrium point. We notice that the split of the design variables into two subspaces allows to easily parallelize the algorithm and this is very promising since it represents a good starting point for the implementation of computationally efficient optimization procedures.

Listing A.1: Evolutionary strategy of a Nash game

```
1. Initialize the subvectors $\mathbf{Y}_i = \mathbf{Y}_i^{(0)}$ , $i = 1,2$;
2. Assign each subvector to a different player as exclusive strategy:
 (a) Player 1: Set $\mathbf{Y}_2 = \mathbf{Y}_2^{(0)}$ and perform $K_1$ optimization steps of $J_1(\mathbf{Y}_1, \mathbf{Y}_2^{(0)})$
     by iterating on $\mathbf{Y}_1$ alone. The optimal result is $\mathbf{Y}_1^{(K_1)}$;
 (b) Player 2: Set $\mathbf{Y}_1 = \mathbf{Y}_1^{(0)}$ and perform $K_2$ optimization steps of $J_2(\mathbf{Y}_1^{(0)}, \mathbf{Y}_2)$
     by iterating on $\mathbf{Y}_2$ alone. The optimal result is $\mathbf{Y}_2^{(K_2)}$;
3. Update both subvectors $\mathbf{Y}_i^{(0)} = \mathbf{Y}_i^{(K_i)}$ , $i = 1,2$ and go to step 2; otherwise a
```

```
Nash equilibrium is achieved and the competitive optimization procedure
is stopped.
```

Remark A.2. Assume that $\overline{\mathbf{Y}} = (\overline{\mathbf{Y}}_1, \overline{\mathbf{Y}}_2)$ realizes a Nash equilibrium of the criteria $J_1(\mathbf{Y})$ and $J_2(\mathbf{Y})$. Let Φ and Ψ be two arbitrary smooth and strictly-monotone increasing functions, then $\overline{\mathbf{Y}}$ also realizes a Nash equilibrium of the criteria $\Phi[J_1(\mathbf{Y})]$ and $\Psi[J_2(\mathbf{Y})]$. Thus the notion of Nash equilibrium is not only independent of the physical units used for the criteria but also of the possible changes in scales. The invariance of the Nash equilibrium solution to units and scales is an interesting property that allows to formulate robust automatic optimization algorithms. As a matter of fact, this is in contrast with classical approaches in MultiDisciplinary Optimization (Section 1.2) where the penalty parameters have to be defined by the user. Thus this strategy results in more robust methods where the sensitivity to *a priori* unknown parameters is minimized.

From the property stated in remark A.2 we conclude that the equilibrium solution - either unique or not - is only influenced by the split of the design vector, which is commonly referred to as *split of territory.* As a matter of fact, the value of Nash equilibrium only depends on the strategy by which each virtual player is allocated a subspace of action and in the algorithm this results in different choices for the subdivision of the vector of design variables $\mathbf{Y} = (\mathbf{Y}_1\ ,\ \mathbf{Y}_2)^T$.

Split of territory

Let $J_1(\mathbf{Y})$ be the primary functional with respect to which sub-optimality should be maintained and $J_2(\mathbf{Y})$ the secondary functional to be optimized under possible constraints. First of all we perform the optimization of the principal criterion $J_1(\mathbf{Y})$ alone and we denote by $\mathbf{Y}_1^*$ the corresponding optimal solution computed with respect to the totality of the design variables N under K active constraints. Then we perform a multiobjective and competitive optimization step to determine a Nash equilibrium between $J_1(\mathbf{Y})$ and $J_2(\mathbf{Y})$.

We introduce a splitting matrix S of dimension $N \times N$ and we define the subvectors $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_{N-p})^T \in \mathbb{R}^{N-p}$ and $\mathbf{V}(\mathbf{v}_p, \dots, \mathbf{v}_1)^T \in \mathbb{R}^p$ as strategies or territories of two virtual players whom the exclusive optimization of $J_1(\mathbf{Y})$ and $J_2(\mathbf{Y})$ is assigned.

$$\mathbf{Y} = \mathbf{Y}(\mathbf{U}, \mathbf{V}) = \mathbf{Y}_1^* + S \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} \tag{A.2}$$

In particular, the subspace spanned by the first $N - p$ column vectors of S can be viewed as the territory assigned to the first player in charge of minimizing the primary criterion $J_1(\mathbf{Y})$ and the subspace spanned by the remaining p columns as the territory assigned to the second player to minimize the secondary functional $J_2(\mathbf{Y})$.

Starting from equation (A.1) we can write the Nash game as a coupled optimization problem (here we consider only minimization problems):

$$\begin{cases} \min\limits_{\mathbf{U} \in \mathcal{N}_{ad}} J_1 \left[\mathbf{Y}(\mathbf{U}, \overline{\mathbf{V}})\right] \\ \min\limits_{\mathbf{V} \in \mathbb{R}^p} J_2 \left[\mathbf{Y}(\overline{\mathbf{U}}, \mathbf{V})\right] \end{cases} \tag{A.3}$$

The first problem is a constrained minimization in which the effect of the active constraints $g_k(\mathbf{Y}) = 0\ ,\ k = 1, \dots, K$ is taken into account by introducing the set of admissible solutions

$$\mathcal{N}_{ad} = \left\{ \mathbf{U} \in \mathbb{R}^{N-p} \quad \middle| \quad g_k \left[\mathbf{Y}(\mathbf{U}, \overline{\mathbf{V}})\right] = 0 \quad , \quad \forall k = 1, \dots, K \right\} \tag{A.4}$$

whereas the second one is an unconstrained minimization problem.

The dimension p of the subvector $\mathbf{V}$ which controls the subspace of action of the second player is adjustable, that is $p \geq 1$; on the other hand, the dimension of the subvector $\mathbf{U}$ must be at least equal to 1 and to the number of active constraints K:

$$\begin{cases} N-p \geq 1 \\ N-p \geq K \end{cases} \quad \Rightarrow \quad 1 \leq p \leq N - \max\{K, 1\}$$

In the limiting case when $N-p = K$, the minimization of $J_1(\mathbf{Y})$ under the given constraints reads as the correction of the K components of the subvector $\mathbf{U}$ in order to satisfy the K scalar constraints.

We analyze the procedure to perform the split of the territory after the initial optimal design point $\mathbf{Y}_1^*$ for the primary criterion is achieved. The splitting strategy is driven by the analysis of sensitivity of $J_1(\mathbf{Y})$, that is the study of the the admissible infinitesimal perturbations of the parameters about $\mathbf{Y}_1^*$: we seek among the perturbations that lie in the subspace identified as the territory of the secondary criterion, the one which causes the least possible degradation of the primary criterion with respect to the minimum previously achieved. Given $\epsilon \in [0, 1]$, we consider a perturbation in the direction of unit vector $\boldsymbol{\omega} \in \mathbb{R}^N$:

$$J_1(\mathbf{Y}_1^* + \epsilon\boldsymbol{\omega}) = J_1(\mathbf{Y}_1^*) + \epsilon\nabla J_1(\mathbf{Y}_1^*) \cdot \boldsymbol{\omega} + \frac{\epsilon^2}{2}\boldsymbol{\omega} \cdot H_1(\mathbf{Y}_1^*)\boldsymbol{\omega} + \mathcal{O}(\epsilon^3) \tag{A.5}$$

Based on the previous idea, the first $N-p$ components of $\boldsymbol{\omega}$ are dedicated to the first player whereas the remaining p are associated with the second player. Moreover we recall that the direction of maximum sensitivity of $J_1(\mathbf{Y})$ is given by its gradient at $\mathbf{Y} = \mathbf{Y}_1^*$, that is $\nabla J_1(\mathbf{Y}_1^*)$.
Thus the following two conditions should be satisfied by the basis $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N)^T$:

1. the first $N-p$ elements of $\boldsymbol{\omega}$ should span the gradient $\nabla J_1(\mathbf{Y}_1^*)$;
2. the difference $|J_1(\mathbf{Y}_1^*+\epsilon\boldsymbol{\omega}) - J_1(\mathbf{Y}_1^*)|$ should be as small as possible when the direction is a tail element of the basis.

