ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

INGEGNERIA ENERGETICA NUCLEARE
E DEL CONTROLLO AMBIENTALE

ING-IND/10 Fisica Tecnica Industriale

XIX CICLO

APPLICATION OF EVOLUTIONARY TECHNIQUES TO ENERGY TRANSFER EFFICIENCY IN HEAT TRANSFER PROBLEMS AND LOW CONSUMPTION BUILDINGS

Il Coordinatore del Corso di Dottorato
Chiar.mo Prof. Ing. Alessandro Cocchi

Il Relatore
Chiar.mo Prof. Ing. Enrico Nobile

Candidato Dott. Francesco Pinto

Trieste, 2007

Contents

Heat Transfer Problems 19

Preface

The continuing growth of available computational resources has modified the role of numerical simulation, that is now increasingly used in industrial applications, and not only in academical environments. In design, construction, and maintenance of any engineering system, technological and managerial decisions have to be taken at several stages. The possibility to perform complex and complete numerical simulations opens new perspectives, raising the issue of which is the best way to face an optimization process, where optimization can be defined as the act of obtaining the best solution under given circumstances. Usually there exist a relation between a set of decisional parmeters and a series of assessment criteria, so called objective functions, which influence the decision-making process. In problems faced in real situations, the relations between decisional parameters and objectives are in most cases unknown. Moreover, there usually are several and conflicting assessment parameters which lead to multi-objective optimization problems, where not a single optimum solution exists, but rather a set of equally valid ones. These aspects underline the substantial weakness of traditional optimization approaches that can deal with single-objective functions, which have to satisfy continuity and derivability constraints.

For these reasons interest has been recently focused on evolutionary optimization techniques, which are heuristic methods that use some mechanisms inspired by biological evolution. An important feature of evolutionary algorithms is the applicability to almost all types of problems, because they do not make any assumption about the system under study as classical techniques do. In addition, these kind of algorithms can deal with truly multi-objective optimizations and are usually robust, in contrast to gradient-based procedures, that are likely to get trapped in local optimal solutions.

Evolutionary techniques have been widely used throughout this thesis, so before describing the reseach activities, in chapter 1 an introduction to optimization methods is given. After a first classification of optimization problems, attention is shifted to *design of experiments*, that is a methodology applicable to the design of all information-gathering activities where variation of decisional parameters is present. It is a technique aimed at gaining the most possible knowledge within a given dataset. In evolutionary optimizations, design of experiments is used to obtain a good initial sampling of the design space, which is of great relevance in reducing optimization effort and improving results. Successively, the concept of Pareto optimality in multiobjective optimizations is introduced, and a series of evolutionary algorithms is illustrated, with particular emphasis on genetic algorithms. *Multi criteria decision making* is then introduced, which refers to the solving of decision problems involving multiple and conflicting goals, coming up with a final solution. Finally, the concept of *metamodel* is given. A metamodel is in pratice a model of a model, whose aim is to virtually explore a design space, thus drastically reducing the exploration time.

The research activities reported in this thesis focus on two principal branches. The first part concerns the study and optimization of heat transfer problems, while the second part deals with energy savings in buildings.

In the fist part two different kind of heat transfer problems are discussed in which the application of evolutionary optimization techniques is exploited to reach the desired goal. The problems that are going to be surveyed deal with geometry shapes and can be considered subsets of shape optimization problems. The objectives of the studies are functions of their physical domain, whose change in form affects the behaviour of the system. In this sense, great attention is to be given to the method by which shapes are mathematically represented.

Over the last few decades *computer aided design* (CAD), *computer aided manufacturing* (CAM), and in general *computer aided engineering* (CAE) tools have been thriving. Nowadays these tools are an absolutely necessary routine in development of products, in whichever sphere of activity. Among the amount of CAE instruments, it is of most relevance a methodology for the geometrical representation of the models to be developed.

Shape optimization is an infinite-dimensional optimization problem in the sense that the input variables of such problems are continuous entities (curves or surfaces) that cannot be determined by a finite number of degrees of freedom. Therefore the issue of a well conditioned geometrical model is of ultimate importance. The choice of a good parametrization is not a trivial task. Depending on the (usually unknown) optimal shape, the model has to be complete enough as to match the desired target. Yet if it is overdeveloped this may lead to slow or unstable optimization processes. In chapter 2 an overview on geometrical representations is outlined, with particular attention to Bézier and NURBS curves and surfaces, that have been used in the following chapters to draw the computational domains, and represent the standard for form description and manipulation in industrial 3D CAD (solid modelling) systems.

The problem considered in Chapter 3 is the multi-objective optimization of two-dimensional convective wavy channels, which represent the fundamental building block of many heat exchangers and heat transfer devices. The study is limited to a single channel at fully developed flow and heat transfer conditions. In this case, channels of periodic cross section can be considered periodic in the flow and thermal fields as well. Therefore the computational domain of interest becomes a single periodic module of the entire geometry. The optimization of the two-dimensional periodic channel is obtained, by means of an unstructured finite-element solver, for a fluid of Prandtl number $Pr = 0.7$, representative of air and other gases. The objectives of the optimization are the minimization of the friction factor and the augmentation of the heat transfer rate. These are clearly conflicting objectives, and a single solution optimizing both objectives does not exist. It is known that opportunities for heat transfer augmentation for two-dimensional steady flows is very limited, but nevertheless, due to computational savings compared to three-dimensional or time-dependent flows, the design space can be chosen rather large, and the accuracy can be verified more economically. In a second phase of the work, the influence of secondary motions on heat transfer is assessed on optimized two-dimensional geometries, and an optimization carried out on a simplified three-dimensional geometrical model.

In chapter 4, inverse heat transfer problems are considered. These are ill-posed problems that admit a solution if, and only if, the geometrical domain can be appropriately modified. In such cases boundary conditions are overspecified in order to make the transfer phenomenon behave in a predefined way. In this work a genetic algorithm has been used to reproduce the two-dimensional direct design of shape considered by other authors, but where a gradient-based method had been applied. A heated substrate is embedded in a solid body and a determined constant surface temperature is sought. The numerical solution of simply conductive problems is less computationally expensive than conjugate (conductive + convective) ones. So, in the first part of the work a two dimensional conductive problem is presented in order to test different geometrical parametrizations. The best geometrical model is then used to extend the optimization to the conjugate case. In the third part of the work a further extension is made towards the solution of a conjugate three-dimensional case. Due to the lack of dedicated CFD procedures in COMSOL, the general purpose finite element solver used in this thesis, a segregated approach to solve Navier-stokes equations has been implemented to carry out the three-dimensional optimization.

In the second part of the thesis, problems related to energy savings in buildings are tackled. Today, a disproportionately high 50% of all primary energy is used in buildings, with 50% to 90% of this to maintain tolerable indoor temperatures - i.e., for space heating and cooling in developed countries. With the total world consumption of marketed energy expected to increase by over 70% in the next two and a half decades, this continued level of energy demand to keep our buildings habitable is clearly not sustainable. A solution to the increase of energy consumption, diminishing fossil fuels, and global warming, that is fairly sought in alternative energy sources, cannot but undergo a process of reduction of power consumption by the design of efficient buildings.

In chapter 5 a passive technique air conditioning system is described. The use of HVAC systems is becoming highly popular, and thus, the development of efficient cooling techniques is a very important research task to prevent an uncontrolled energy consumption increase. Night ventilation is a passive cooling technique that can significantly reduce the cooling loads and energy requirements, but a trade off must be made between energy cost savings and zone thermal comfort. The impact of night ventilation on energy consumption is affected by climate, building and control parameters. In this scenario the application of evolutionary multi-objective techniques can be helpful in developing optimized cooling systems. In this chapter, it is shown the coupling an optimization tool with a building simulation code, in order to assess the impact of different parameters on the effectiveness of night ventilation applied to an office building.

Finally, in chapter 6 a numerical study on dynamic insulation systems is presented. Dynamic insulation effectively saves energy by exploiting contra-flux heat and mass transport through an air permeable medium, in order to reduce the overall heat transfer coefficient. The present study seeks to achieve a fundamental understanding of heat transfer across the ventilated cavities of a dynamic insulation layer when air flows through the wall. Theory on dynamic insulation is well developed in the monodimensional stationary case, where analytical solutions to the energy equation exist. Yet in cavities of actual dynamic insulation constructions the flow pattern is two dimensional. This is the first investigation of its type undertaken to evaluate the effects of the two dimensional path in the cavities.

Chapter One

An overview on optimization techniques

In design, construction, and maintenance of any engineering system, technological and managerial decisions have to be taken at several stages. The objective of such a decision process is either to maximize the desired benefit or to minimize the required effort in terms of time, money, materials, and energy. Such a process can be abstracted from the engineering field and applied to whatsoever situation in which human logic is present.

A decision-making environment can be linked to the concept of system. A system is an entity, a set of logical or physical connections that gives determinate outputs, when it undergoes to certain inputs. Whoever wants to look into a system, he has first of all to educe a model of it: the simplest possible representation, yet bearing the most important features of the system itself. By means of its model a system can be studied and improved using mathematical tools, whenever a quantitative relation between inputs and outputs can be established. Thus a system ca be seen as a *black box* acting on inputs to prodice outputs, as in the following relation:

$$\mathbf{O} = \left\{ \begin{array}{c} O_1 \\ ... \\ O_m \end{array} \right\} = f(\mathbf{X}) = f \left\{ \begin{array}{c} X_1 \\ ... \\ X_n \end{array} \right\} \tag{1.1}$$

where $\mathbf{O}$ is a set of outputs and f is a generic relation linking outputs to inputs $\mathbf{X}$.

Optimization is the act of obtaining the best solution under given circumstances, Rao fairly states in [1]. From a system point of view, this means one is searching a maximun, or respectively a minimum for function f, depending on the desired goal. Without loss of generality, noting that the maximun of f coincides with the minimum of its opposite $-f$, optimization problem can be taken as minimization ones.

The existence of optimization methods can be traced to the birth of differential calculus, that allows minimization of functionals, in both unconstrained and constrained domains. But in real problems, function f is unlikely to be a simple analytical expression, in which case the study of the function is suggested by classical mathematical analysis. It is rather a usually unknown relation, that might lack of continuity, derivativeness, connectedness. Differential calculus is not of any help in such circumstances.

When a relation is unknown a *trial and error* methodology is the oldest practice, and no further contributions to optimization techniques has been provided until the advent of digital computers, which have made implementation of optimization procedures a feasible task. From an optimization viewpoint inputs and outputs in eq. 1.1 can be renamed after their conceptual meanings. Inputs are usually known in literature as *design variables*, while outputs, being the goal of an optimization process, are known as *objective functions* or simply *objectives*. In many practical problems, design variables cannot be chosen arbitrarily, but they have to satisfy specified requirements. These are called *design constraints*. Even the objectives could undergo restrictions. They are called*functional constraints*. In addition to eq. 1.1, the two kind of constraint only just introduced, can be formally expressed in the case of inequality relations as

$$g\left(\mathbf{X}\right) \leq 0 \tag{1.2a}$$

$$m\left(f\left(\mathbf{X}\right)\right) \leq 0 \tag{1.2b}$$

where g and m are two general applications. Equality relations are easily obtained replacing the symbol $\leq$ with $=$. Optimization problems can be classified in several ways, that depend on different aspects of problems themselves. In [1], the following classifications are highlighted.

Classification can be based on:

1. *Existence of constraints.* As stressed earlier, problems can be classified constrained or unconstrained. Constraint handling is not a trivial task for the most part of optimization techniques

2. *Nature of design variables.* f can be function of a *primitive* set of variables depending on further parameters, thus becoming *trajectory optimization problems* [2]

3. *Physical structure of the problem.* Depending on the structure of the problem, *optimal control* theory can be applied, where a global cost functional is minimized to obtain the desired solution.

4. *Nature of the relations involved.* When known or at least well guessed, the nature of the equations governing the model of the system under study can address the choice to the most efficient among a set of optimization methods. Linear, quadratic geometric and nonlinear programming are examples.

5. *Permissible values of design variables.* Design variables can be real valued or discrete.

6. *Deterministic Nature of the variables.* The deterministic or stochastic nature of the parameters is a criterion to classify optimization problems. In particular,

the concept of *Robust Design* or *Robust Optimization* has gained popularity in recent times [3].

7. *Separability of the functions.* A problem is considered separable if f functions ca be considered a combination of functions of single design variables $f_1(X_1)$, $f_2(x_2), \ldots,$ $f_2(X_n)$ and f becomes:

$$f(\mathbf{X}) = \sum_{i=1}^{n} f_i(X_i)$$

The advantage of such a feature is that in nonlinear problems, nonlinearities are mathematically independent [4].

8. *Number of objective functions.* Depending on the number of objective functions, problems can be single- or multi-objective. This is an outstanding distinction, for in multi-objective optimizations objectives are usually conflicting. No single optimum exist, rather a set of designs the decision maker has to choose among. This is one of the motivations that have led to the birth of *Evolutionary Multi Objective Optimization* (EMOO) [5], widely used in this thesis.

Depending on the characteristics of optimization problems many techniques have been developed to solve them. Techniques can be roughly divided into two categories.

1. *Traditional mathematical programming techniques.* They require a certain knowledge of the relation between objectives and design variables, and they are usually best suited for single objective optimizations.