When $\mathbf{Y} = \mathbf{Y}_1^*$ the optimality condition on the Lagrangian functional implies that the gradient of the primary functional is a linear combination of the K active constraint gradients:

$$\nabla_{\mathbf{Y}} L(\mathbf{Y}, \boldsymbol{\lambda}) = 0 \quad \Leftrightarrow \quad \nabla J_1(\mathbf{Y}_1^*) + \sum_{k=1}^{K} \lambda_k \nabla g_k = 0$$

Thus the first condition is achieved by enforcing the first K elements of the basis to have the same span as the gradients of the K active constraints. For this reason, the $\boldsymbol{\omega}_k$'s $k = 1, \ldots, K$ are obtained by applying the Gram-Schmidt orthogonalization procedure to the projected gradients as described in section 2.5. In [Dés11] the best choice for $\boldsymbol{\omega}_k$'s $k = 1, \ldots, K$ is proved to be the set of the eigenvectors of the matrix $H_1' = PH_1(\mathbf{Y}_1^*)P$ where H_1 is the Hessian matrix and P the projection matrix as in equation (2.33): these eigenvectors contain the null space of the projection matrix P, thus the first condition is satisfied if the ordering is such that these vectors appear first; moreover the basis is orthogonal thus the tail elements are orthogonal to the first K, that is they are orthogonal to $\nabla J_1(\mathbf{Y}_1^*)$ and also the second condition is satisfied.

Under the assumptions made above and within the described framework, some general results may be prooved. In particular, in his work, J.-A. Désidéri proves the optimality of

the orthogonal decomposition of the Hessian matrix for the splitting strategy; moreover he characterizes the properties of Nash equilibria and establishes a relationship with Pareto-optimality. For further details about this theorem and its proof we refer to [Dés07].
Here we just recall two major achievements related to Nash equilibria in competitive optimization.
An existence result
Let us introduce a continuation parameter $\zeta \in [0,1]$ in order to define a concurrent optimization problem

$$\begin{cases} \min\limits_{\mathbf{U}\in\mathcal{N}_{ad}} J_1\left[\mathbf{Y}(\mathbf{U},\overline{\mathbf{V}})\right] \\ \min\limits_{\mathbf{V}\in\mathbb{R}^p} J_{1,2}\left[\mathbf{Y}(\overline{\mathbf{U}},\mathbf{V})\right] \end{cases} , \quad J_{1,2}(\mathbf{Y}) = \frac{J_1(\mathbf{Y})}{J_1(\mathbf{Y}_1^*)} + \zeta\left(\theta\frac{J_2(\mathbf{Y})}{J_2(\mathbf{Y}_1^*)} - \frac{J_1(\mathbf{Y}_1)}{J_1(\mathbf{Y}_1^*)}\right) \tag{A.6}$$

where $\mathcal{N}_{ad}$ is defined in equation (A.4), θ is a strictly-positive relaxation parameter ($\theta > 1$ for over-relaxation and $\theta < 1$ for under-relaxation) and $\overline{\mathbf{Y}} = \mathbf{Y}(\overline{\mathbf{U}},\overline{\mathbf{V}})$ is the corresponding Nash equilibrium point.
This allows to modify the optimal configuration for $J_1(\mathbf{Y})$ by gradually and smoothly introducing a competition with the secondary functional $J_{1,2}(\mathbf{Y})$ in a Nash game. For $\zeta = 0$ the original optimal solution $\mathbf{Y}_1^*$ is a Nash equilibrium point of the trivial game where only one objective functional is present. By continuity, the Nash equilibrium solution exists, at least for sufficiently small values of ζ.
A performance result
This procedure allows to identify an orthogonal decomposition of the parameter space such that the p tail elements of the basis correspond to the directions of least variation of the primary functional $J_1(\mathbf{Y})$ about its minimum under the active constraints. These eigenvectors span the subspace of dimension p in which the primary functional is the most insensitive to small variations of the design vector; thanks to this information, a second phase of optimization to reduce the secondary functional $J_2(\mathbf{Y})$ without excessively worsening the first one can be performed.

Competitive optimization for two-disciplines problems

Let us consider two objective functionals $J_1(\mathbf{Y})$ and $J_2(\mathbf{Y})$. The initial design point $\mathbf{Y}^{(0)}$ is Pareto-stationary that is

$$\alpha\nabla J_1(\mathbf{Y}^{(0)}) + (1-\alpha)\nabla J_2(\mathbf{Y}^{(0)}) = 0 \quad , \quad \alpha \in [0,1]$$

Three cases are possible:

- Pareto-stationarity of type I: $\nabla J_1(\mathbf{Y}^{(0)}) = \nabla J_2(\mathbf{Y}^{(0)}) = 0$;
- Pareto-stationarity of type II: $\nabla J_1(\mathbf{Y}^{(0)}) = 0$ and $\nabla J_2(\mathbf{Y}^{(0)}) \neq 0$ or vice versa;
- Pareto-stationarity of type III: $\nabla J_1(\mathbf{Y}^{(0)}) + \gamma\nabla J_2(\mathbf{Y}^{(0)}) = 0$, $\gamma = \frac{1-\alpha}{\alpha} > 0$.

We propose to examine the different scenarios arising after the algorithm achieves a Pareto-stationary point for (J_1, J_2). For the sake of simplicity, first we consider the case of convex functionals then we extend our observations to the non-convex case.

Convex case

Pareto-stationarity of type I

At $\mathbf{Y} = \mathbf{Y}^{(0)}$ both criteria achieve local minima $J_1(\mathbf{Y}^{(0)})$ and $J_2(\mathbf{Y}^{(0)})$ of their own. Hence the configuration cannot be improved and the optimization process is over.

Pareto-stationarity of type II

Since $\nabla J_1(\mathbf{Y}^{(0)}) = 0$, the functional $J_1(\mathbf{Y})$ has achieved a local minimum whereas $J_2(\mathbf{Y})$ can still be improved. The algorithm can either be stopped because the corresponding configuration is acceptable or proceed towards a competitive optimization phase. In this framework, a Nash equilibrium is sought starting from a hierarchical split of the variables in the orthogonal basis obtained by the eigenvectors of the Hessian matrix $H_1(\mathbf{Y}^{(0)})$.

Pareto-stationarity of type III

The achieved design point $\mathbf{Y}^{(0)}$ realizes a Pareto-optimal solution and in absence of additional criteria to improve the results, the optimization procedure is stopped.

If we relax the hypothesis on the local convexity of the functionals, the strategy is more delicate and we discuss different scenarios according to several assumptions that can be made on the eigenvalues of the Hessian matrices $H_1(\mathbf{Y}^{(0)})$ and $H_2(\mathbf{Y}^{(0)})$.

Non-convex case

Pareto-stationarity of type I

For $i = 1, 2$ let us consider the variations $\delta J_i(\mathbf{Y})$ of the criteria J_i's arising from a perturbation $\boldsymbol{\delta}\mathbf{Y}$ of the design vector $\mathbf{Y}$ about $\mathbf{Y}^{(0)}$.

$$\delta J_i(\mathbf{Y}^{(0)}) = J_i(\mathbf{Y}^{(0)} + \boldsymbol{\delta}\mathbf{Y}) - J_i(\mathbf{Y}^{(0)}) = \nabla J_i(\mathbf{Y}^{(0)}) \cdot \boldsymbol{\delta}\mathbf{Y} + \boldsymbol{\delta}\mathbf{Y} \cdot H_i(\mathbf{Y}^{(0)})\boldsymbol{\delta}\mathbf{Y} \quad , \quad i = 1, 2$$

Under the hypothesis of Pareto-stationarity of type I, the principal terms of the expansions δJ_i's are quadratic. Since we are dealing with the non-convex case, at least one of the Hessian matrices is not positive-definite.

If $H_1(\mathbf{Y}^{(0)})$ is positive-definite and only $H_2(\mathbf{Y}^{(0)})$ has some negative eigenvalues, J_1 has achieved a minimum whereas J_2 can still be improved. The optimization procedure can either stop or proceed to seek a Nash equilibrium by means of a hierarchical split of the variables based on the eigensystem of matrix $H_1(\mathbf{Y}^{(0)})$.

If both $H_1(\mathbf{Y}^{(0)})$ and $H_2(\mathbf{Y}^{(0)})$ have some negative eigenvalues, we define two families of linearly independent eigenvectors associated with these eigenvalues:

$$\mathcal{F}_1 = \{\mathbf{u}_1, \ldots, \mathbf{u}_p\} \qquad \mathcal{F}_2 = \{\mathbf{v}_1, \ldots, \mathbf{v}_q\} \tag{A.7}$$

Given the vector of coefficients $(\alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q)^T \neq \mathbf{0}$, if the family $\mathcal{F}_1 \cup \mathcal{F}_2$ is linearly dependent, that is

$$\sum_{i=1}^{p} \alpha_i \mathbf{u}_i - \sum_{j=1}^{q} \beta_j \mathbf{v}_j = \mathbf{0}$$

then a descent direction for both criteria exists and can be written as

$$\boldsymbol{\omega}_r = \sum_{i=1}^{p} \alpha_i \mathbf{u}_i = \sum_{j=1}^{q} \beta_j \mathbf{v}_j$$

Thanks to the linear independence of the families $\mathcal{F}_1$ and $\mathcal{F}_2$ separately, $\boldsymbol{\omega}_r$ is not equal to zero and an optimization step along this direction can be performed.

If the spaces spanned by the two families in equation (A.7) are independent, the global procedure can either stop or proceed by splitting the territories respectively related to $\mathcal{F}_1$ and $\mathcal{F}_2$ in order to seek a Nash equilibrium.

Pareto-stationarity of type II

In this scenario, only the gradient vector associated with one criterion is zero. If the Hessian matrix - namely $H_1(\mathbf{Y}^{(0)})$ - is positive-definite, $J_1(\mathbf{Y})$ has achieved a local minimum and again we propose a hierarchical split based on the structure of the eigenvectors of $H_1(\mathbf{Y}^{(0)})$ to seek a Nash equilibrium for the problem.

Otherwise, if $H_1(\mathbf{Y}^{(0)})$ has some negative eigenvalues we consider the family of associated eigenvectors

$$\mathcal{F}_1 = \{\mathbf{u}_1, \dots, \mathbf{u}_p\}$$

Under these hypotheses, if $\nabla J_2(\mathbf{Y}^{(0)})$ is not orthogonal to the space spanned by $\mathcal{F}_1$, a descent direction common to both criteria exists and can be used to improve all the objective functionals. If $\nabla J_2(\mathbf{Y}^{(0)}) \perp \langle \mathbf{u}_1, \dots, \mathbf{u}_p \rangle$, we propose to seek a Nash equilibrium using $\mathcal{F}_1$ as strategy for player one and the remaining eigenvectors of $H_1(\mathbf{Y}^{(0)})$ as strategy for the other player.

Pareto-stationarity of type III

Let us consider the direction defined as follows

$$\mathbf{u}_{1,2} = \frac{\nabla J_1(\mathbf{Y}^{(0)})}{\|\nabla J_1(\mathbf{Y}^{(0)})\|} = -\frac{\nabla J_2(\mathbf{Y}^{(0)})}{\|\nabla J_2(\mathbf{Y}^{(0)})\|} \tag{A.8}$$

Along $\mathbf{u}_{1,2}$, the two criteria vary in opposite ways and no rational decision can be made in the absence of other information. We analyze the feasible movements within the hyperplane orthogonal to $\mathbf{u}_{1,2}$ and we define a projection matrix

$$P_{1,2} = I - [\mathbf{u}_{1,2}][\mathbf{u}_{1,2}]^T$$

Thus the reduced Hessian matrices are given by

$$H_1'(\mathbf{Y}^{(0)}) = P_{1,2}H_1(\mathbf{Y}^{(0)})P_{1,2} \qquad H_2'(\mathbf{Y}^{(0)}) = P_{1,2}H_2(\mathbf{Y}^{(0)})P_{1,2} \tag{A.9}$$

and by orthogonality to the gradient vectors this case can be analyzed within the framework of Pareto-stationary points of type I in a subspace of dimension $N-1$.

Competitive optimization for multi-disciplines problems

Eventually we consider the case of n different criteria $J_i(\mathbf{Y})$, $i = 1, \dots, n$. In order to mantain the Pareto-stationarity condition as much as possible we define the following functionals

$$J_A(\mathbf{Y}) = \sum_{\substack{i=1 \\ i\neq k}}^{n} \alpha_i J_i(\mathbf{Y}) \qquad J_B(\mathbf{Y}) = J_k(\mathbf{Y}) \tag{A.10}$$

Under this assumption, we restrict our original problem to a two disciplines case where $\nabla J_A(\mathbf{Y}^{(0)}) = 0$. We notice that the choice of k is critical for the formulation of the problem and may be strongly influenced by the designer's bias to improve a specific criterion. In general, we propose to choose the index k in order to maximize the orthogonal projection of the gradient $\nabla J_B(\mathbf{Y}^{(0)})$ onto the subspace assigned to the second virtual player.

From the previous section, we know that this subspace is entirely defined by the diagonalization of the reduced Hessian matrix $H_1'(\mathbf{Y}^{(0)})$ as in (A.9). Thus for each fixed k we get

$$\mathbf{u}_k^{(0)} = \nabla J_k(\mathbf{Y}^{(0)}) = \sum_{i=1}^{N} \beta_i^{(0)} \boldsymbol{\omega}_i$$

where $\beta_i^{(0)} = (\mathbf{u}_k^{(0)}, \boldsymbol{\omega}_i)$ by orthogonality. Since the subspace assigend to the second player is described by the tail elements of the vector $\boldsymbol{\omega}$, the index k is chosen in order to maximize the following quantity

$$\mathcal{R}(\boldsymbol{\omega}) = \sqrt{\sum_{i=N-p+1}^{N} (\beta_i^0)^2}$$

By means of this procedure, we approximately mantain the Pareto-optimality of the solution while maximizing the potential payoff that can be achieved by a sequence of competitive optimization problems where the improvement of the criterion $J_k(\mathbf{Y})$ is sought alone.

A general introduction to Nash games is available in [Nas51]. For a detailed treatment of optimization problems by means of competitive strategies we refer to [AEM07], [Dés07] and [Dés11].

Appendix B

Proofs of the main results

Here we propose an overview of the proofs of the major results about *Multiple-Gradient Descent Algorithm* as in the works of J.-A. Désidéri. We refer to [Dés12c] for the original proofs by the author.

Relationship between Pareto-optimality and Pareto-stationarity

Proof. (Proposition 2.2) Let r be the rank of the familiy of gradient vectors $\{\mathbf{u}_i\}_{i=1}^n$ where $\mathbf{u}_i = \nabla J_i(\mathbf{Y}^{(0)})$:

$$r = rank\{\mathbf{u}_i\}_{i=1}^n = dim\langle \mathbf{u}_1, \dots, \mathbf{u}_n \rangle$$

We analyze the scenarios arising from different values of r.
If $r = 0$ all the gradient vectors are equal to zero thus a convex combination equal to zero exists and the result is trivial.
If $r = 1$, the gradient vectors are colinear, that is $\mathbf{u}_i = \beta_i \mathbf{u}$ where $\mathbf{u}$ denotes a unit vector; the β_i's are the coefficients of the convex combination and they are not all equal to zero. We consider a perturbation $\delta\mathbf{Y} = -\epsilon\mathbf{u}$ in a working ball centered in $\mathbf{Y}^{(0)}$ and we compute the corresponding variations over the functionals J_i's:

$$J_i(\mathbf{Y}^{(0)} + \delta\mathbf{Y}) = J_i(\mathbf{Y}^{(0)}) + \nabla J_i(\mathbf{Y}^{(0)}) \cdot \delta\mathbf{Y} + \mathcal{O}((\delta\mathbf{Y})^2)$$

Thus given a sufficiently small value of the parameter $\epsilon > 0$, the perturbation of the functional J_i reads as

$$\begin{aligned} \delta J_i = J_i(\mathbf{Y}^{(0)} + \delta\mathbf{Y}) - J_i(\mathbf{Y}^{(0)}) = \nabla J_i(\mathbf{Y}^{(0)}) \cdot \delta\mathbf{Y} + \mathcal{O}((\delta\mathbf{Y})^2) = \\ = \beta_i \mathbf{u} \cdot (-\epsilon\mathbf{u}) + \mathcal{O}(\epsilon^2 \mathbf{u} \cdot \mathbf{u}) = -\beta_i\epsilon + \mathcal{O}(\epsilon^2) \end{aligned}$$

Let us assume that all coefficients β_i's are of the same sign. On one hand, if $\beta_i > 0\ \forall i = 1, \dots, n$, at least one criterion would diminish whereas the other ones would remain unchanged to $\mathcal{O}(\epsilon^2)$ but this is a contradiction since $\mathbf{Y}^{(0)}$ is supposed to be Pareto-optimal. On the other hand, if $\beta_i < 0\ \forall i = 1, \dots, n$, at least one criterion increases whereas the others do not and again this scenario is not possible due to the hypothesis of Pareto-optimality of $\mathbf{Y}^{(0)}$. Hence, both positive and negative coefficients exist and zero coefficients are admissible too.
Let β_1 and β_2 be such that $\beta_1\beta_2 < 0$; we can compute the coefficients for the convex combination in equation (2.2) as follows

$$\alpha_1 = -\frac{\beta_2}{\beta_1 - \beta_2} \quad , \quad \alpha_2 = \frac{\beta_1}{\beta_1 - \beta_2} \tag{B.1}$$

and we notice that the coefficients in equation (B.1) fulfill the requirements of positivity and partition of the unit, that is

$$\alpha_1 > 0 \quad , \quad \alpha_2 > 0 \quad , \quad \alpha_1 + \alpha_2 = 1$$