2. *Evolutionary Algorithms* (EA). They are heuristic methods that use some mechanisms inspired by biological evolution: reproduction, mutation, recombination, natural selection and survival of the fittest. Their most important feature is the applicability to almost all types of problems, because they do not make any assumption about the system under study as classical techniques do. Namely, they can be used when relation f is a completely unknown function.
In their Multi-objective version, *Multi Objective Evolutionary Algorithms* (MOEA), being part of the just only cited reseach area known as evolutionary multi objective optimization (EMOO), they are capable of dealing with truly multi-objective optimizations [6].

Differential calculus is the first example of traditional programming techniques, but there is a widely developed literature [1]on the subject. Linear programming, quadratic programming, nonlinear programming, are just examples. To perform the optimization processes exposed in this thesis it has been made use of evolutionary techniques. The software modeFRONTIER© has been exploited, which is a state-of-the-art optimization tool that includes the most of instruments relative to data analysis and single and multi objective optimizations.

In this chapter a review addresses to evolutionary optimization techniques and common practice in evolutionary optimum search implemented in the software.

1.1 Design of Experiment

Heuristic evolutionary techniques do not make any assumption on the relation between objectives and design variables, thus providing an analogy with experimental dataset analysis. A good initial sampling, which allows an initial guess on the relations between inputs and outputs, is of great relevance in reducing optimization effort and improving results [7].

Design Of Experiments (DOE) is a methodology applicable to the design of all information-gathering activities where variation of decisional parameters (design variables) is present. It is a technique aimed at gaining the most possible knowledge within a given dataset. The first statistician to consider a formal mathematical methodology for the design of experiments was Sir Ronald A. Fisher, in 1920.

Before the advent of DOE methodology, the traditional approach was the so called *One Factor At a Time* (OFAT). Each factor (design variable) influencing the system used to be moved within its interval, while keeping the others constant. Inasmuch as it require a usually large number of evaluations, such a process reveals quite time consuming. Fischer's approach is to consider all variables simultaneously, varying more than one at a time, as to obtain the most relevant information with the minimun of the effort. It is a powerful tool for designing and analyzing data. It eliminates redundant observations, thus reducing time and resources in experiments and giving a clearer understanding of the influence of design variables. Three main aspects must be considered in choosing a DOE:

1. The number of design variables (i.e. domain space dimension);
2. The effort of a single experiment;
3. The expected complexity of the objective function.

1.1.1 DOE algorithms

Full Factorial Full factorial (FF) algorithm sample each variable span for n values called *levels* and evaluates every possible combination. The number of total experiments is

$$N = \prod_{i=1}^{k} n_i \tag{1.3}$$

where n_i is the number of levels for the i-th variable and k is the number of design variables. Full factorial provides a very good sampling of the variables domain space, giving complete information on the influence of each parameter on the system. The higher the number of levels, the better information. Anyway, this algorithm bears an important drawback. The number of samples exponentially increase with the number of variables, which makes the use of FF unaffordable in many practical circumstances.

Reduced Factorial Reduced Factorial DOE tries to overcome the main limitation of Full Factorial (i.e. the large number of designs) while keeping its advantages. Full Factorial is performed on a sub-set of input variables. By applying this sampling algorithm high order interactions are difficult to estimate. Fortunately, higher-order interactions are rarely important and for most purposes it is only necessary to evaluate the main effects of each variable.

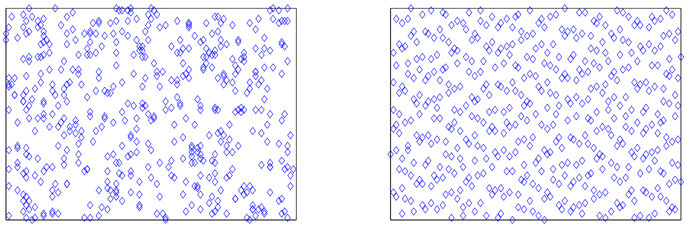
Cubic Face Centered This method is equivalent to a two levels FF plus the midpoints of the design space hypercube. The experiments are placed in the design variables hyper-cube as follows:

1. On each vertex;
2. On the center of each face;
3. On the hypercube's center.

The total number of experiments for n variables is $2^n + 2n + 1$. This method allows the computation of second order interactions and can be useful when the problem is weakly non linear and a full factorial with three levels is too expensive.

Random Sequence The random algorithm fills randomly the design space, applying a uniform distribution. This sampling is best suited for high number of input variables, where other algorithms result in a too expensive DOE.

Sobol Sobol algorithm creates sequences of n points that fill the n-dimensional space more uniformly than random sequence does. This types of sequences are called *quasi-random* sequences. This term is misleading, since there is nothing random in this algorithm. The data in this type of sequence are chosen as to avoid each other, filling in a uniform way the design space.



(a) random sampling

(b) sobol sampling

Figure 1.1 Sobol sampling picks uniformly distributed designs

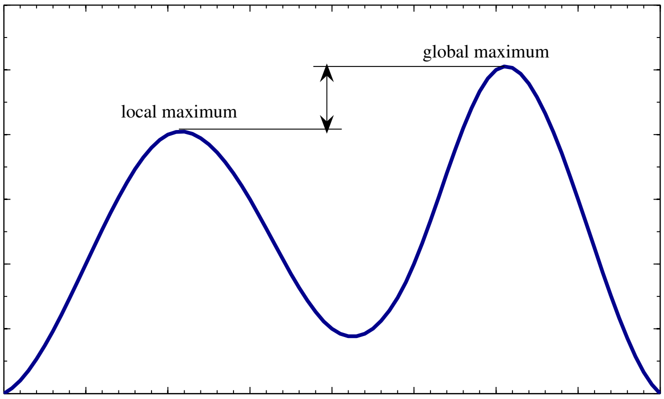


Figure 1.2 multiple extrema points

1.2 Optimization Algorithms

Optimization algorithms investigate the behaviour of a system, seeking for design variable combinations that give optimal performances. In terms of objective functions values, an optimal performance means the attainment of extrema. Extrema are points in which the value of the function is either maximum or minimum. Generally speaking, a function might present more than one extreme point, called *local extrema*, see figure 1.2. It is of great importance for an algorithm to be capable of finding the *global extremum* of an objective with the minimum effort. Three are the main characteristics that distinguish and classify the efficiency an optimization algorithm:

- **Robustness** Robustness is the capability of reaching a global optimum point without being stuck in local extrema, or blocked for lack of useful data. This is the most important feature in measuring the efficiency of an optimization technique. The more an algorithm is robust, the higher the chance to reach a global optimum.

- **Accuracy** Accuracy is the ability of an algorithm to reach the actual extrema, either global or local, when in the proximity of it. Usually, accuracy and robustness are conflicting attributes, so robust algorithms are not accurate and vice versa.

- **Convergence rate** Convergence rate is a measure of the effort an algorithm has to carry to reach its goal out. Again, robust algorithms are usually slow, yet fast but not robust ones might not reach the goal at all.

This survey focus on evolutionary techniques that are usually robust but neither accurate nor fast. Yet, as already stressed, their most important attribute and reason for

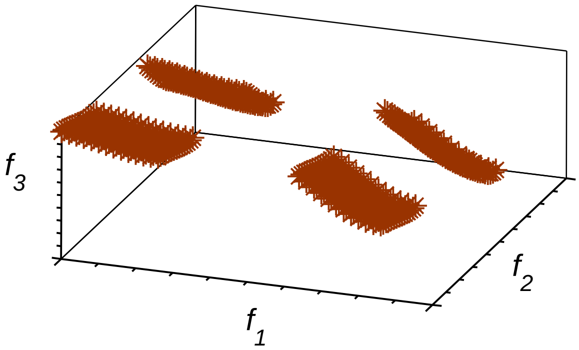


Figure 1.3 DTLZ6 test function

a spread use is the applicability to almost any single- or multi-objective optimization problem of whichever complexity. In particular EMOO is a recent area of study, in which a lot of work still has to be done. The weakest aspect of EMOO research lies on the theoretical side. Being EA heuristic processes, most of the current reseach aims at proving actual convergence [6, 8]. Nevertheless the wide literature on successful applications of evolutionary optimizations sets EMOO as a promising field. As an example in figure 1.3 the DTLZ6 function*Pareto set* [1] is shown. It has been proposed by Deb et al. [9] to represent among others a benchmark for algorithm testing. It has 22 design variables and 2^{20} disconnected regions. Quite a cumbersome problem that show the potential of evolutionary algorithms.

1.2.1 Pareto optimality

As soon as there are many, possibly conflicting, objectives, it is rarely the case that there is a single design variables combination that simultaneously optimizes all the objective functions in a multi-objective optimization problem. Rather, there usually exists a whole set of possible solutions of equivalent quality. This can be the case of heat exchangers, studied in chapter 3: the desired objectives might be to maximize heat transfer rate per unit volume, to minimize the pumping power, to minimize cost and weight, and to minimize the performance degradation due to fouling. These goals are clearly conflicting and, therefore, there is no single optimum to be found. For this reason, the so-called *Pareto dominance* or *Pareto optimality* concept must be used, according to which *design a dominates design b* if and only if

$$(\forall i\; f_i(a) \geq f_i(b)) \cap \left(\exists\, j :\; f_j(a) > f_j(b)\right) \tag{1.4}$$

where f_i is the *i-th* objective function, and for simplicity it is assumed that we are considering the maximization of the n objectives. Expression (1.4) means that at least

[1] Pareto set concept will be introduced hereafter in this chapter

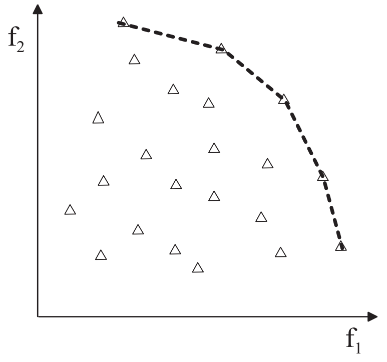


Figure 1.4 Pareto front for a two objectives optimization.

in one purpose design a is better than design b, while in the others they might be equal. Let us consider, as an example, a problem where it is desired to maximize two objective functions f_1 and f_2. Each design evaluation, during the optimization, produces an $[f_1, f_2]$ couple, and all these data can be represented graphically as in figure 1.4. The relations (1.4) allows the determination of a design set - those joined by a dashed line - which is called *Pareto front* or *Pareto optimal set* and whose dimension is equal to $n - 1$. In this example $n = 2$, and the front is a line.

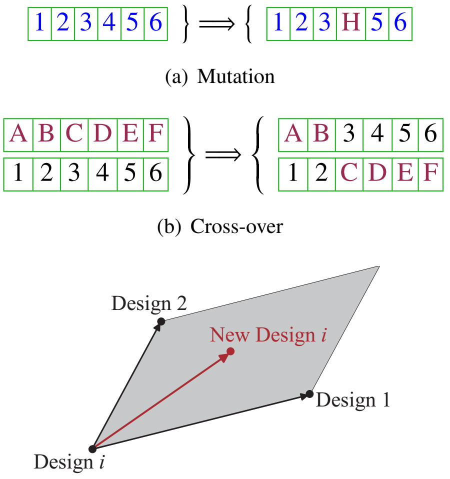
1.2.2 Basic Evolutionary algorithms

Genetic Algorithm

Genetic algorithm (GA) is the most popular type of EA. The basic idea underlying the method comes from the behaviour of living organisms in nature. An initial set of *individuals*, called initial *population* undergoes a natural selection process. So each individual can be seen as a DNA string. Parental populations give birth to offsprings. Genetic algorithms works on individuals as coded bit strings, thus they need discrete variables intervals. The new generations are created following a series of genetic rules:

- **Selection** Selection operator randomly shift a defined number of individual to the next generation, keeping them unchanged. The probability for an individual to undergo a process of selection is weighed on the fitness value of each design. The better the fitness, the higher the probability to be selected for the new population.

$$P(\mathbf{x}_i) = \frac{F(\mathbf{x}_i)}{\sum_{j=1}^{n} F(\mathbf{x}_j)}$$



(a) Mutation

(b) Cross-over

(c) Directional Cross-over

Figure 1.5 evolutionary operators of a Genetic Algorithm

- **Mutation** Mutation operator consists in the random substitution of some bits (nucleotides) in the numeric string representing an individual. the role of mutation is to enhance the probability of exploring untouched areas of the design space avoiding premature convergence. Mutation generally involves less than 10% of the individuals.
- **cross-over** cross-over is a genetic recombination between two individuals, whose strings are accidentally cut and rejoined

modeFRONTIER© version of GA implements a fourth operator called *directional cross-over*. It assumes that a direction of improvement can be detected comparing the fitness values of two reference individuals. This operator usually speeds up the convergence process, though it reduces robustness.

Directional cross over works as follows:

1. Select an individual i;
2. Select reference individuals i_1 and i_2
3. create the new individual as:

$$\mathbf{x}_{i_{new}} = \mathbf{x}_i + s sign(F_i - F_{i_1})(\mathbf{x}_i - \mathbf{x}_{i_1}) + s sign(F_i - F_{i_2})(\mathbf{x}_i - \mathbf{x}_{i_2})$$

where s and t are two random parameters. In figure 1.5 genetic operators behaviour is sketched. Each operator can be applied with a certain probability. Different combination of operators probability may lead to different robustness, accuracy and convergence rate

Evolution strategies

Evolution strategies use real-vectors as coding representation, and primarily mutation and selection as search operators.