Under previous assumptions we get

$$\alpha_1 \mathbf{u}_1 + \alpha_2 \mathbf{u}_2 = -\frac{\beta_2}{\beta_1 - \beta_2}\beta_1 \mathbf{u} + \frac{\beta_1}{\beta_1 - \beta_2}\beta_2 \mathbf{u} = 0$$

thus the Pareto-stationarity is established for $r = 1$.
Now we consider the general case where $r \in [2, n-1]$. We can find a convex combination of $r+1$ gradient vectors that is equal to zero and by means of a permutation of the indices we can write it as the combination of the first $r+1$ elements:

$$\mathbf{u}_1 + \sum_{k=2}^{r+1} \mu_k \mathbf{u}_k = 0 \tag{B.2}$$

We claim that $\mu_k \geq 0 \; \forall k \geq 2$. To prove that, let us assume instead that $\mu_2 < 0$ and define V as the span identified by the remaining $r-1$ gradient vectors as follows:

$$V = \langle \mathbf{u}_3, \ldots, \mathbf{u}_{r+1} \rangle$$

Hence $dimV \leq r-1 \leq n-2 \leq N-2$ and $dimV^{\perp} \geq 2$.
Let $\boldsymbol{\omega}$ be an arbitrary element of $V^{\perp}$ and consider the scalar product of (B.2) with $\boldsymbol{\omega}$:

$$\left(\mathbf{u}_1 + \sum_{k=2}^{r+1} \mu_k \mathbf{u}_k \ , \ \boldsymbol{\omega} \right) = \Big(\mathbf{u}_1 \ , \ \boldsymbol{\omega} \Big) + \Big(\mu_2 \mathbf{u}_2 \ , \ \boldsymbol{\omega} \Big) = 0$$

that yields to the following relationship among the Fréchet derivatives

$$dJ_1(\Omega; \boldsymbol{\omega}) = -\mu_2 dJ_2(\Omega; \boldsymbol{\omega}) \qquad \forall \boldsymbol{\omega} \in V^{\perp}$$

If we obtain the identity $0 = -\mu_2 \cdot 0 \quad \forall \boldsymbol{\omega} \in V^{\perp}$, the gradient vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ both belong to V and consequently the whole family $\{\mathbf{u}_i\}_{i=1}^{n}$ belongs to V as well since by assumption its rank is r. However this is not possible because $dimV \leq r-1 < r$, thus there exists an element $\boldsymbol{\omega} \in V^{\perp}$ such that

$$dJ_1(\Omega; \boldsymbol{\omega}) = -\mu_2 dJ_2(\Omega; \boldsymbol{\omega}) \neq 0$$

Let us suppose that $dJ_1(\Omega; \boldsymbol{\omega}) > 0$. Then an infinitesimal perturbation of the design point $\mathbf{Y}$ in the direction $-\boldsymbol{\omega}$ causes a reduction of both criteria $J_1(\mathbf{Y})$ and $J_2(\mathbf{Y})$ and leaves the others unchanged to second order. Hence the initial design point $\mathbf{Y}^{(0)}$ is not Pareto-optimal which is in contradiction with our hypothesis. Hence $\mu_2 \geq 0$ and by performing a similar procedure we can prove that all the μ_k's have to be non-negative $\forall k \geq 2$.
By setting $\mu_1 = 1$ and $\alpha_i = \frac{\mu_i}{\sum_{k=1}^{n} \mu_k} \quad \forall i = 1, \ldots, n$ we get the relationship we were seeking for $r \in [2, n-1]$:

$$\sum_{i=1}^{n} \alpha_i \mathbf{u}_i = \mathbf{0} \quad , \quad \alpha_i \geq 0 \;\; \forall i = 1, \ldots, n \quad \sum_{i=1}^{n} \alpha_i = 1$$

Eventually we analyze the case where $r = n$, that is the gradient vectors are independent. In this scenario the Pareto-optimal design point $\mathbf{Y}^{(0)}$ is the solution of a family of optimization problems where we minimize a given functional $J_i(\mathbf{Y})$ within a region of admissible perturbations for the other J_k's $k \neq i$:

$$\min_{\mathbf{Y}\in\mathcal{U}_i^{ad}} J_i(\mathbf{Y}) \quad , \quad \mathcal{U}_i^{ad} = \left\{ \mathbf{Y} \in \mathbb{R}^N \;\middle|\; g_k(\mathbf{Y}) = J_k(\mathbf{Y}) - J_k(\mathbf{Y}^{(0)}) \leq 0 \quad \forall k = 1, \dots, n \quad k \neq i \right\} \tag{B.3}$$

In details, the minimization problem subject to inequality constraints in equation (B.3) stands for all the functionals. Thus the Lagrangian functional associated with the problem (B.3) reads $\forall i = 1, \dots, n$

$$L_i(\mathbf{Y}, \boldsymbol{\lambda}) = J_i(\mathbf{Y}) + \sum_{\substack{k=1\\k\neq i}}^{n} \lambda_k g_k(\mathbf{Y})$$

where $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n)^T$ are the non-negative Lagrange multipliers associated with the constraints g_k's. Since $\mathbf{Y}^{(0)}$ is the solution of the previously described optimization problem, it is a stationary point for $L_i(\mathbf{Y}, \boldsymbol{\lambda})$ $\forall i = 1, \dots, n$:

$$0 = \frac{\partial L_i}{\partial \mathbf{Y}}(\mathbf{Y}^{(0)}, \boldsymbol{\lambda}) = \mathbf{u}_i + \sum_{\substack{k=1\\k\neq i}}^{n} \lambda_k \mathbf{u}_k$$

This result is in contradiction with the hypothesis $r = n$ over the rank of the family of gradient vectors $\mathbf{u}_i$, $i = 1, \dots, n$ which has to be rejected. Therefore $r \leq n-1$ and all the possible cases have been examinated. Hence we evince that a Pareto-optimal design point is Pareto-stationary. □

Existence and uniqueness of the minimal-norm element $\boldsymbol{\omega} \in \overline{\mathcal{U}}$

Proof. (Proposition 2.3) As defined in equation (2.4), $\overline{\mathcal{U}}$ is a closed set, closure of its interior $\mathcal{U}$ which is made of the elements of $\overline{\mathcal{U}}$ associated to the strictly-positive coefficients $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^T$. Hence the norm admits at least one realization of a minimum in $\overline{\mathcal{U}}$ and we define $\boldsymbol{\omega}$ the minimal-norm element in the convex hull (2.4):

$$\boldsymbol{\omega} = \underset{\mathbf{u} \,\in\, \overline{\mathcal{U}}}{\operatorname{argmin}} \; \|\mathbf{u}\| \tag{B.4}$$

We established the existence of a minimal-norm element in $\overline{\mathcal{U}}$ and we suppose that there are two realizations $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ such that $\|\boldsymbol{\omega}_1\| = \|\boldsymbol{\omega}_2\|$. Since $\overline{\mathcal{U}}$ is convex, $\forall \epsilon \in [0, 1]$ the element $\mathbf{z} = (1-\epsilon)\boldsymbol{\omega}_1 + \epsilon\boldsymbol{\omega}_2$ belongs to $\overline{\mathcal{U}}$. We can rewrite $\mathbf{z}$ as follows

$$\mathbf{z} = \boldsymbol{\omega}_1 + \epsilon(\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) = \boldsymbol{\omega}_1 + \epsilon\boldsymbol{\omega}_{2,1} \quad , \quad \boldsymbol{\omega}_{2,1} = \boldsymbol{\omega}_2 - \boldsymbol{\omega}_1$$

Since $\mathbf{z} \in \overline{\mathcal{U}}$ and $\boldsymbol{\omega}_1$ is the minimal-norm element in the convex hull, we have

$$(\boldsymbol{\omega}_1 + \epsilon\boldsymbol{\omega}_{2,1}, \boldsymbol{\omega}_1 + \epsilon\boldsymbol{\omega}_{2,1}) = \|\mathbf{z}\|^2 \geq \|\boldsymbol{\omega}_1\|^2 = (\boldsymbol{\omega}_1, \boldsymbol{\omega}_1)$$

thus we obtain

$$2\epsilon(\boldsymbol{\omega}_1, \boldsymbol{\omega}_{2,1}) + \epsilon^2(\boldsymbol{\omega}_{2,1}, \boldsymbol{\omega}_{2,1}) \geq 0 \tag{B.5}$$