Assuming as a reference a real valued function

$$f : \mathbf{x} \in A \subset \mathbb{R}^n \longrightarrow \mathbb{R}$$

evolution strategies works as follows:

1. Choose a random initial population of m $\mathbf{x}_i$ individuals
2. Create an offspring vector $\mathbf{x}_i'$ by adding a Gaussian random variable N with zero mean and determined standard deviation σ_i to each component x_k of the parent vectors
$$\mathbf{x}_i' = \mathbf{x}_i + N(0, \sigma_i)$$
3. selecting m individuals among parents and children by comparing their fitness.

The classical variant of evolution strategies is *(1+1)evolution strategy* where a single parent competes with a single child.

The first version employed constant standard deviation (which can be seen as a step size) in each dimension, thus leading to slow convergence. Now, it employs the 1/5-th success step size control rule proposed by Schwefel and Rechenberg in order to control the variance of the mutation steps automatically.

Reproduction is carried out in two steps. The first step consists in computing a parameter p that gives the relative frequency of steps where the child replace the parent during selection. If this quantity is grater than 1/5 then the standard deviation is increased by a certain factor c, while if p is smaller than 1/5 it is reduced by the same factor.

The (1+1)-Evolution Strategy is a very fast evolutionary algorithm. But Compared to GAs it might reach converge prematurely. However, its rank-based selection mechanism and the gradual success-based refinement of the step-sizes, starting from high step-sizes, makes it robust for slightly noisy functions and in the presence of discontinuities.

Frequently single step-size is not an optimal choice because the contours of f are rarely equally spaced in each direction, so performances can be enhanced by introducing independent standard deviations.

Another way to improve Evolution Strategies is by modifying the reproduction scheme: the simultaneous variation of μ parents can be used to obtain one offspring(*(μ + 1)-ES*) and λ (*(μ, lambda)-ES*) offspring per generation can be used for selection. Finally, an aging parameter κ for individuals can be introduced to set the maximal number of iterations that an individual can survive in the parent population. (*(μ, κ, λ)-ES*)

1.2.3 Multi Objective Approaches

When there is a single objective function the definition of a metric for valuating design *fitness*[2] is straightforward. Pareto optimal set reduces to a single point and the best design is a global extreme.

On the other hand, the introduction of multiple objectives tangles thing a bit. It has been already stressed that in multiobjective optimization there exist a set of solutions that present equal quality or effectiveness. These are the non dominated designs, defined as in eq. 1.4. The problem arise of how to compare and judge designs in order to get a scalar evaluation scale for their fitness.

MOEAs approaches can be roughly divided into three categories*Aggregating Functions*, *Population-Based Approaches*, *Pareto-based Approaches*

Aggregating Functions

The most straightforward approach in handling multiple objectives is the use of an arithmetical combination of all the objective. Thus the resulting single function can be studied with any of the single objective algorithms either evolutionary or classical. Aggregating approaches are the oldest mathematical programming methods found in literature [1].

Applied to EA, the aggregating functions approach does not require any change to the basic search mechanism. Therefore it is an efficient, simple, and of easy implementation. It can be successfully used on simple multiobjective optimizations that present continuous convex Pareto fronts. An example of this approach is is a linear sum of weights of the form:

$$min \sum_{i=1}^{m} w_i f_i(\mathbf{x})$$

where the weight w_i represent the relative importance of the m-th objective function. The weighting coefficients are usually assumed to sum at 1:

$$\sum_{i=1}^{m} w_i = 1$$

Aggregating functions may be linear as in the previous example or non linear. Both types of function have been used with evolutionary algorithms but, generally speaking, aggregating methods are underestimate by EMOO researchers because of some limitations in generating complex Pareto fronts. However nonlinear aggregating function do not necessarily present such limitations [5]. Aggregating functions are widely used

[2] *fitness* is the measure of how a design variable set *fit* the goal of an optimization.

in another, completely separate from EMOO branch of optimum solutions search, called *Multi-Criteria Decision Analysis* (MCDA) or *Multi Criteria Decision Making* (MCDM) [10]. MCDA is a discipline aimed at supporting decision makers who are faced with making numerous and conflicting evaluations. EA users are used to apply MCDM for *a posteriori* analysis of optimization results, but the discipline can be applied in a much more sophisticated way for *a priori* analysis.

MCDM is briefly introduced in section 1.3.

Population-Based Approaches

In these techniques the population of an EA is used to diversify the search, but the concept of Pareto dominance is not directly incorporated into the selection process.

The first approach of this kind is the *Vector Evaluation Genetic Algorithm* (VEGA) introduced by Schaffer [11]. At each generation this algorithm performs the selection operation based on the objective switching rule, i.e., selection is done for each objective separately, filling equally portions of mating pool (the new generation) [12]. Afterwards, the mating pool is shuffled, and crossover and mutation are performed as in basic GAs. Solutions proposed by VEGA are locally non-dominated, but not necessarily globally non-dominated. This comes from the selection operator which looks for optimal individuals for a single objective at a time. This problem is known as *speciation*. Groups of individuals with good performances within each objective function are created, but non-dominated intermediate solutions are not preserved.

Pareto-based Approaches

The drawbacks of VEGA as a starting point, Goldberg [13] proposed a way of tackling multi-objective problems that would become the standard in MOEA for several years.

Pareto-based approaches can be historically divided into two generations. The first is characterized by *fitness sharing* and *niching* combined with the concept of *Pareto raking*.

Keeping in mind the definition of non-dominated individual given in section 1.2.1, an individual's rank correspond to the number of individuals by which it is dominated. Pareto front element have a rank equal to 1.

The most representative algorithms of the first generation are *Nondominated Sorting Genetic Algorithm* (NSGA), proposed by Srinivas and Deb [14], *Niched-Pareto Genetic Algorithm* (NPGA) by Horn et al. [15], and *Multi-Objective Genetic Algorithm* by Fonseca and Fleming [16].

The second generation of MOEAs was born with the introduction of *elitism*. Elitism refers to the use of an external population to keep track of non-dominated individuals. In such a way good solution are never disregarded in generating offsprings.

1. **MOGA-II** Uses the concept of Pareto ranking. Considering a population of n individuals if at generation t individual x_i is dominated by $p_i^{(t)}$ designs of the

current generation, its rank is given by:

$$rank(x_i, t) = 1 + p_i^{(t)}$$

All non-dominated individuals are ranked 1 Fitness assignment is performed by:

a) Sort population according to rank;

b) Assign fitness to individuals by interpolating from the best to the worst

c) Average the fitness of individuals with the same rank. In this way the global fitness of the population remains constant., giving a quite selective pressure on better individuals.

The modeFRONTIER© version of MOGA-II includes the following smart elitism operator [17]:

a) MOGA-II start with an initial population P of size n and an empty elite set $E = \emptyset$

b) For each generation compute $P' = P \cup E$

c) If cardinality of P' is greater than cardinality of P, reduce P' randomly removing exceeding points

d) Generate P'' by applying MOGA algorithm to P'

e) Calculate P'' fitness and copy non-dominated designs to E

f) Purge E from duplicated and dominated designs

g) If cardinality of E is greater than cardinality of P randomly shrink the set

h) Update P with P'' and return to step (b)

2. **NSGA-II** It employs a fast non-dominated sorting procedure and uses the crowding distance (which is an estimate of the density of solutions in the objective space) as a diversity preservation mechanism. Moreover, NSGA-II has an implementation of the crossover operator that allows the use of both continuous and discrete variables. The NSGA-II does not use an external memory as MOGA does. Its elitist mechanism consist of combining the best parents with the best offspring obtained (a $(\mu + \lambda)$-selection as in evolution strategies).

1.2.4 Gradient based algorithms

Gradient based methods employ the partial derivatives of the objective function to find the directions of maximum increment and move forward to an extremum.

This kind of algorithms can be applied only to single objective optimizations, and are particularly used for refinement purposes as they are usually accurate. As they need for a starting point and follow a path given by a gradient, depending on the initial conditions they have high chance to get stuck in local extrema.

Restrictions apply to the design space, as it has to be continuous and differentiable. The success of gradient based methods depends even on the absence of distubing noise on the objective function. They can be used in multiobjective problems, once an aggregate function has been created, but they are generally used in combination with RSM techniques, as the number of experiments to calculate partial derivatives increases with the number of design variables.

There are several algorithms that are based on the gradient method, like Cauchy, Newton, quasi-Newton, Broyden-Fletcher-Goldfarb-Shanno (BFGS), Sequential Quadratic Programming (SQP), Conjugate Gradient (CG), etc. The basic concept goes back to *steepest descent Cauchy method* (1847): the others implement modifications in order to enhance convergence[1, 18]. Steepest descent method basic implementation is:
given a function:

$$f : \mathbf{x} \in A \subset \mathbb{R}^n \longrightarrow \mathbb{R}$$

1. Choose an arbitrary initial point $\mathbf{x}_i$ setting the initial iteration $i = 1$

2. Search for the steepest descent direction:

$$-\nabla f(\mathbf{x}_i)$$

3. update point $\mathbf{x}_i$:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \lambda_i \nabla f(\mathbf{x}_i)$$

   where λ_i is a step length parameter

4. test $\mathbf{x}_{i+1}$ for optimality.

5. if optimum reached stop process, else go back to point 2

1.2.5 Downhill Simplex method

The downhill simplex method is a single-objective optimization algorithm due to Nelder and Mead. It requires only function evaluations, not derivatives.

It is not very efficient in terms of number of function evaluations it requires. However, the downhill simplex method may frequently be the best method to use because of its simplicity. It is best suited for not highly non-linear continuous functions.

The algorithm needs an initial set of $n + 1$ initial points (n being the number of variables) and proceeds iteratively rejecting, at each step, the worst configuration. The set of $k + 1$ configurations is a geometric figure called simplex: in two dimensions a simplex is a triangle, while in three dimensions it is a tetrahedron. The replacement point in the simplex is computed accordingly to three predefined rules:

- Reflection
- Expansion
- Contraction

After the initialization the downhill simplex method takes a series of steps, most steps just moving the point of the simplex where the function is highest through the opposite face of the simplex to a lower point (going downhill). These steps are called reflections, and they are constructed to conserve the volume of the simplex (hence maintain its non-degeneracy). If a preferential direction is found, simplex tries an expansion to increase convergence speed. During the final steps, simplex contracts itself to fit the lowest point, until desired accuracy is obtained.

1.3 Multi-Criteria Decision Making (MCDM)

As already stated, in a multi-objective optimization process it is impossible to find out a unique best solution, but rather a whole group of designs that dominate the others: this group is known as the Pareto front or Pareto optimal set. All Pareto optimal solutions can be regarded as equally desirable in a mathematical sense. But from an engineering viewpoint, at the end on an optimization the goal is a single solution to be put into practice. Hence the need for a decision maker (DM) able to identify the most preferred one among the solutions. The decision maker is a person who is able to express preference information related to the conflicting objectives.

Ranking between alternatives is a common and difficult task, especially when several solutions are available or when many objectives or decision makers are involved.

Decisions are taken over a limited set of good alternatives mainly by experience and competence of the single DM. Therefore the decision stage can be said *subjective and qualitative* rather than objective and quantitative. Multi-Criteria Decision Making (MCDM) refers to the solving of decision problems involving multiple and conflicting goals, coming up with a final solution that represents a good compromise that is acceptable to the entire team. As already underlined, when dealing with a multiobjective optimization the decision making stage can be done in three different ways [19]:

- **Decision-making and then search (a priori approach)** The preferences for each objective are set by the decision-makers and then, one or various solutions satisfying these preferences have to be found.
- **Search and then decision-making (a posteriori approach)** Various solutions are found and then, the decision-makers select the most adequate. The solutions presented should represent a trade-off between the various objectives.
- **Interactive search and decision-making** The decision-makers intervene during the search in order to guide it towards promising solutions by adjusting the preferences in the process.

It is already been hinted [10, 19] about the existence of a whole set of procedures for decision making a priori. modeFRONTIER© implementation of MCDM allows the user to classify all the available alternatives through pair-wise comparisons on attributes and designs. Moreover, modeFRONTIER helps decision makers to verify the coherence of relationships. Thus it is an a posteriori or interactive oriented tool.

To be coherent, a set of relationships should be both rational and transitive. To be rational means that if the decision maker thinks that solution A is better than solution B, then solution B is worse than solution A. To be transitive means that if the decision maker says that A is better than B and B is better than C, then solution A should be always considered better than solution C. In this way the tool allows the correct grouping of outputs into a single utility function that is coherent with the preferences expressed by the user Four algorithms are implemented:

- **Linear MCDM** When the number of decision variables is small;
- **GA MCDM** This algorithm does not perform an exact search so it can be used even when the number of decision attributes is big;
- **Hurwicz criterion** Used for the uncertain decision problems;
- **Savage criterion** Used for the uncertain decision problems where both the decision states and their likelihoods are unknown

1.4 Response Surface Methodology (RSM)

As stated at the beginning of this survey, a model of a system is its simplest possible representation, yet bearing the most important features of the system itself. In this section the concept of *metamodel* will be introduced together with some actual applications of the concept.

the word *meta-* (from Greek $\mu\epsilon\tau\alpha$ = "beyond") is a prefix used to indicate a concept which is an abstraction from another concept. Applied to engineering or in general to data analysis and optimum search, the basic concept of *metamodeling* is to construct a simplified model of the model itself (be it known or unknown) with a moderate number of experiments and then use the approximate relationship to make predictions at additional untried inputs [20].