We notice that the inequality (B.5) has to stand even for small values of ϵ when the first term is dominant, so we get the following condition

$$(\boldsymbol{\omega}_1, \boldsymbol{\omega}_{2,1}) \geq 0$$

If we choose $\epsilon = 1$, we get $\mathbf{z} = \boldsymbol{\omega}_2$ thus $\|\boldsymbol{\omega}_2\|^2 \geq \|\boldsymbol{\omega}_1\|^2$; moreover the inequality (B.5) is strict unless $\boldsymbol{\omega}_{2,1} = 0$. From the initial hypothesis on $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ we know that the equality between the norms should hold; thus the only possibility is that $\boldsymbol{\omega}_{2,1} = 0$ and the uniqueness of the minimal-norm element in $\overline{\mathcal{U}}$ is established. □

Proof. (Proposition 2.4) Let $\mathbf{v} = \mathbf{u} - \boldsymbol{\omega}$. Since $\overline{\mathcal{U}}$ is convex, by using the hypothesis $\mathbf{u} \in \overline{\mathcal{U}}$ and the definition of $\boldsymbol{\omega}$ as in (B.4) we deduce that $\mathbf{v}$ belongs to $\overline{\mathcal{U}}$ too.
Moreover

$$\forall \epsilon \in [0,1] \quad \boldsymbol{\omega} + \epsilon \mathbf{v} \in \overline{\mathcal{U}}$$

and since $\boldsymbol{\omega}$ is the minimal-norm element in $\overline{\mathcal{U}}$ we get

$$\|\boldsymbol{\omega}\|^2 + 2\epsilon(\boldsymbol{\omega}\ ,\ \mathbf{v}) + \epsilon^2\|\mathbf{v}\|^2 = (\boldsymbol{\omega} + \epsilon\mathbf{v}\ ,\ \boldsymbol{\omega} + \epsilon\mathbf{v}) \geq (\boldsymbol{\omega}\ ,\ \boldsymbol{\omega})$$

Thus $2\epsilon(\boldsymbol{\omega}\ ,\ \mathbf{v}) + \epsilon^2\|\mathbf{v}\|^2 \geq 0$ and this result has to hold even for sufficiently small values of the parameter ϵ, that is $(\boldsymbol{\omega}\ ,\ \mathbf{v}) \geq 0$. From the definition of $\mathbf{v}$ we get

$$(\boldsymbol{\omega}\ ,\ \mathbf{u} - \boldsymbol{\omega}) \geq 0 \quad \Leftrightarrow \quad (\mathbf{u}\ ,\ \boldsymbol{\omega}) \geq \|\boldsymbol{\omega}\|^2$$

Given the arbitrary of $\mathbf{u} \in \overline{\mathcal{U}}$ the result follows straightforward. □

Establishment of Multiple-Gradient Descent Algorithm

Proof. (Theorem 2.5) Let us consider a generic design point $\mathbf{Y}^{(0)}$ and the convex hull $\overline{\mathcal{U}}$. Proposition 2.3 states that the minimal-norm element exists and is unique. Thus we can update our initial design point by computing $\mathbf{Y} = \mathbf{Y}^{(0)} - \widetilde{\rho}\boldsymbol{\omega}$, where $\widetilde{\rho}$ is a properly determined step size that assures that all the criteria improve.
Let $\mathbf{u}_i = \nabla J_i(\mathbf{Y}^{(0)})$; if $\boldsymbol{\omega} = \mathbf{0}$, from the definition of $\overline{\mathcal{U}}$ we get

$$\sum_{i=1}^{n} \alpha_i \mathbf{u}_i = \mathbf{0} \quad , \quad \alpha_i \geq 0 \quad \forall i = 1, \ldots, n \quad , \quad \sum_{i=1}^{n} \alpha_i = 1$$

that is, $\mathbf{Y} = \mathbf{Y}^{(0)}$ is Pareto-stationary.
Otherwise, computing the Fréchet derivatives of the functionals $J_i\ ,\ i = 1, \ldots, n$ along the direction $-\boldsymbol{\omega}$ we get

$$dJ_i(\mathbf{Y}^{(0)}; -\boldsymbol{\omega}) = \nabla J_i(\mathbf{Y}^{(0)}) \cdot (-\boldsymbol{\omega}) = (\mathbf{u}_i\ ,\ -\boldsymbol{\omega}) = -(\mathbf{u}_i\ ,\ \boldsymbol{\omega}) \leq 0$$

where the last inequality holds thanks to proposition 2.4. Since this is true $\forall i = 1, \ldots, n$, the vector $-\boldsymbol{\omega}$ defines a descent direction common to all criteria.
Now we consider the case of $\boldsymbol{\omega} \in \mathcal{U}$, that is none of the inequality constraints is saturated ($\forall i = 1, \ldots, n \quad \alpha_i > 0$). Under these assumptions, the element $\boldsymbol{\omega}$ has the following form

$$\boldsymbol{\omega} = \mathbf{u} = \sum_{i=1}^{n} \alpha_i \mathbf{u}_i$$

where the vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^T \in \mathbb{R}^n$ is the solution of the finite-dimensional minimization problem

$$\boldsymbol{\alpha} = \underset{\mathbf{u} \in \mathcal{U}}{\operatorname{argmin}}\, p(\mathbf{u}) \quad , \quad p(\mathbf{u}) = (\mathbf{u} \ , \ \mathbf{u})$$

subject to the following constraint:

$$\sum_{i=1}^{n} \alpha_i = 1$$

Thus the Lagrangian functional to take into account the constraint in the original problem writes as

$$L(\boldsymbol{\alpha}, \lambda) = p(\mathbf{u}) + \lambda\left(\sum_{i=1}^{n} \alpha_i - 1\right)$$

where $\boldsymbol{\alpha}$ is the optimization variable and λ is a positive Lagrange multiplier. The first-order optimality conditions read as

$$\begin{cases} \dfrac{\partial}{\partial \lambda} L(\boldsymbol{\alpha}, \lambda) = 0 \\ \dfrac{\partial}{\partial \alpha_i} L(\boldsymbol{\alpha}, \lambda) = 0 \quad , \quad \forall i = 1, \ldots, n \end{cases}$$

The second line implies $\forall i = 1, \ldots, n$

$$\frac{\partial}{\partial \alpha_i} p(\mathbf{u}) + \lambda = 0$$

Computing the derivative of $p(\mathbf{u})$ with respect to a given α_i we get:

$$\frac{\partial}{\partial \alpha_i} p(\mathbf{u}) = \frac{\partial}{\partial \alpha_i} (\mathbf{u} \ , \ \mathbf{u}) = 2\left(\frac{\partial \mathbf{u}}{\partial \alpha_i} \ , \ \mathbf{u}\right) = 2\left(\frac{\partial}{\partial \alpha_i}\left[\sum_{j=1}^{n} \alpha_j \mathbf{u}_j\right] \ , \ \mathbf{u}\right) = 2(\mathbf{u}_i \ , \ \boldsymbol{\omega})$$

thus the Fréchet derivatives $(\mathbf{u}_i \ , \ \boldsymbol{\omega})$ are equal $\forall i = 1, \ldots, n$ and independently of i it holds

$$2(\mathbf{u}_i \ , \ \boldsymbol{\omega}) = -\lambda \Leftrightarrow (\mathbf{u}_i \ , \ \boldsymbol{\omega}) = -\frac{\lambda}{2}$$

Eventually, any arbitrary element $\mathbf{u} \in \overline{\mathcal{U}}$ can be written in the following form

$$\mathbf{u} = \sum_{i=1}^{n} \mu_i \mathbf{u}_i \quad , \quad \mu_i \geq 0 \quad \forall i = 1, \ldots, n \quad \sum_{i=1}^{n} \mu_i = 1$$

Since the coefficients $\mu_i \ , \ i = 1, \ldots, n$ constitute a partition of the unity, the scalar product equals are equal to a constant value

$$(\mathbf{u} \ , \ \boldsymbol{\omega}) = \left(\sum_{i=1}^{n} \mu_i \mathbf{u}_i \ , \ \boldsymbol{\omega}\right) = \sum_{i=1}^{n} \mu_i (\mathbf{u}_i \ , \ \boldsymbol{\omega}) = (\mathbf{u}_i \ , \ \boldsymbol{\omega}) = -\frac{\lambda}{2}$$

By setting $\mathbf{u} = \boldsymbol{\omega}$ we deduce that $-\lambda/2$ is exactly the value of $\|\boldsymbol{\omega}\|^2$ which is our thesis. □

Convergence of Multiple-Gradient Descent Algorithm iterates

Proof. (Theorem 2.6) Let us assume that the objective functionals $J_i(\mathbf{Y})$, $i = 1, \dots, n$ are strictly positive and are infinite whenever $\|\mathbf{Y}\| \to \infty$.
The sequence of values defined by *MGDA* for any criterion $\{J_i(\mathbf{Y}^{(k)})\}_{k=1}^{\infty}$ is positive and monotone-decreasing, thus it is bounded. Moreover $\forall i = 1, \dots, n$

$$J_i(\mathbf{Y}) \to \infty \quad , \quad \|\mathbf{Y}\| \to \infty$$

thus the sequence of iterates $\{\mathbf{Y}^{(k)}\}_{k=1}^{\infty}$ is bounded too.
This implies the existence of a subsequence $\{\mathbf{Y}_j^{(k)}\}$ that converges to a limiting design-point $\mathbf{Y}^*$. In particular, $\mathbf{Y}^*$ is necessarily Pareto-stationary; otherwise, if $\boldsymbol{\omega} \neq \mathbf{0}$ a new iteration of *MGDA* could be performed improving all criteria of a finite amount and a better design point would be determined.