This process involves the choice of an experimental design, a metamodel type and its functional form for fitting, and a validation strategy to assess the metamodel fit. A typical *metamodeling* sequence for engineering design is:

1. **Model formulation** Identification and understanding of the problem. Definition of design variables, objectives, and constraints;
2. **Design selection** Definition of an initial dataset of true values (experimental data or numerical simulation), usually by means of DOE techniques;

3. **Metamodel fitting** Definition of the interpolating model;
4. **Metamodel assessment** Definition of performance criteria to characterize the fidelity of the metamodel. This is often called a *validation* process;
5. **Gaining insight** Evaluation on the metamodels errors help gaining information of the main influence of design variables on the system;
6. **Using the metamodel** Once validated, the metamodel is used to predict responses and perform virtual optimizations.

Responce surface Methodology (RSM) is a method first introduced by G. E. P. Box and K. B. Wilson in 1951. The main idea of the original methods is to use DOE techniques to create a dataset and then exploit first-degree polynomial interpolation to assess variable influence on the actual model.

RSM is a metamodeling technique to virtually explore a design space. The great advantage of RSM consist in almost instant responses, in contrast to actual physical experiments or usually high CPU time consuming numerical simulations nowadays required in engineering processes.

There are many interpolation techniques and each of which has pros and cons that must be weighted up to choose the one that gives the best performance for the particular problem considered.

1.4.1 Singular Value Decomposition (SVD) response surface

Singular Value Decomposition is one of the simplest ways to build a response surface. This response surface employs the Least Square Sum (LSS) method to estimate the regression coefficients $\beta_1,\ldots,\beta_n$ of a regression function $g(\mathbf{x},\beta)$. Mathematically, the least sum-of-squares criterion consists in minimizing:

$$Q = \sum_{i=1}^{n}[f(\mathbf{x}_i) - g(\mathbf{x}_i,\beta)]^2$$

Usually SVD tries fitting data with very simple regression functions like linear polynomials, quadratic polynomials and exponential functions.

1.4.2 K-nearest response surface

K-nearest is a statistical methods that interpolates the function value using weighted average of the known values in the k nearest points:

$$f(\mathbf{x}_0) \approx f(\mathbf{x}_0) = \sum_{i=1}^{k}\lambda_i f(\mathbf{x}_i)$$

The weights λ_i are obtained by the inverse distance and an inverse distance exponent p:

$$\lambda_i = \frac{\frac{1}{dist_i^p}}{\frac{1}{\sum_{i=1}^{k} dist_i^p}}$$

1.4.3 Kriging response surface

Kriging is a statistical tool developed for geostatistics by Matheron (1963) and named in honour of the South African mining engineer D. G. Krige.

This method uses variogram to express the spatial variation and it minimizes the error of predicted values, which are estimated by spatial distribution of the known data. In a spatial correlation metamodel the design variables are assumed to be correlated as a function of distance during prediction, hence the name spatial correlation metamodel. These metamodels are extremely flexible since the metamodel can either provide an exact interpolation of the data, or smooth the data, providing an inexact interpolation, depending on the choice of the correlation function [20]. A spatial correlation metamodel is a combination of a polynomial model plus departures of the form:

$$\phi(\mathbf{x}) = g(\mathbf{x}) + Z(\mathbf{x})$$

where $\phi(\mathbf{x})$ is the unknown function of interest $g(\mathbf{x})$ is a polynomial approximation, and $Z(\mathbf{x}$ is the realization of a normally distributed Gaussian random process with mean zero, variance σ^2 , and non-zero covariance. While $g(\mathbf{x})$ globally approximates the design space, $Z(\mathbf{x})$ creates localized deviations so that the kriging model interpolates the k sampled data points.

Kriging is associated with the acronym B.L.U.E (Best Linear Unbiased Estimator). An estimator is said to be a best linear unbiased estimator (BLUE) if:

1. it is a linear estimator (it can be expressed as a linear combination of the sample observations);

2. it is unbiased (the mean of error is 0);

3. no other linear unbiased estimator has a smaller variance.

A BLUE is not necessarily the best estimator, since there may well be some non-linear estimator with a smaller sampling variance than BLUE. In many situations, however, the efficient estimator may be so difficult to find that we have to be satisfied with the BLUE (if the BLUE can be obtained).

The major premise of kriging interpolation is that every unknown point can be estimated by the weighted sum of the known points. The matrix of the covariances of all the sample points in the search neighbourhood operates to take into account data redundancy.

Two points that are close to each other in one direction and have a high covariance are redundant. So the process takes care of the clustering of the data points.

Heat Transfer Problems

Chapter Two

Numerical representation of geometrical shapes

The problems that are going to be surveyed in the first part of this thesis deal with geometry shapes. The objectives of the studies are functions of their physical domain, whose change in form affects the behaviour of the system. In this sense, great attention is to be given to the method by which shapes are mathematically represented.

Over the last few decades *computer aided design* (CAD), *computer aided manufacturing* (CAM), and in general *computer aided engineering* (CAE) tools have been thriving. Nowadays these tools are an absolutely necessary routine in development of products, in whichever sphere of activity. Among the amount of CAE instruments, it is of most relevance a methodology for the geometrical representation of the models to be developed.

Geometric entities can be divided into two categories:*definable* entities, the ones from classical geometry (lines, planes, conics) and whose shape is analytically defined, and *non-definable* entities, that is complex shapes that do not posses a straightforward mathematical formulation. the most of the form that gives birth to industrial products is of the second kind. Hence the necessity has aroused to create and codify appropriate methods to represent this kind of shapes.

Spline curves are likely to be among the most used in naval architecture. They are the numerical representation of a tool commonly used in drafting: a long, thin, flexible strip of wood, plastic, or metal held in place at defined control points. The elasticity of the strip combined with the constraint applied would cause the strip to take the smoothest possible shape. Spline curves are an evolution of simple linear interpolation, and belong to the category named *interpolating methods*. Yet there's another family of methods to define a curve from a set of given points: *approximating methods*. In seeking a method for describing curves, a key target is the achievement of the desired goal (a certain shape) with the least possible information. Linear interpolation, for example, may lead to good representations, provided the number of points be huge enough to guarantee accuracy. Depending on the curvature complexity, the number of points may increase cumbersomely, with a great amount of data to be managed. Nevertheless, problems originate, linked to continuity and derivativeness issues. Among the approximating methods *Bézier curves* were born in automotive area to easily represet cars shapes, while *B-spline curves* are a further evolution that due to the reduced influence of their *basis functions*[1] allow a local handling of curves shapes. NURBS (*Non Uniform Rational B-Splines*) are a standard in industrial drafting, and they have spread even in digital animation. Their wide use lies on a series of reasons:

- a common mathematical formulation for both basic analytical forms (conics) and freeform curves;
- high flexibility in drawing an ample variety of profiles;
- fast computations, by means of stable algorithms;
- invariance to affine transformations.

Mathematical representation of curves (the extension to surfaces is straightforward) might be either *non-parametric*, where in its most general implicit form can be expressed as $f(x, y) = 0$ or *parametric*, where the curve $\mathbf{C}$ is given in vectorial notation as a function of a parameter t:

$$\mathbf{C}(t) = [x(t), y(t)] \tag{2.1}$$

The parametric representation has the following features:

- the shape of the curve only depends from the mutual relationship of control points;
- curve with infinite slope is easily described, because the derivative in the parameter is always defined;
- geometrical entities can be easily represented in computer graphics.

In this thesis both Bézier and NURBS curves and surfaces have been used as numerical representation of shapes. In this chapter a short introduction to both entities is outlined.

2.1 Bézier Curves

Bézier curves allow an univocal approximation of a given set of points, and are expressed by the following parametric equation:

$$\mathbf{C}(t) = \sum_{i=0}^{q} \mathbf{p}_i f_i(t) \qquad t \ \epsilon \ [0, 1] \tag{2.2}$$

[1] Basis function will be introduced and defined hereafter in this chapter

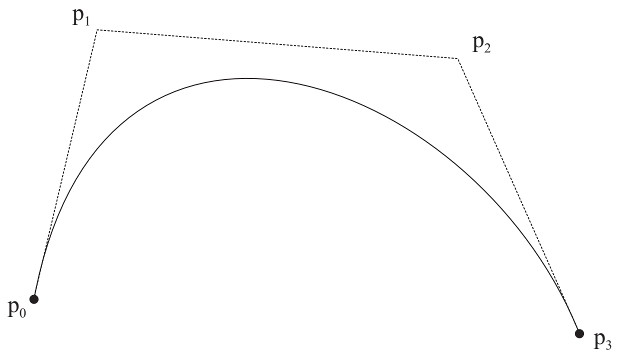


Figure 2.1 third order Bézier curve

where $\mathbf{p}_i$ is the i-th control point that defines the curve. The curve is the result of a weighted linear combination of the points, the weights varying as a function of t. Joined together, the control points form the *control polygon*, as sketched in figure 2.1 for a four points curve. The functions $f_i(t)$ have to obey a series of properties [21, 22]:

1. the $f_i(t)$ must interpolate the first and last point of the control polygon. That is, the curve begin in $\mathbf{P}_0$ and ends in $\mathbf{P}_q$, being $q + 1$ the number of points;
2. the tangents to the curve at the first and last point must have the same direction of the sides of the control polygon;
3. the $f_i(t)$ have to be symmetric with respect to the interval of definition for the parameter t. It means that an inversion in the ordering of the points does not affect the curve shape, but just the direction in which it is transversed.

these conditions are fulfilled by a family of functions, called *Bernstein Polynomials*, denoted as $B_{i,q}(t)$. Thus equation 2.2 becomes:

$$\mathbf{C}(t) = \sum_{i=0}^{q} \mathbf{p}_i B_{i,q}(t) \qquad t \in [0, 1] \tag{2.3}$$

where

$$B_{i,q}(t) = \frac{q!}{i!(q-i)!} t^i (1-t)^{q-i} \tag{2.4}$$

The degree of the polynomials approximating a curve depends on the number of control points. Precisely, when the number of points id $q + 1$ the polynomial degree is q. Without lack of generality, in this survey third order Bézier curves will be considered, as they are the ones available in FEMLAB/COMSOL©, the software used in this thesis as numerical solver. When $q = 3$ the parametric equation states as follows:

$$\mathbf{C}(t) = (1-t)^3\mathbf{p_0} + 3t(1-t)^2\mathbf{p_1} + 3t^2(1-t)\mathbf{p_2} + t^3\mathbf{p_3} \tag{2.5}$$

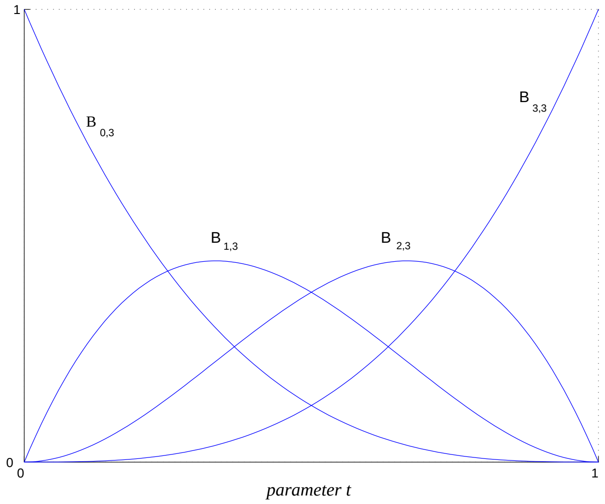


Figure 2.2 basis functions

Figure 2.2 shows the trend of the Bernstein basis functions. At the extremes of the parametric interval the weight given to the associated points is equal to 1, that leads to the interpolation of the first and last points.

2.1.1 Bézier curves derivatives

The derivative of a q-th degree Bernstein polynomial cab be written as:

$$\frac{\mathrm{d}}{\mathrm{d}t} B_{i,q}(t) = q \left[B_{i-1,q-1}(t) - B_{i,q-1}(t) \right] \tag{2.6}$$

hence, the first derivative of a Bézier curve becomes:

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{C}(t) = q \sum_{i=0}^{q} \left[B_{i-1,q-1}(t) - B_{i,q-1}(t) \right] \mathbf{p}_i \tag{2.7}$$

that after rearranging yields to :

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{C}(t) = q \sum_{i=0}^{q-1} \Delta \mathbf{p}_i B_{i,q-1}(t) \tag{2.8}$$

where:

$$\Delta \mathbf{p}_i = \mathbf{p}_{i+1} - \mathbf{p}_i \tag{2.9}$$

In figure 2.3 a third degree Bézier curve and its derivative are sketched.

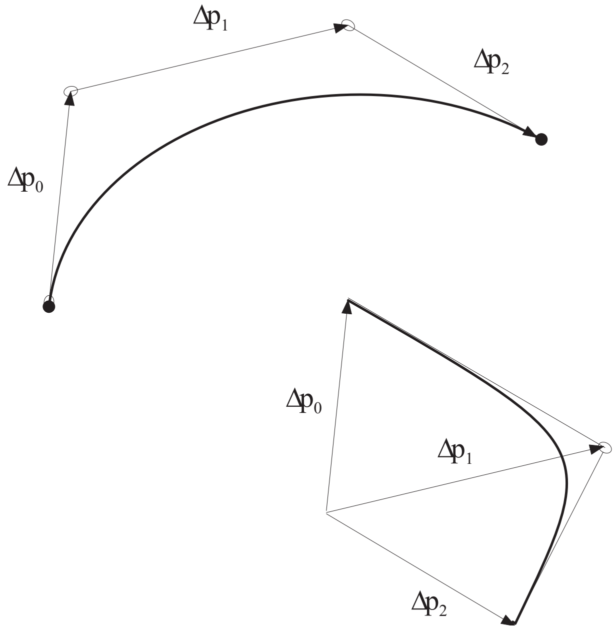


Figure 2.3 a third order Bézier curve and its derivative scaled of a factor $q = 3$

2.1.2 The Matrix form of Bézier curve

Some authors ([21]) prefer to write Bézier curves in matrix form. To derive it, equation 2.5 can be rewritten as:

$$\mathbf{C}(t) = [(1-t)^3,\ \ 3t(1-t)^2,\ \ 3t^2(1-t),\ \ t^3]\,[\mathbf{p_0},\ \ \mathbf{p_1},\ \ \mathbf{p_2},\ \ \mathbf{p_3}]^T \tag{2.10}$$

that expressed in a more compact and general form becomes:

$$\mathbf{C}(t) = \mathbf{B}_q\mathbf{P}_q \tag{2.11}$$

where the matrix of the basis functions $\mathbf{B}_q$ and that of the coefficient of the curve $\mathbf{P}_q$ are highlighted. Equation 2.10 can be rewritten as follows:

$$\begin{aligned}\mathbf{C}(t) \quad = \quad & [(1-3t+3t^2-t^3)\ \ (3t-6t^2+3t^3)\ldots \\ & \qquad\qquad \ldots(3t^2-3t^3)\ \ t^3]\,[\mathbf{p_0}\ \ \mathbf{p_1}\ \ \mathbf{p_2}\ \ \mathbf{p_3}]^T\end{aligned} \tag{2.12}$$

With a step further it is possible to delineate the expression:

$$\mathbf{C}(t) = [t^3\ t^2\ t\ 1]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \mathbf{p_0} \\ \mathbf{p_1} \\ \mathbf{p_2} \\ \mathbf{p_3} \end{bmatrix} \tag{2.13}$$

that in a compact way becomes, for a generic degree q:

$$\mathbf{C}(t) = \mathbf{T}\,\mathbf{M_q}\mathbf{P_q} \tag{2.14}$$

The expressions for the first derivatives are straightforwardly obtained reducing by derivation the degree of the monomial vector $\mathbf{T}$, and eliminating the last row in matrix $\mathbf{M}$:

$$\mathbf{C}^{'}(t) = 3 \cdot [t^2\ t\ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -4 & 2 & 0 \\ -1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p_0} \\ \mathbf{p_1} \\ \mathbf{p_2} \\ \mathbf{p_3} \end{bmatrix} \tag{2.15}$$

In a recursive way, the second derivatives are:

$$\mathbf{C}^{''}(t) = 6 \cdot [t\ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 1 & -2 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p_0} \\ \mathbf{p_1} \\ \mathbf{p_2} \\ \mathbf{p_3} \end{bmatrix} \tag{2.16}$$

The matrix form 2.14 does not describe an actual Bézier curve. It is rather the monomial form of the curve, which is *numerically unstable* [22, 23, 24, 25] due ti unavoidable inaccuracies with the use of finite precision computers. Thus the monomial form should be avoided where accuracy is of any importance.

2.1.3 Composite Bézier curves

Multiple Bézier curves may be joined so as to generate shapes that are too complex for a single curve. In combining two curves together, the smoothness of the resulting curve has to be somehow controlled. Let $\mathbf{p_0}, \ldots, \mathbf{p_3}$ and $\mathbf{p_3}, \ldots, \mathbf{p_6}$ be the control points of two cubic (third order) Bézier curves, as in figure 2.4. Since they share point $\mathbf{p_3}$, they clearly form a continuous curve. Yet with this minimal requirement sharp corners are allowed. From what is known about endpoint conditions, the tangents bear the same direction of the control polygon sides. Thus the collinearity of points $\mathbf{p_2}, \mathbf{p_3}, \mathbf{p_4}$ ensures the so called *first order geometrical continuity*, G^1. For parametric curves, the concept of first order geometrical continuity differs from the condition of *differentiability*, C^1, The second being a much stronger constraint. C^1 implies G^1, but the relation is not bi-univocal. From equation 2.8 comes that a differentiable parametric curve is obtained when the following condition is satisfied:

$$\mathbf{p_2} - \mathbf{p_3} = \mathbf{p_3} - \mathbf{p_4} \tag{2.17}$$

. The continuity of a composite curve is a function of the chosen parametrization rather than of the actual geometry. A certain parametrization describes a definite shape, but a chosen shape can be drawn by different parametric curves. The distinction between C^1 and G^1 might look trivial at a first sight, but in fact it bears great implications on the

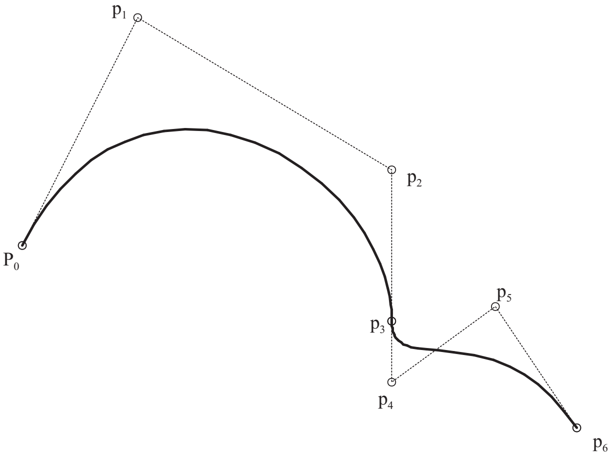


Figure 2.4 two joined third order Bézier curves

analytical management of the geometrical entities. As an example, a CNC machine works on a piece by means of parametric functions, whose discontinuity may generate errors in movements and accelerations, even if the shapes to be made are smooth.

2.2 Bézier Patches

Bèzier patches are the natural three-dimensional extension of Bézier curves. Keeping in mind the definition of a Bézier curve, eq 2.3, surfaces are univocally determined by a *control polyhedron*, analogous to the 2D control polygon. The parametric formulation defining the surface is a bi-polynomial form. Thus there are two parameters, t and s, which govern the shape of the patch. The general equation for a curve in a Bézier-Bernstein notation is

$$\mathbf{C}(t,s) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{p}_{i,j} B_{i,q}(t)\, B_{j,r}(s) \qquad t, s \in [0,1] \tag{2.18}$$

Functions $B_{i,q}(t)$ and $B_{j,r}(s)$ are made up as in 2.4 and the points $\mathbf{p}_{i,j}$ are the vertexes of the control polyhedron. The polynomial degrees q and r can be different for the two parametric direction t and s. As for the curves case, some properties of Bézier patches are going to be patches, limiting the treatment to the case of bi-cubic (third order polynomials in both direction) surfaces. The set of points composing the control polyhedron is a 4 × 4 matrix. Even for patches a matrix notation can be derived:

$$\mathbf{C}(t,s) = [(1-t)^3 \;\; 3t(1-t)^2 \;\; 3t^2(1-t) \;\; t^3]\, \mathbf{P} \begin{bmatrix} (1-s)^3 \\ 3s(1-s)^2 \\ 3s^2(1-s) \\ s^3 \end{bmatrix} \tag{2.19}$$

that performing a basis transformation as in eq. 2.13 reaches the form:

$$\mathbf{C}(t,s) = \mathbf{T}\,\mathbf{M}^T\,\mathbf{P}\,\mathbf{N}\,\mathbf{S} \tag{2.20}$$

where for $q = r = 3$

$$\mathbf{M} = \mathbf{N} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.21}$$

and **P**, sometimes called the *geometry matrix* has the form:

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_{11} & \mathbf{p}_{12} & \mathbf{p}_{13} & \mathbf{p}_{14} \\ \mathbf{p}_{21} & \mathbf{p}_{22} & \mathbf{p}_{23} & \mathbf{p}_{24} \\ \mathbf{p}_{31} & \mathbf{p}_{32} & \mathbf{p}_{33} & \mathbf{p}_{34} \\ \mathbf{p}_{41} & \mathbf{p}_{42} & \mathbf{p}_{43} & \mathbf{p}_{44} \end{bmatrix} \tag{2.22}$$

The distribution of the points on the matrix can be imagined as a bi-dimensional projection of the patch. In general, only angular points ($\mathbf{p}_{11}$, $\mathbf{p}_{14}$, $\mathbf{p}_{41}$, and $\mathbf{p}_{44}$) lie on the surface. Points $\mathbf{p}_{12}$, $\mathbf{p}_{13}$, $\mathbf{p}_{24}$, $\mathbf{p}_{34}$ and their reciprocal define the Bézier curves that bound the surfaces. the remaining points control the shape inside the patch. See figure 2.5

When using composite patches, in order to reach a smooth surface, in [26] the authors suggest the following condition:

$$\begin{aligned} \frac{\partial \mathbf{S}_1}{\partial s}(1,0) &= \frac{\partial \mathbf{S}_2}{\partial s}(0,0) \\ \frac{\partial \mathbf{S}_1}{\partial s}(1,1) &= \frac{\partial \mathbf{S}_2}{\partial s}(0,1) \end{aligned} \tag{2.23}$$

where $\mathbf{S}_1$ and $\mathbf{S}_2$ are two flanked surfaces , joined on sides of parametric coordinate t as in figure 2.6. The derivatives in the joined side direction depend only on the side points. Thus higher order continuity requirements are naturally satisfied by the equipollence of the side points. On the other hand, condition 2.23 only states the equivalence of the derivatives in the perpendicular direction to the joined side (also called *cross-boundary derivatives*) at the extremes of s interval.The condition given in [26] gives no information about the existence of a tangent plane along the curve $\mathbf{S}_1(t,1) = \mathbf{S}_2(t,0)$. For a complete and in-depth review on the subject, the reader is

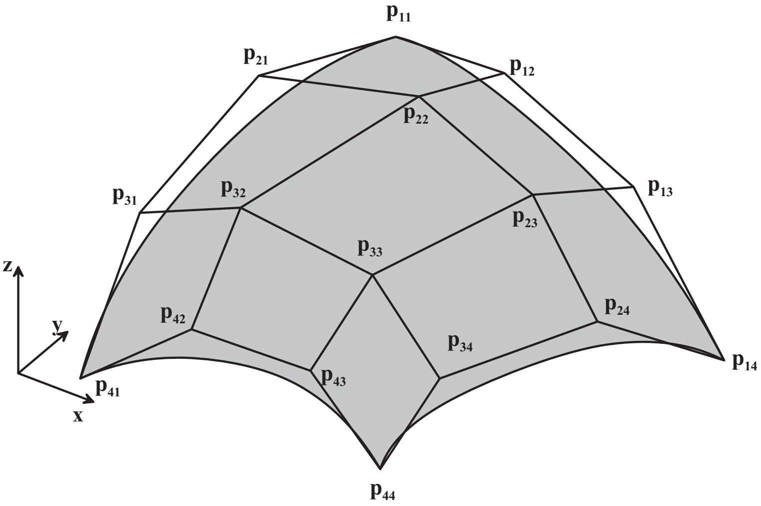


Figure 2.5 Bi-cubic Bézier patch

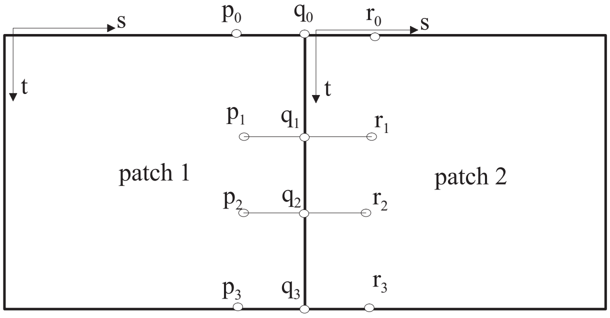


Figure 2.6 two flanked surfaces

referred to [22]. For the purposes of this thesis, where composite surfaces are used to model 3D geometries, it is sufficient to bear in mind it is possible to demonstrate, similarly to what has been done for curves, that the cross-boundary derivatives only depend on two rows of points. As evenness condition, the collinearity of points with the same index in t direction is imposed, as clearly shown in figure 2.6. This does not ensure the existence of a tangential plane, but grants a sufficiently smooth junction.