Now we consider the scenario in which the assumption on the strictly positivity of the J_i's $\forall i = 1, \dots, n$ does not hold. We prove that a scaling procedure of the functionals is possible without loss of generality. Let $\mathcal{B}_R = \mathcal{B}(\mathbf{Y}^{(0)}, R)$ an open ball centered in $\mathbf{Y}^{(0)}$ with radius R; by assuming $J_i(\mathbf{Y}) \in \mathcal{C}^2(\Omega)$ $\forall i = 1, \dots, n$ we propose the following substitution based on the Hessian computation

$$\overline{J}_i(\mathbf{Y}) = exp\left\{ \alpha_i \frac{\|H_i(\mathbf{Y}^{(0)})\|}{\|\nabla J_i(\mathbf{Y}^{(0)})\|^2} (J_i(\mathbf{Y}) - J_i(\mathbf{Y}^{(0)})) \right\} \tag{B.6}$$

where the α_i's are dimensionless constants, ∇J_i's represent as usual the gradient vectors of the functionals and H_i's the Hessian matrices. $\|H_i(\mathbf{Y}^{(0)})\|$ is the Frobenius norm of the Hessian matrix, that is

$$\|H_i(\mathbf{Y}^{(0)})\| = \sqrt{tr[H_i(\mathbf{Y}^{(0)})^T H_i(\mathbf{Y}^{(0)})]}$$

New criteria (B.6) are dimensionless and vary as the original ones. In particular $\forall i = 1, \dots, n$ we choose the constants α_i's such that

$$\alpha_i \frac{\|H_i(\mathbf{Y}^{(0)})\|}{\|\nabla J_i(\mathbf{Y}^{(0)})\|} = \frac{\gamma}{R} \sim 1$$

and we get

$$\overline{J}_i(\mathbf{Y}^{(0)}) = 1 \qquad \nabla J_i(\mathbf{Y}^{(0)}) = \frac{\gamma}{R}$$

As previously pointed out, it is important to assure the value of the functionals to be infinite whenever $\|\mathbf{Y}\|$ is infinite. Thus we introduce the function $\Phi(\mathbf{x}) : \mathbb{R} \to \mathbb{R}$ such that $\Phi \in \mathcal{C}^{\infty}(\mathbb{R})$ and $\Phi(\mathbf{x}) \sim \mathbf{x}$, $\mathbf{x} \to +\infty$:

$$\Phi(\mathbf{x}) = \begin{cases} 0 & , \ \mathbf{x} \leq \mathbf{0} \\ \mathbf{x} exp\left\{ -\frac{1}{\mathbf{x}^2} \right\} & , \ \mathbf{x} > \mathbf{0} \end{cases}$$

By introducing a new strictly positive constant ϵ, we can reformulate (B.6) as

$$\overline{\overline{J}}_i(\mathbf{Y}) = \overline{J}_i(\mathbf{Y}) + \epsilon \Phi\left(\frac{\|\mathbf{Y} - \mathbf{Y}^{(0)}\|^2}{R^2} - 1 \right) \tag{B.7}$$

We notice that inside the open ball $\mathcal{B}_R$, $\overline{\overline{J}}_i(\mathbf{Y})$ is identical to $\overline{J}_i(\mathbf{Y})$ and it grows like $\|\mathbf{Y}\|^2$ outside. Moreover, at the boundary of the open ball, the matching of the new functionals with the original ones is infinitely smooth and we get

$$\lim_{\|\mathbf{Y}\|\to\infty} \overline{\overline{J}}_i(\mathbf{Y}) = \infty$$

By substituting $\overline{\overline{J}}_i(\mathbf{Y})$ to $J_i(\mathbf{Y})$, we restrict ourselves to a case where the assumptions of the first part of the proof are satisfied. Thus all scenarios are considered and this concludes the proof. □

Remark B.1. Most of the results presented so far can be extended to the infinite-dimensional case. Here we highlight the similarities and the differences in the major steps of the strategy previously described for the finite-dimensional case:

- The functional setting for the problem is a Hilbert space H, possibly a subspace of L^2.
- The concept of gradient operator extends the notion of gradient vector to the infinite-dimensional case.
- Definition 2.1 holds for the infinite-dimensional case and the result of proposition 2.2 is straightforward.
- In proposition 2.4, the set defined by the convex combination of the gradient operators is no more closed when dealing with infinite dimension but the extension is straightforward if the minimal norm element is sought in the convex hull of the family of $\mathcal{U}$'s.
- Theorem 2.5 that establishes *MGDA* procedure holds also in the infinite-dimensional case.
- Some issues appear in the convergence result 2.6 that is extended to the infinite-dimensional case by theorem B.2.

Theorem B.2. *If the sequence of iterates* $\{\mathbf{Y}^{(k)}\}$ *generated by* Multiple-Gradient Descent Algorithm *is infinite, it admits a weakly convergent subsequence. We conjecture that the limit* $\mathbf{Y}^*$ *is a Pareto-stationary design-point.*

Remark B.3. When dealing with *MGDA* variants that use the information carried by the Hessian operator (Section 2.3.3), additional issues may arise in the infinite-dimensional case. For instance, an *a priori* analysis of the regularity of the operators has to be performed in order to identify specific conditions on the Hessian operators to assure the existence of the Hilbert-Schmidt norm, which is an extension of the Frobenious norm to the infinite-dimensional case.

Bibliography

[AC02] J.J. Alonso and H.-S. Chung. Using gradients to construct cokriging approximation models for high-dimensional design optimization problems. In *40th AIAA Aerospace Sciences Meeting and Exhibit*, pages 2002–0317, 2002.

[AEM07] B. Abou El Majd. *Algorithmes hiérarchiques et stratégies de jeux pour l'optimisation multidisciplinaire Application à l'optimisation de la voilure d'un avion d'affaires.* Ph.D. Thesis, Université de Nice Sophia-Antipolis, 2007.

[AJT04] G. Allaire, F. Jouve, and A.-M. Toader. Structural optimization using sensitivity analysis and a level-set method. *J. Comput. Phys.*, 194(1):363–393, 2004.

[AL99] N.M. Alexandrov and R.M. Lewis. Comparative properties of collaborative optimization and other approaches to MDO. Research Report 99-24, ICASE - NASA Langley Research Center, 1999.

[All06] G. Allaire. *Conception optimale de structures.* Springer, 2006.

[BB05] D. Bucur and G. Buttazzo. *Variational Methods in Shape Optimization Problems.* Progr. Nonlinear Differential Equations Appl. Birkhäuser Boston, 2005.

[BBdVC$^+$06] Y. Basilevs, L. Beirao da Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric analysis : Approximation, stability and error estimates for h-refined meshes. *Math. Mod. Meth. Appl. Sci.*, 16(7):1031 – 1090, 2006.

[BDVS12] L. Blanchard, R. Duvigneau, A.-V. Vuong, and B. Simeon. Shape Gradient Computation in Isogeometric Analysis for Linear Elasticity. Research Report RR-8111, INRIA, 2012.

[BF10] S. Bruvollm and M.S. Floater. Transfinite mean value interpolation in general dimension. *J. Comp. Appl. Math.*, 233(7):1631 – 1639, 2010.

[BPV97] J.H. Bramble, J.E. Pasciak, and A.T. Vassilev. Analysis of the inexact Uzawa Algorithm for saddle point problems. *SIAM J. Numer. Anal*, 34:1072–1092, 1997.

[Cea86] J. Cea. Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût. *ESAIM: Mathematical Modelling and Numerical Analysis*, 20(3):371–402, 1986.

[CH09] S. Cho and S.-H. Ha. Isogeometric shape design optimization: exact geometry and enhanced sensitivity. *Structural and Multidisciplinary Optimization*, 38:53–70, 2009.

[Cha03] A. Chambolle. A density result in two-dimensional linearized elasticity, and applications. *Arch. Ration. Mech. Anal.*, 167(3):211–233, 2003.

[CHB09] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009.

[Che75] D. Chenais. On the existence of a solution in a domain identification problem. *J. Math. Anal. Appl.*, 5:189–289, 1975.