2.3 NURBS curves

NURBS (*Non-Uniform Rational B-spline*) curves consist of many polynomial pieces, offering much more versatility than Bézier curves do. The general formulation of a NURB follows the one given for Bézier curves:

$$\mathbf{C}(t) = \sum_{i=0}^{n} R_{i,q}(t)\mathbf{p}_i \tag{2.24}$$

where n is the number of control points $\mathbf{p}$, q is the order of the polynomial describing the curve and $R_{i,q}$ are the *basis-functions* that mix the influence of the different points, as an averaged sum. The NURBS basis-functions are obtained from eq. (2.25), introducing the concept of weight w_i of a single point:

$$R_{i,q}(t) = \frac{N_{i,q}(t)w_i}{\sum_{j=0}^{n} N_{j,q}(t)w_j} \tag{2.25}$$

In eq. (2.25) $N_{i,q}$ are obtained by means of the Cox-De Boor algorithm [27]:

$$N_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2.26a}$$

$$N_{i,q}(t) = \frac{(t - t_i)N_{i,q-1}(t)}{t_{i+q-1} - t_i} + \frac{(t_{i+q} - t)N_{i+1,q-1}(t)}{t_{i+q} - t_{i+1}} \tag{2.26b}$$

where t_i are the so-called *knots*. The *knots* are organized in an array:

$$\mathbf{T} = [\underbrace{0, ..., 0}_{q+1}, t_{q+1}, ..., t_{m-q-1}, \underbrace{1, ..., 1}_{q+1}] \tag{2.27}$$

The *knots* are normalized values that point out the basis-functions intervals of interest. The relationship between number of *knots* m, number of control points n and order q is:

$$m = n + q + 1 \tag{2.28}$$

This means that using NURBS with a fixed order we can approximate any number of control points simply choosing the right dimension of the *knots* array. An example of

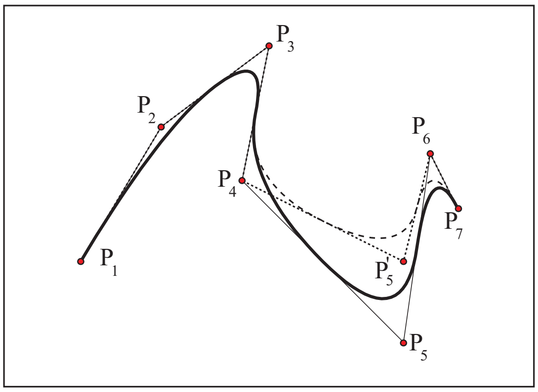


Figure 2.7 Curve, control polygon, and *Local-support*

a NURBS and its basis-functions are depicted respectively in figure 2.7 and figure 2.8. In figure 2.8 the condensation of basis-functions at the extremes of the parameter range makes the curve able to interpolate the first and the last control points. In figure 2.8 is clear that each basis-function is non-zero in a number of intervals at most equal to the order of the curve. This introduces the concept of *local support*: a change in the control point $\mathbf{P}_i$ or in the weight w_i only affects the curve in the parameter range $t \in \left[t_i, t_{i+q}\right[$. This ensures that editing a given point would only modify the shape in its neighborhood and not globally. Therefore, whatever complex the shape of the geometric entity may be, at any point, it can be represented by a piecewise low degree polynomial and using a unique curve definition. The control points are joined by the *control polygon* which roughly approximates NURBS pattern, as in the case of Bézier curves.

The presence of weights, w_i, manages to draw the curve near the control points, increasing the possibility of ruling its shape. Besides, the weights allow representing conic forms, which requires rational polynomial function for their description.

The continuity is of C^∞ class along the parameter t range, apart at the knots where it is of $C^{(q-r+1)}$, being r the multiplicity of the knot [27].

2.3.1 Periodic NURBS

In chapter 3 NURBS are used to describe compact heat exchanger channels. The channels have to show a periodic smooth pattern, yet for computational cost reasons a single period has to be represented. In order to obtain a periodic curve using NURBS form, it is possible to repeat a single NURBS module period after period. The periodicity is imposed by shifting a copy of the original $\mathbf{P}_i$ vector of a module-length quantity, and by repeating the knots interval as many times as required.

As stated in the previous section, the continuity class of a NURBS is $C^{(q-r+1)}$, being

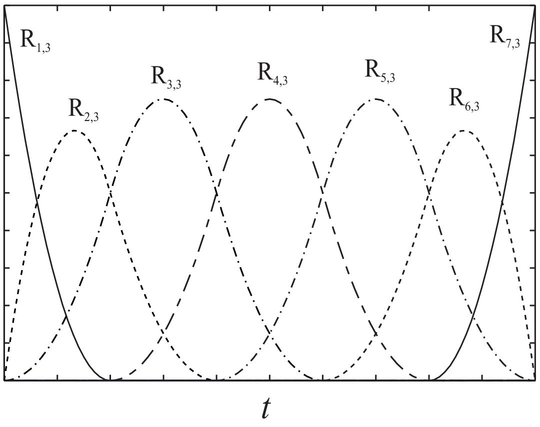


Figure 2.8 Basis-functions for a single curve

the multiplicity of the knot at the extremes $r = q + 1$. The need arise for a method to control the smoothness of a NURBS at the endpoints, where the single module repeats itself. The continuity is guaranteed by eliminating knots multiplicity at the extremes of the basic interval. In figure 2.9, a 9 control point NURBS cubic periodic curve is sketched, while figure 2.10 shows the basis functions of the curve without knot multiplicity. Along the periodic curve, basis functions D-E-F of module i, are coincident to basis functions A-B-C of module $i + 1$. By this process, the extreme points of a single module are no more interpolated, but the curve gains the desired continuity class.

2.3.2 2D NURBS-Bézier Conversion

FEMLAB/COMSOL©, does not support the use of NURBS, but only third degree Bézier curves. In chapter 3, with the aim to test an efficient optimization procedure, local modifications of the shape have been considered of importance. Thus a MATLAB© code has been implemented to create NURBS, following the *Cox-Be Boor recurrence relation* [27], eq. 2.26. Being NURBS rational piece-wise polynomial forms, they are capable of correctly approximate conics. However, for the use is going to be done of them, there is no need of this characteristic. Moreover, in evolutionary optimization processes, the higher the number of variables, the slower the convergence rate. For this reason, the weighting functions in eq. 2.25 are taken of a constant unitary value. Moreover, for the same reason the *knots* distribution is set constantly uniform, leaving to the sole points distribution the definition of the curve. This reduces the NURBS curves to a subset of theirs known as *B-Spline* curves, nevertheless causing any lack of generality to the work done. To draw shapes in FEMLAB/COMSOL©, a MATLAB© conversion routine is used to shift from NURBS to composite Bézier

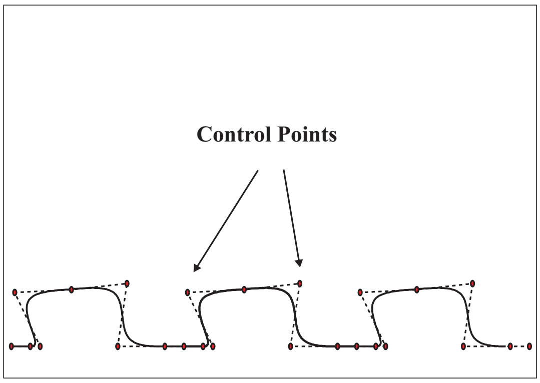


Figure 2.9 Periodic curve

form. Particularly, third order piece-wise polynomial NURBS are converted into third order Bézier. The necessary number of Bézier curves to cover the whole NURBS is a function of the knots span. The number of knots intervals correspond to the number of Bézier curves. A possible way to build composite Bézier from NURBS curves is to proceed as in the following.

1. Every knot interval $t \in [t_k, t_{k+1}]$ is associated to a normalized $t^* \in [0, 1]$ one;

2. As the endpoints of a Bézier curve interpolate its control polygon, the first and last control points are calculated, for each knot interval by $\mathbf{C}_k^B(0) = \mathbf{C}(t_k)$ and $\mathbf{C}_k^B(1) = \mathbf{C}(t_{k+1})$, being $\mathbf{C}_k^B(0)$ the k-th Bézier curve;

3. As two further points are needed, two transit conditions are imposed by choosing two appropriate values t_1 and t_2 in the interval $[t_k, t_{k+1}]$.

4. Once obtained a closed linear system, it is solved to obtain the desired Bézier points:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ B_{0,3}(t_1^*) & B_{1,3}(t_1^*) & B_{2,3}(t_1^*) & B_{3,3}(t_1^*) \\ B_{0,3}(t_2^*) & B_{1,3}(t_2^*) & B_{2,3}(t_2^*) & B_{3,3}(t_2^*) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1^{B,k} \\ \mathbf{p}_2^{B,k} \\ \mathbf{p}_3^{B,k} \\ \mathbf{p}_4^{B,k} \end{bmatrix} = \begin{bmatrix} \mathbf{C}(t_k) \\ \mathbf{C}(t_1) \\ \mathbf{C}(t_2) \\ \mathbf{C}(t_{k+1}) \end{bmatrix}$$

   where $\mathbf{p}_i^{B,k}$ is the i-th control point of the k-th Bézier curve.

The result of this operation is shown on a non periodic curve in figure 2.11 where, following the arrow, NURBS representation can be shifted on a Bézier representation of the same curve. It's evident the increased number of control points in the Bézier case. Besides the local support given by its piece-wise polynomial definition, a NURBS representation bears the intrinsic properties of C^∞ continuity. This property has to be imposed to Bézier curves by appropriate relation between control points.

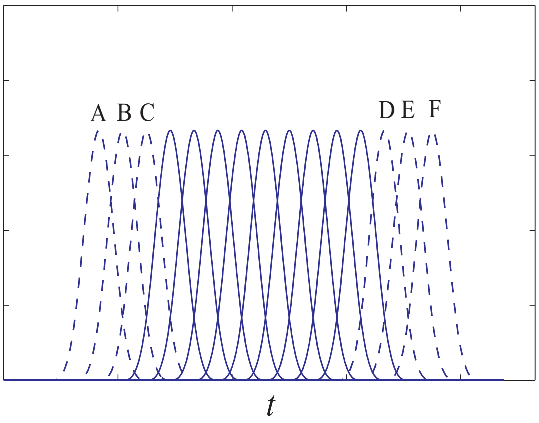


Figure 2.10 Basis-functions for a periodic curve

2.4 NURBS surfaces

As for Bézier patches , NURBS surfaces are the three dimensional extension of curves. Even in this case there exist a control polyhedron that the surface approximates. The surfaces are bi-polynomial forms, thus there are two parameters t and s transversing the surface in orthogonal directions. Control points form a bi-dimensional array $n \times m$ where n is the number of control points associated to t and m is the one associated to s. for both the direction equation eqn:NURBSref is valid. The equation defining a NURBS surface as a function of t and s is:

$$\mathbf{S}(t, s) = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,j}(t, s)\mathbf{p}_{i,j} \tag{2.29}$$

$$R_{i,j}(t, s) = \frac{N_{i,p}(t)N_{j,q}(s)w_{i,j}}{\sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} \tag{2.30}$$

where p and q are the degree of the polynomial basis-functions $N_{i,p}$ and $N_{j,q}$, obtained in a recursive way as in equation 2.26, and i, j are the weighting functions. In 3D the number of parameter to define a surface drastically increase related to the 2D case, making an evolutionary optimization problem cumbersome. Even more in this case, reducing the number of variables is important. So, as for the two dimensional case, in chapter 3 unitary weighting functions and uniform knots interval are used, thus limiting the NURBS surfaces to B-Spline. Both polynomial degree and control points number can vary in the two parametric directions, which makes NURBS a quite versatile geometrical tool.

In figure 2.12 an example of a 3D NURBS is depicted. It is defined by a 4 × 4 matrix of control points, linear along parametric direction t (geometrical direction x

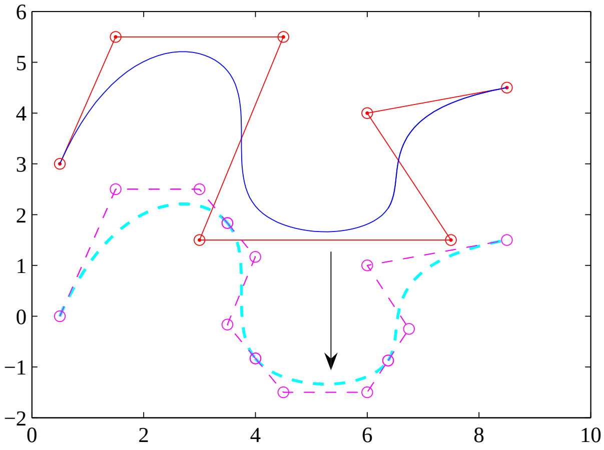


Figure 2.11 example of conversion from NURBS to Bézier

in figure), and quadratic in parametric direction s (geometrical direction y in figure).

Periodic surfaces can be obtain eliminating the knot multiplicity at the endpoints and shifting a copy of the control points of a module-length quantity, exactly in the same way it has been done for 2D NURBS. Figure 2.13 shows an example of a NURBS surface, periodic in t direction (geometrical direction y in figure).

2.4.1 2D NURBS-Bézier Conversion

The objective is do develop a NURBS surface in a series of Bézier patches, likewise in the 2D case. Bézier curves and surfaces are not piece-wise polynomial as NURBS are. Their basis functions are defined over the entire parametric interval. As the degree of the NURBS polynomial defines the number of non null basis function in each knot interval, in order to get an equivalent degree transformation, a Bézier surface must be obtained for each knots span. Figure 2.14 highlights an example of 3D NURBS-Bézier conversion. The NURBS surface is piece-wise cubic and periodic in both direction. It is obtained with a 10×4 matrix of control points. It is to be noted that for complex enough surfaces, the number of Bézier patches awfully increases.

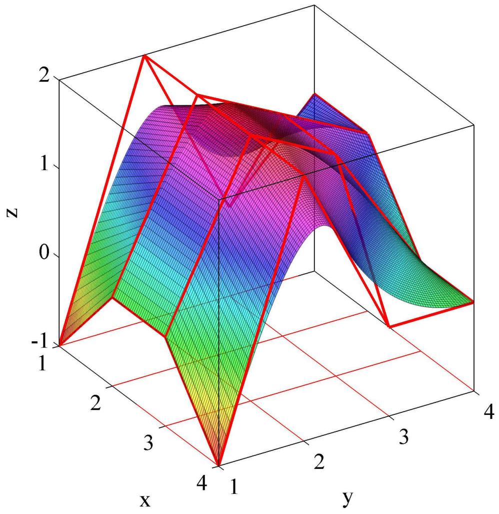


Figure 2.12 Example of a NURBS surface

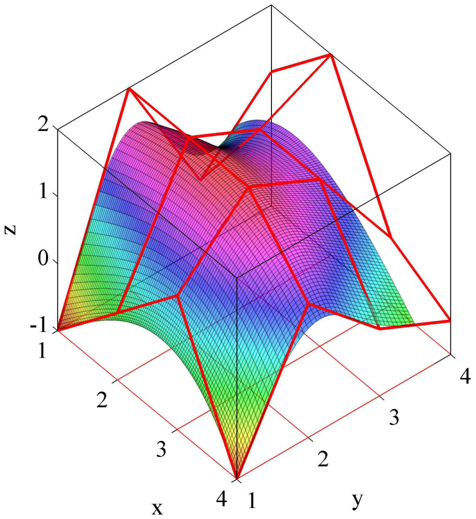


Figure 2.13 NURBS periodic along direction y

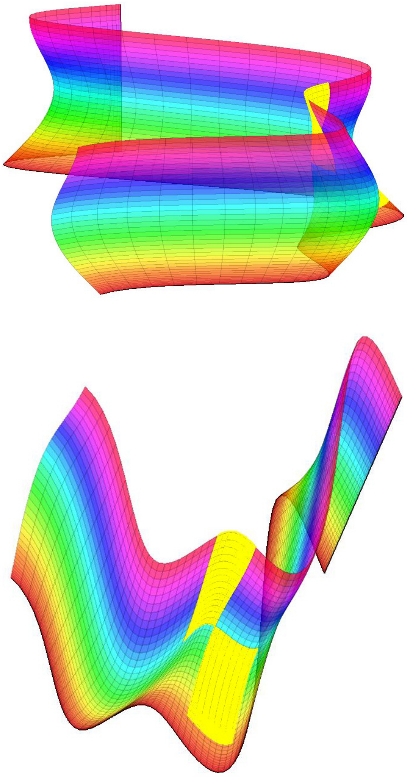


Figure 2.14 Periodic NURBS surface: yellow wireframes highlight two Bézier patches covering the NURBS

Chapter Three

Convective wavy channels optimization

Convective wavy channels represent the building block of an ample variety of heat exchangers. From an engineering point of view a desired target is to modify the shape of the channels in order to maximize their heat transfer rate, without an excessive penalty in their pressure losses.

In this chapter it is described how this has been achieved by coupling FEMLAB, a general-purpose unstructured finite element solver, with modeFRONTIER, a multi-objective optimization system. It is now widely recognized that the recent progress in the performance of computing hardware, and the availability of ready-to-use sophisticated numerical packages, has increased the role of Computer Aided Engineering (CAE) in the engineering design practice. As pointed out e.g. by Farouki [28], the major difficulty in automating a CAE optimization procedure, is the proper linking of the various stages of the design process, namely model creation by means of a CAD tool, mesh generation, setting of properties and boundary conditions, numerical analysis and performance evaluation, and finally being able to optimize the shape, or other functional parameters, of the system.

In this case FEMLAB, a general unstructured FE solver, takes care of solving the forced convective problem defined in the domain, while modeFRONTIER, a powerful and flexible optimization tool, exploits FEMLAB engine to run the optimization task.

This process is a truly multi-objective one, since it is desired, from a design point of view, to maximize the heat transfer rate, in order to e.g. reduce the volume of the equipment, and to minimize the friction losses, which are proportional to the pumping power required. These two goals are clearly conflicting, as it is well known that the increase in heat transfer rate is accompanied by an even larger increase of friction losses. Therefore, there is no a single optimum to be found, and for this reason a MOGA (Multi-Objective Genetic Algorithm) is used together with the so-called Pareto's dominance, which allows to obtain a design set, rather than a single configuration. The geometry of the channel is parametrized by means of NURBS (Non-Uniform Rational B-Splines), and their control points represent the design variables.

An alternative, and simpler, parametrization is obtained by means of piecewise-linear profiles and compared with the smooth one generated by NURBS representation.

3.1 Problem statement

The problem described in this chapter is the multi-objective optimization of two-dimensional (2D) and three-dimensional (3D) convective wavy channels, which represent the fundamental building block of an ample set of heat exchanger and heat transfer devices. The study is limited to a single channel at fully developed flow and heat transfer conditions. In such a circumstance, channels of periodic cross section form can be considered periodic in the flow and thermal fields, as well. Therefore the computational domain of interest becomes a single periodic module of the entire geometry, as depicted in figure 3.1 and 3.2 for a two-dimensional and three-dimensional channel respectively. In [29], Patankar and Liu describe a general way to consider fully developed regime, giving periodic boundary conditions for the velocity, pressure and thermal fields. In this works the same conditions expressed in [29] for the

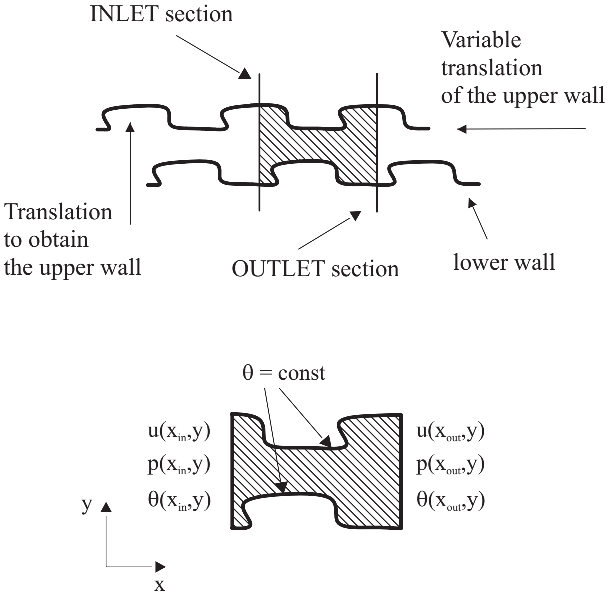


Figure 3.1 Repeating module of a two-dimensional wavy channel.

fluid-dynamic problem have been used, while for the thermal field a slightly different approach has been introduced. It is detailed later in this chapter.

The study has been limited to the steady, laminar flow regime both because it is found in many practical circumstances and because this leads to low computational cost, while still guaranteeing a large number of design variables. The Reynolds number value chosen for the simulation is $Re = 200$, while the Prandtl number is assumed $Pr = 0.7$, representative of air.

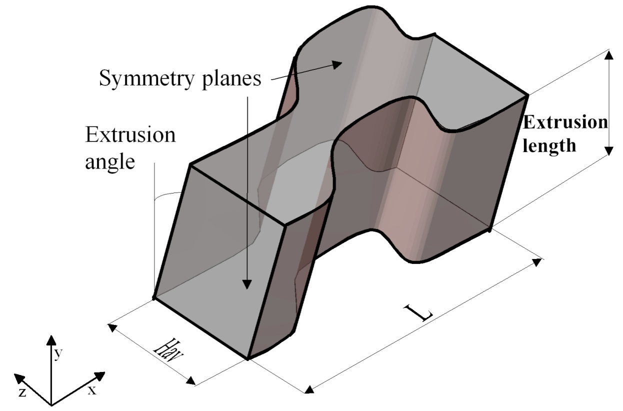


Figure 3.2 Repeating module of a three-dimensional wavy channel.

3.1.1 Dimensional analysis

In achieving this optimization task, dimensional analysis has been used for similitude pourposes, in order to render a more general treatment of the subject.

Taking into account an incompressible fluid in forced convection regime, the phenomenon of convective heat transfer is dominated by six independent variables as to be represented by the following relation:

$$f(L, \mathbf{u}, \rho, \mu, \alpha, \Delta\, T) = 0 \qquad (3.1)$$

where the symbols stand for respectively a length scale, a velocity reference, density, dynamic viscosity, thermal diffusivity, a reference temperature difference. According to Buckingam Π-theorem, as there are four fundamental units (Metre, Kilogram, Second, and Temperature), four variables are chosen as scale factors yielding to two dimensionless groups that lead the phenomenon. the following four scaling factors are chosen:

Length The geometrical domain is scaled by a reference dimension chosen as the mean hydraulic diameter of the channel , where the hydraulic diameter, D_h, is defined as

$$D_h = \frac{4A_c}{\mathrm{C}} \tag{3.2}$$

where A_c is the cross sectional area and C is the wetted perimeter of the cross-section. In the case of a 2D model, a channel of mean height $H_a v$ is considered indefinitely long in the z direction, thus the hydraulic diameter becomes:

$$D_h = \lim_{z \to \infty} \frac{4zH_{av}}{2(z+H)_{av}} = 2H_{av} \tag{3.3}$$

Velocity the velocity field $\mathbf{u} = (u, v, w)$ (where w is everywhere equal to zero in 2D) is scaled respect to the value of the mean velocity, U_{av} in the channel.

Density the density of the fluid is chosen to scale the mass dependant quantities.

Temperature Named T_w the temperature at the walls of the channel, and $T_{b,in}$ the bulk temperature at the inlet section of the channel, the dimensionless temperature, θ id defined as:

$$\theta = \frac{T - T_{wall}}{T_{b,in} - T_{wall}} \tag{3.4}$$

The bulk temperature or *mixing cup temperature* [30] is defined as an enthalpic balance at a generic section of the channel as:

$$T_{b,x} = \frac{1}{A_{c,x}\,u_{av}} \int_{A_{c,x}} T\,(\mathbf{u} \cdot \mathbf{n}) \mathrm{d}A_c, x \tag{3.5}$$

where all the terms are evaluated at the section of generic streamwise coordinate x.

In the wake of this choice, the normalized equations of conservation present two dimensionless groups which lead fluid flow and heat transfer:

Reynolds number This group represents the ratio of inertial forces ($U\rho$) to viscous forces (μ/D_h) and consequently it quantifies the relative importance of these two types of forces, and the consequent kind of flow regime (laminar, transient, or turbulent). In fluid dynamic it is used to as a criterion for determining dynamic similitude in similar geometries with possibly different types of fluid or flow rates.

$$Re = \frac{\rho\,U_{av}\,D_h}{\mu}$$

Prandtl number It stands for the ratio between momentum diffusivity, expressed by kinematic viscosity $\nu = \mu/\rho$, and thermal diffusivity $\alpha = k/\rho c_p$. Prandtl number states the relative effectiveness of convective over conductive heat transfer:

$$Pr = \frac{\mu\, c_p}{k} = \frac{\nu}{\alpha}$$

In incompressible fluids the pressure is loosely coupled to the flow field in the sense that it is not present in the continuity equation, and its role is to enforce incompressibility in the momentum equation. Nevertheless, pressure is to be scaled, and the kinetic energy density term, $(\rho \cdot U_{av}^2)$, is used. In the end all the primitive dimensional variables get transformed, and dimensionless groups are introduced, as summed up in table 3.1.

Table 3.1 dimensionless quantities

variable	Expression
x	x^*/D_h
y	y^*/D_h
z	z^*/D_h
u	u^*/U_{av}
v	v^*/U_{av}
w	w^*/U_{av}
p	$p^*/(\rho\, U_{av}^2)$
θ	$(T(x,y) - T_w)/(T_{b,in} - T_w)$
Re	$(\rho\, U_{av}\, D_h)/\mu$
Pr	$(\mu\, c_p)/k$

3.1.2 Governing equations

In this section the dimensionless form of the continuity, navier-stokes, and energy equation will be merely exposed for the simplifying conditions assumed in this work, as derivation from a general formulation is pure exercise and can be found in the majority of basic books on fluid-dynamics. The flow field is assumed to be incompressible, laminar, stationary, of constant thermophysical properties, and dynamically and thermally fully developed. Thus any time dependant term is simplified from the equations, as well as any term related to density variation. In this conditions, the following set of equations arise:

$$\nabla \cdot \mathbf{u} = 0 \tag{3.6}$$

$$\mathbf{u} \cdot \nabla \mathbf{u} = \frac{1}{Re} \nabla^2 \mathbf{u} - \nabla p \tag{3.7}$$

$$\mathbf{u} \cdot \nabla \theta = \frac{1}{Re\, Pr} \nabla^2 \theta \tag{3.8}$$

where the velocity vector **u** corresponds to (u, v, w), and that can be rewritten in scalar notation for the 3D case:

mass conservation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{3.9}$$

momentum conservation - Navier-Stokes

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) - \frac{\partial p}{\partial x} \tag{3.10a}$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) - \frac{\partial p}{\partial y} \tag{3.10b}$$

$$u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = \frac{1}{Re}\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) - \frac{\partial p}{\partial z} \tag{3.10c}$$

energy conservation

$$u\frac{\partial \theta}{\partial x} + v\frac{\partial \theta}{\partial y} + w\frac{\partial \theta}{\partial z} = \frac{1}{RePr}\left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} + \frac{\partial^2 \theta}{\partial z^2}\right) \tag{3.11}$$

The equations for the 2D case can be straightforwardly obtained, taking into account that the velocity component w is everywhere equal to zero, as well as the derivatives respect to z direction.

Before posing boundary conditions, it is worthy to focus on the meaning of the pressure term along the channel line. Neglecting any discussion about the entrance and exit regions of the channel, in the fully developed regime zone the velocity profiles can be considered periodic of the same period of the channel. On the other hand, the effect of the pressure field is to allow the fluid flow, acting against friction forces due to the viscous behaviour of the mean. The dissipative work is negligible and its contribution is not included in the energy equation, yet a pressure drop is present along the streamwise direction. The pressure field can be split into two contributions:

1. a *linear decaying* term bound to the pumping energy;
2. a *periodic* term related to the detailed local motion.