[CHR07] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Comp. Meth. Appl. Mech. Eng.*, 196(41-44):4160 – 4183, 2007.

[Cre93] N.A.C. Cressie. *Statistics for Spatial Data*. Wiley, Revised edition, 1993.

[dB72] C. de Boor. On calculating with B-Splines. *J. Approximation Theory*, 6:50–62, 1972.

[DC12] R. Duvigneau and P. Chandrashekar. Kriging-based optimization applied to flow control. *Int. J. Numer. Meth. Fluids*, 69(11):1701–1714, 2012.

[Dés07] J.-A. Désidéri. Split of Territories in Concurrent Optimization. Research Report RR-6108, INRIA, 2007.

[Dés09] J.-A. Désidéri. Multiple-Gradient Descent Algorithm (MGDA). Research Report RR-6953, INRIA, 2009.

[Dés11] J.-A. Désidéri. Cooperation and competition in multidisciplinary optimization - Application to the aero-structural aircraft wing shape optimization. *Comp. Opt. and Appl.*, 2011.

[Dés12a] J.-A. Désidéri. MGDA II: A direct method for calculating a descent direction common to several criteria. Research Report RR-7922, INRIA, 2012.

[Dés12b] J.-A. Désidéri. MGDA variants for multi-objective optimization. Research Report RR-8068, INRIA, 2012.

[Dés12c] J.-A. Désidéri. Multiple-Gradient Descent Algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313 – 318, 2012.

[dGAJ08] F. de Gournay, G. Allaire, and F. Jouve. Shape and topology optimization of the robust compliance via the level set method. *ESAIM: Control, Optimisation and Calculus of Variations*, 14:43–70, 0 2008.

[DJS10] M.R. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with T-splines. *Comp. Meth. Appl. Mech. Eng.*, 199(5-8):264 – 275, 2010.

[DPAM00] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast elitist multi-objective genetic algorithm: NSGA-II. *IEEE T. Evolut. Comput.*, 6:182–197, 2000.

[DZ01] M.C. Delfour and J.-P. Zolésio. *Shapes and geometries: analysis, differential calculus, and optimization.* SIAM, Philadelphia, USA, 2001.

[Eva10] L.C. Evans. *Partial differential equations.* AMS, 2010.

[FF98] C.M. Fonseca and P.J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. II. application example. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, pages 38–47, 1998.

[HCB05] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Comp. Meth. Appl. Mech. Eng.*, 194(39-41):4135 – 4195, 2005.

[HM03] J. Haslinger and R.A.E. Mäkinen. *Introduction to Shape Optimization: Theory, Approximation, and Computation.* SIAM, Philadelphia, USA, 2003.

[Hor95] C. O. Horgan. Korn's inequalities and their applications in continuum mechanics. *SIAM Review*, 37(4):pp. 491–511, 1995.

[HP05] A. Henrot and M. Pierre. *Variation et optimisation de formes: Une analyse géométrique.* Springer, 2005.

[HRS10] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for nurbs-based isogeometric analysis. *Comp. Meth. Appl. Mech. Eng.*, 199(5–):301 – 313, 2010.

[HTM11] B. Hassani, S.M. Tavakkoli, and N.Z. Moghadam. Application of isogeometric analysis in structural shape optimization. *Scientia Iranica*, 18(4):846 – 852, 2011.

[KC00] J.D. Knowles and D.W. Corne. Approximating the nondominated front using the Pareto Archived Evolution Strategy. *Evol. Comput.*, 8(2):149–172, 2000.

[KFBY03] P. Kagan, A. Fischer, and P.Z. Bar-Yoseph. Mechanically based models: Adaptive refinement for b-spline finite element. *Int. J. Numer. Meth. Eng.*, 57(8):1145–1175, 2003.

[Kir81] U. Kirsch. *Optimum structural design: concepts, methods, and applications.* McGraw-Hill, 1981.

[KN05] A.J. Keane and P.B. Nair. *Computational approaches for aerospace design: the pursuit of excellence.* Wiley, 2005.

[Mie99] K. Miettinen. *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers Dordrecht, Netherlands, 1999.

[MS76] F. Murat and J. Simon. Sur le contrôle par un domaine géométrique. Research Report 76 015, Laboratoire d'Analyse Numérique de l'Université Paris 6, 1976.

[Mye82] D. Myers. Matrix Formulation of Co-Kriging. *Mathematical Geology*, 14(3), 1982.

[Nas51] J.F. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[NW99] J.A. Nocedal and S.J. Wright. *Numerical optimization.* Springer-Verlag New York, USA, 1999.

[PAC$^+$10] Y. Parte, D. Auroux, J. Clément, M. Masmoudi, and J. Hermetz. *Collaborative optimization.* Wiley-ISTE, 2010.

[Pir84] O. Pironneau. *Optimal Shape Design for Elliptic Systems.* Springer-Verlag, 1984.

[PT97] L. Piegl and W. Tiller. *The NURBS book.* Springer-Verlag New York, USA, 1997.

[PT98] L.A. Piegl and W. Tiller. Computing the derivative of NURBS with respect to a knot. *Computer Aided Geometric Design*, 15(9):925 – 934, 1998.

[Sal08] S. Salsa. *Partial Differential Equations in Action: From Modelling to Theory.* Springer-Verlag Italia, Milano, 2008.

[SLSH12] M.A. Scott, X. Li, T.W. Sederberg, and T.J.R. Hughes. Local refinement of analysis-suitable T-splines. *Comp. Meth. Appl. Mech. Eng.*, 213-216(0):206 – 222, 2012.

[SSAPS03] J. Sobieszczanski-Sobieski, T.D. Altus, M. Phillips, and R. Sandusky. Bilevel Integrated System Synthesis for Concurrent and Distributed Processing. *AIAA Journal*, 41(10):1996–2003, 2003.

[SSH97] J. Sobieszczanski-Sobieski and R.T. Haftka. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural optimization*, 14(1):1–23, 1997.

[SZ92] J. Sokołowski and J.P. Zolésio. *Introduction to shape optimization: shape sensitivity analysis.* Springer-Verlag, 1992.

[Ted07] N. Tedford. *Comparison of MDO Architectures Within a Universal Framework.* Canadian theses. University of Toronto (Canada), 2007.

[VGJS11] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comp. Meth. Appl. Mech. Eng.*, 200(49-52):3554 – 3567, 2011.

[VSS05] G. Venter and J. Sobieszczanski-Sobieski. A parallel Particle Swarm Optimization algorithm accelerated by asynchronous evaluations. *J. Aeros. Comp. Inf. Com.*, 2005.

[WFC08] W.A. Wall, M.A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Comp. Meth. Appl. Mech. Eng.*, 197(33-40):2976 – 2988, 2008.

[Wil97] C.K.I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In *Learning and Inference in Graphical Models*, pages 599–621. Kluwer, 1997.

[ZDD11] A. Zerbinati, J.-A. Désidéri, and R. Duvigneau. Comparison between MGDA and PAES for multi-objective optimization. Research Report RR-7667, INRIA, 2011.

[ZDD12] A. Zerbinati, J.-A. Désidéri, and R. Duvigneau. Application of Metamodel-Assisted Multiple-Gradient Descent Algorithm (MGDA) to Air-Cooling Duct Shape Optimization. In *ECCOMAS - 2012*, Wien, Austria, 2012.

[Zol82] J.P. Zolésio. Les dérivées par rapport aux noeuds des triangulations et leurs utilisation en identification de domaine. CRMA Research Report 1110, University of Montreal, 1982.

Acknowledgments

J'ai toujours pensé que l'école, c'était d'abord les professeurs.
(D. Pennac)

Desidero ringraziare innanzitutto il Prof. Luca Formaggia - relatore per la seconda volta - e, come la prima ormai tre anni fa, sempre disponibile nonostante i mille impegni e la distanza spesso remassero contro. Lo ringrazio soprattutto per l'interesse mostrato da subito nei confronti di questo progetto di tesi e per l'entusiasmo con cui mi ha spinto a partire per lavorare nel gruppo di Jean-Antoine Désidéri.

Donc, je veux remercier toute l'équipe Opale de INRIA pour m'accueillir à Sophia Antipolis. Merci Antoine et Régis pour toutes les réponses, les idées et pour les dessins au tableau. Merci Adrien pour les pause-cafés passés dans le bureau à m'expliquer le code et merci pour me parler en italien quand j'étais trop fatigué pour communiquer en français. Puis, je veux remercier toute l'équipe pour ne jamais renoncer à me demander si je voulais faire du sport avec vous même après six mois que je refusais!

La scuola (l'università, nel mio caso) è stata fatta veramente prima di tutto dagli insegnanti: un sentito ringraziamento va a tutti i maestri che ho avuto modo di incontrare in questi anni al Dipartimento di Matematica e in altri Dipartimenti (ma soprattutto in quello di Matematica, eh...). In particolare, per questi ultimi mesi di progetti e tesi e progetti per il futuro, il mio grazie va a Simona ed Elena, per non aver pensato che fossi uno stalker a comparire sempre non annunciato davanti alla porta dei vostri uffici (o a non averlo fatto trasparire quando anche l'avete pensato `:)`).

Por fin, gracias al Profe Juan del Departamento de Matematica de la Universidad Santa María por introducirme al mundo de la optimización y por estar siempre disponible para preguntas, sugerencias, cartas o cualquier cosa. Le agradezco mucho por todo.

Un grazie di cuore agli amici vecchissimi, vecchi, nuovi e nuovissimi che mi hanno accompagnato in questi anni: Bettola, Puzzi, Ari, Elena, il perché *è al di là del bene e del male* (e chissà che c'entri pure le tiercé!); Eri per le convivenze, i progetti, le serate ma soprattutto per l'indipendenza (forse l'abbiamo raggiunta!); Andre per fare più vita sociale di me sempre, comunque e in ogni luogo; Annalix per la forchetta nel naso e altro, altro ancora; Ele Mush perché, sì, abbiamo dei problemi ma le persone che ci circondano non migliorano certo la situazione; Davide per la casa di design, gli occhi di design ma l'ospitalità di una volta; Anna per il buonumore; Nic - mio gemello separato alla nascita - per gli ottimi gusti musicali e la poiesi; Ste per essere stato il mio dirimpettaio di banco, il mio compagno di banco e il mio di fronte di banco assecondando i miei schizzi da Auletta designer; Terry per la capacità di ascoltare; Cami per ricordare sempre tutto di tutti; Anna (Maria) per nutrirci; Nahuel per i grassi in eccesso; Marco per essere una fonte inesauribile di situazioni imbarazzanti!

Grazie al Regio PoliM, alla Nave, all'aria condizionata del Tender, alla sindrome di Stoccolma, all'Aul(ett)a tesisti, allo spleen che vi risiede e al feng shui.

Quiero agradecerle a mi Chile largo y angosto y a todos mis chilenos (aweonaos, jajaja) queridos, por su energía y sus colores. ¡Porque la guata es italiana pero el corazón chileno! Gracias Nasty, Paci, Vico, Negra, Codita, Elena, Jemi, Basti, Paola, Ro, Cata, Pia, Jorge, Felipe, Nico, Pancho, Jean Paul, Pablo, (más sobrio que el) Lili, Lore, Ana, Kata y todos los compadres. Gracias a Casa Nahuel. Gracias a Valpo, puerto extraordinario y a todas las marchas, a los carretes, a las piscolas y a la buena onda latina.

Grazie a: i mezzi pubblici, teatro di incontri e di numerose svolte essenziali per questa tesi; l'informatica (anche quando rasenta la stregoneria) e i buchi nel kernel Linux, perché iniziare ogni giornata con il computer che si accende normalmente è troppo mainstream; Julio Iglesias, perché ne sapeva; l'internet, fonte inesauribile di autostima; la storia, la filosofia, la letteratura, l'arte e la loro versione 2.0 (wiki) per la direzione artistica; Mjölnir; l'AIM; Whitneeeey; Giordano Bruno (e uno); la babbaggine e i probblemi; i miei capelli, per essere restati al loro posto; LaTeX, perché da grande fare quello che scrive gli oroscopi in LaTeX resta un ottimo piano B; Giorgio Canali, fedele compagno di tante serate; Ellen Degeneres, Stefan Kramer, Natalino Balasso, l'Armadillo e Panda, per il sorriso nei momenti in cui ce n'era DAVVERO bisogno.

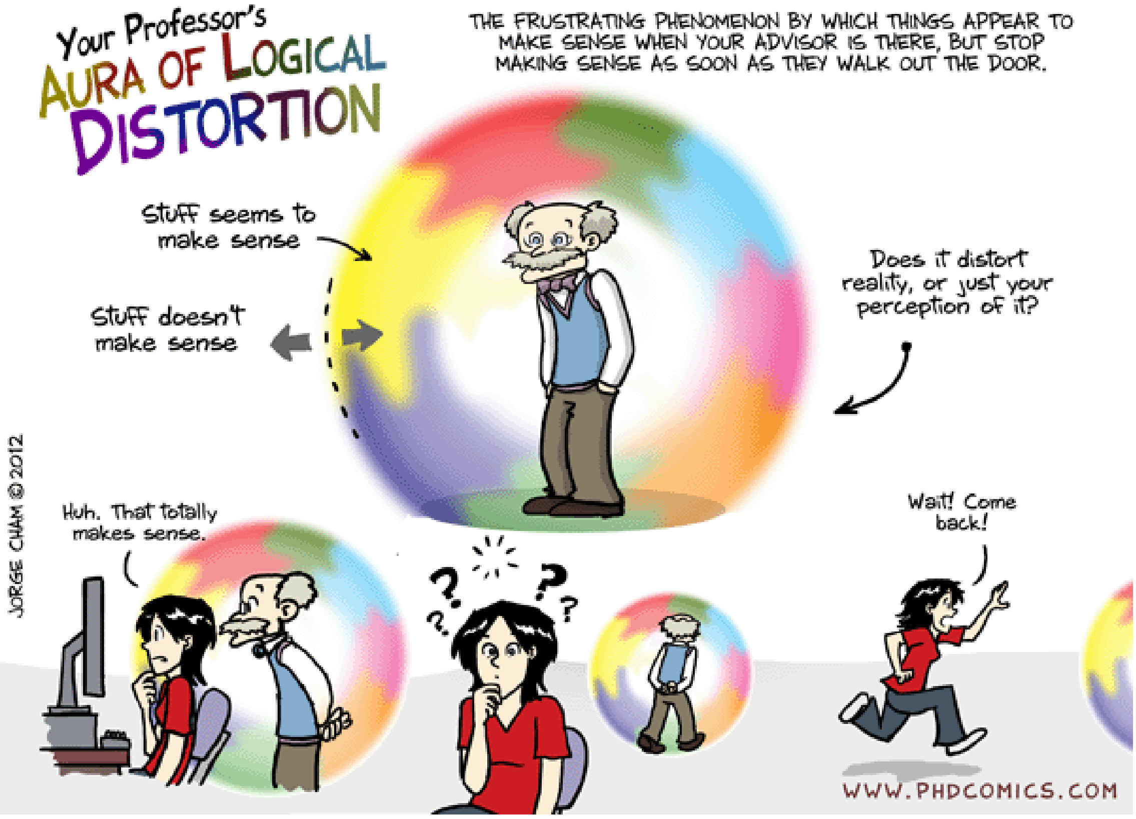
Gracias a todos los compañeros, merci a tous mes amis, thanks to all my friends. Grazie a Mirco, Noemi, Zambo, Massi, Mike, Ste, Nico, Albe, Anna, Manu, Frap, Gio, Luca, Cami, Marta, Simo Rosso, Bia, Giada, Ali, Giacomo, Mattia, Cecio, Paso, Elvio e a chi sa che è stato dimenticato perché io sono uno svampito: questa tesi è anche merito vostro, ma soprattutto mio, come disse la saggia Cadonna.

Grazie a Sarita e Minissima, per l'enoteca da Sarita con Mini cucina e quel sapore di casa a Lambrate Capitale. Ringrazio i miei genitori che mi sono sempre stati vicini e continuano a esserlo, qui oggi e tutte le volte che prendo un biglietto di sola andata per *Nonsisabenedove*. Grazie davvero speciale a EleBele: nonostante tu faccia di tutto per spedirmi all'estero appoggiando tutte le mie scelte e spronandomi sempre e comunque, sappi che non ti libererai di me tanto facilmente! Grazie anche a Claudio e un grazie in anteprima a chi sta arrivando.

Grazie alla musica, compagna inseparabile e colonna sonora costante.

Gracias a la vida que me ha dado tanto.

Matteo


Your Professor's
AURA OF LOGICAL DISTORTION
THE FRUSTRATING PHENOMENON BY WHICH THINGS APPEAR TO MAKE SENSE WHEN YOUR ADVISOR IS THERE, BUT STOP MAKING SENSE AS SOON AS THEY WALK OUT THE DOOR.
Stuff seems to make sense
Stuff doesn't make sense
Does it distort reality, or just your perception of it?
JORGE CHAM © 2012
Huh. That totally makes sense.
Wait! Come back!
WWW.PHDCOMICS.COM

Courtesy of "Piled Higher and Deeper" by Jorge Cham - www.phdcomics.com