The behaviour of the pressure along a channel of varying cross-sectional area is depicted in figure 3.3. The split of the pressure assumes the following expression:

$$p(x, y, z) = -\beta \cdot x + P(x, y, z) \tag{3.12}$$

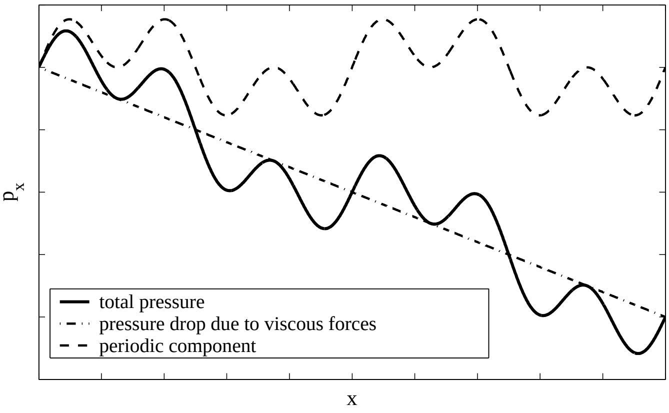


Figure 3.3 Typical behaviour of the mean pressure along the line of a channel with non constant section

Therefore the momentum equation in the x direction can be rewritten as:

$$\mathbf{u} \cdot \nabla u = \frac{1}{Re} \nabla^2 u - \frac{\partial P}{\partial x} + \beta \tag{3.13}$$

where β becomes a volume force term whose value, influencing the Reynolds number, will be adjusted in the solution procedure.

3.1.3 Boundary conditions

Fluid dynamic The condition of periodic velocity profiles can be expressed in the inlet and outlet boundaries as follows:

$$\mathbf{u}_{in} = \mathbf{u}_{out} \tag{3.14}$$

and, in general:

$$\mathbf{u}(x, y, z) = \mathbf{u}(x + L, y, z) \tag{3.15}$$

where L is the length of the repeating module. After the split of the total pressure, the newly introduced periodic pressure variable, P, can be handled as the velocity field:

$$P_{in} = P_{out} \tag{3.16}$$

and, in general:

$$P(x, y, z) = P(x + L, y, z) \tag{3.17}$$

Finally, *no-slip* conditions are imposed at the walls. The only difference between the two dimensional and three dimensional cases is the presence of two symmetry planes, at the edges of the extrusion, as will be shown later.

Temperature Shah and London [30] discuss fully developed flow in parallel plates for many wall-boundary condition cases. In this study a constant-temperature boundary condition is used, as representative of *e.g.* automotive radiators, at high liquid flow rates, with negligible wall thermal resistance. Note that the constant heat flux is a simpler boundary condition to deal with, see [31].

By *fully developed thermal field* it is meant that the Nusselt number is constant along the flow.

Barletta and Zanchini [32] define the conditions for the existence of a thermally developed region for laminar forced convection in circular ducts.

As in Patankar et al. [29], the concept of the thermally developed regime is generalized to the *periodic* thermally developed one. The condition for a constant cross-sectional channel writes as follows:

$$\frac{\partial \theta_p}{\partial x} = 0 \tag{3.18}$$

where θ_p is the local non-dimensional temperature

$$\theta_p = \frac{T(x,y) - T_w}{T_{b,x} - T_w} \tag{3.19}$$

In the case of periodic wavy channel, a weak condition, between two sections a period length apart, can be imposed:

$$\theta_p(x,y) = \theta_p(x+L,y) \tag{3.20}$$

The use of the periodic temperature field, defined in eq. (3.19), leads to a volume force term, in the energy equation, which depends on the x streamwise coordinate. So, another equation must be introduced [29], but this is cumbersome to implement on the triangular unstructured grids that COMSOL/FEMLAB uses. For this reason another strategy has been used to tackle the problem.

A fixed arbitrary reference value has been adopted to make the temperature non-dimensional. The value chosen, for convenience, is the bulk temperature at the inlet boundary.

So the periodicity condition, eq. (3.20), changes into:

$$\theta(x_{in},y) = \theta(x_{out},y) \cdot \sigma \tag{3.21}$$

where

$$\theta(x,y) = \frac{T(x,y) - T_w}{T_{b,in} - T_w} \tag{3.22}$$

and σ is the ratio between the inlet and outlet temperature differences:

$$\sigma = \frac{T_{b,in} - T_w}{T_{b,out} - T_w} \tag{3.23}$$

In place of the adjoint equation introduced by Patankar [29], an iterative procedure based on an energy balance has been introduced to reach fully developed conditions. It will be explained in the next section.

3.2 Numerical methods

The numerical solution of the problem was carried out by means of the FEMLAB software package. At present, the scripting language of FEMLAB (rel. 3.1i) is constituted by MATLAB ".m" files [33]. After the modellization of the problem, an iterative method to solve the fluid-dynamic and thermal fields with the imposed boundary conditions has been implemented in MATLAB scripting language.

3.2.1 Fluid dynamic iterative solution

As noted in eq. (3.13), a forcing term β appears, due to the pressure splitting introduced in eq. (3.12). Since the Reynolds number is a given constant of the problem, the non-dimensional mean velocity in the channel must be unitary. It is, therefore, necessary to find the correct value of β that ensures this condition. Other authors [34] have used proportional-integrative iterative controls to reach the correct value of the pressure gradient, starting from a trial value. At first a similar approach has been attempted, but it proved to be not very efficient in converging to the correct value of β, and this can be a limiting factor for CPU-intensive optimization studies.

By the definition of the friction factor f as the non-dimensional surface shear stress [30], it is easy to show, by applying the second Newton's law that:

$$f = \frac{\beta}{2} \tag{3.24}$$

Recall that, for internal flows in laminar regime, the friction factor is proportional to the inverse of the Reynolds number. From the definition of Re, it follows that:

$$\beta \propto U_{av} \tag{3.25}$$

This relation is not strictly valid in channels with varying cross section, but nevertheless it is expected a proportionality law such as:

$$\beta \propto U_{av}^{m} \tag{3.26}$$

to hold, with a value of the exponent m close to 1. In these conditions, evaluating the flow field with a test value for β, taking into account that the desired average velocity is unitary, and updating iteratively the pressure gradient as follows:

$$\beta_{n+1} = \frac{\beta_n}{U_{av,n}} \tag{3.27}$$

it is expected to reach the exact value of β in few steps. This has been verified during this study, where only 3 to 6 steps are required to reach a value of β which gives an error on Re below 0.1%.

3.2.2 Thermal field iterative solution

The thermal field is computed, after the velocity field has been obtained, with an iterative approach. This is based on the fact that the heat flux at the wall has to be balanced by the enthalpy difference between inlet and outlet. The task is to find a value of σ, eq. (3.23), that ensures this balance. From this balance, written here for clarity reasons in dimensional form, it follows:

$$\dot{m}c_p\left(T_{b,in} - T_{b,out}\right) = \int_w -k\frac{\partial T}{\partial n}\mathrm{d}s \tag{3.28}$$

assembling the dimensional terms and remembering eq. (3.23), one is left with

$$1 - \frac{1}{\sigma} = \frac{2}{Re\,Pr}\int_w -\frac{\partial \theta}{\partial n}\mathrm{d}s \tag{3.29}$$

At this point, there is a relation between σ and the heat flux at the wall, and in particular an incorrect value of σ leads to a generation term on the outlet boundary, which has no physical meaning. So, starting from a tentative value of σ and updating it iteratively by means of (3.29), it leads to the correct solution.

Once the correct thermal field has been calculated, the mean Nusselt number, Nu, is obtained as:

$$Nu = \frac{1}{2L}\int_w Nu_x\mathrm{d}s \tag{3.30}$$

$$Nu_x = \frac{h_x D_h}{k} \tag{3.31}$$

where Nu_x and h_x are, respectively, the local values of the Nusselt number and heat transfer coefficient. From an energy balance, again in dimensional form, at a generic section of the channel, one has:

$$\dot{m}c_p\mathrm{d}T_b = -h_x(T_b - T_w)\mathrm{d}s \tag{3.32}$$

Multiplying each side by $(\mu\,k\,D_h)$, expressing the mass flow in its components and integrating, one finally obtains

$$Nu = \ln\sigma\frac{D_h}{4L}Re\,Pr \tag{3.33}$$

3.2.3 Direct problem solution

FEMLAB is a general purpose, multiphysics, unstructured FE package [35]. The computational domain is first discretized by means of an unstructured grid, made of triangular elements. FEMLAB allows using different kind of element shape functions, i.e. Lagrange, Hermite, Argyris up to cubic form [36]. As a good compromise between accuracy and robustness the unequal-order interpolation scheme [35] has been used: second order for the velocity components and the temperature and first order for the pressure. The nonlinear solver uses a damped iterative Newton method, while the linearized problem is solved in a coupled way, by means of the UMFPACK routine included in the FEMLAB package.

Grid independence

A grid-independence study on 2D geometries was performed, for different geometrical configurations of the channel (designs), in order to ensure the accuracy of the results presented. In figure 3.4 the test for the depicted geometry is shown. As a compromise between accuracy and computing costs, the latter particularly high for multi-objective optimization tasks, mesh resolutions of the order of $(5 - 7) \times 10^3$ elements have been used throughout this study.

In the 3D extension of the study, due to computational cost and memory usage a grid independent mesh has not been achieved. This is due to the lack in the version used of COMSOL/FEMLAB of solving procedures dedicated to fluid dynamic computations, which are highly computational demanding. The number of elements used in 3D calculation is about 10^4.

3.3 Periodic module construction

In this study the shape of a convective channel is represented either by linear-piecewise profiles, or by NURBS, and a comparison between these geometries is performed. The former type of wall shape is used in several industrial applications. However, the second type allows a better fluid-dynamic behavior, i.e. a decrease in wall shear stress, for the same heat transfer rate, and in addition gives more freedom in the selection of possible shapes. This aspect, as it will be seen later, is of particular importance in shape optimization, since it has been noted that one of the most important features in this analysis is the choice of the design variables to be used, and how the shape is parametrized in terms of this design variables [37]. Selecting too many variables will complicate the optimization problem, with consequent increase of the computational time required, but, vice-versa, choosing too few variables may result in a limited range of shape alternatives being obtained. It is therefore a fundamental requirement that a wide range of shapes, defined by a relatively small number of parameters, are within the reach to the method of optimization used. It must be said that various tools for geometrical parametrization of smooth curves exist and they could be interchanged with each other. One can mention *Spline*, *Bézier* and obviously *NURBS* forms [27].

The FEMLAB release used, 3.1, works only with curve in Beźier form. The NURBS entity has been constructed by means of in-house coded functions. In few words a number of Bézier patches equal to the number of intervals defined by knots

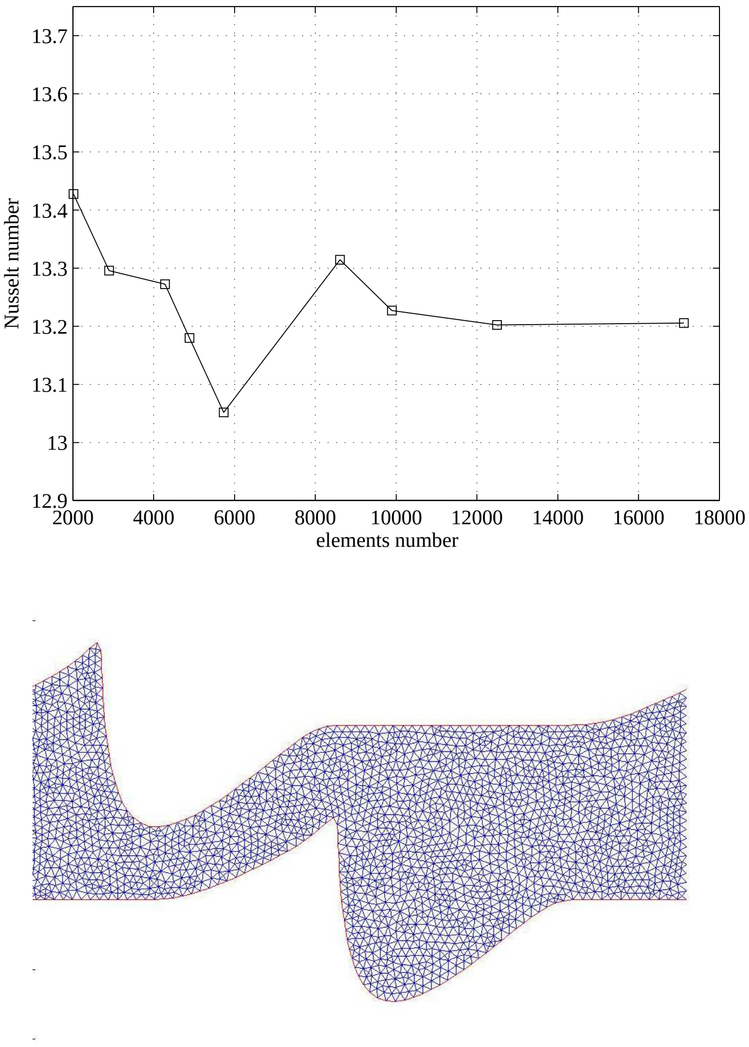


Figure 3.4 Grid independence test example

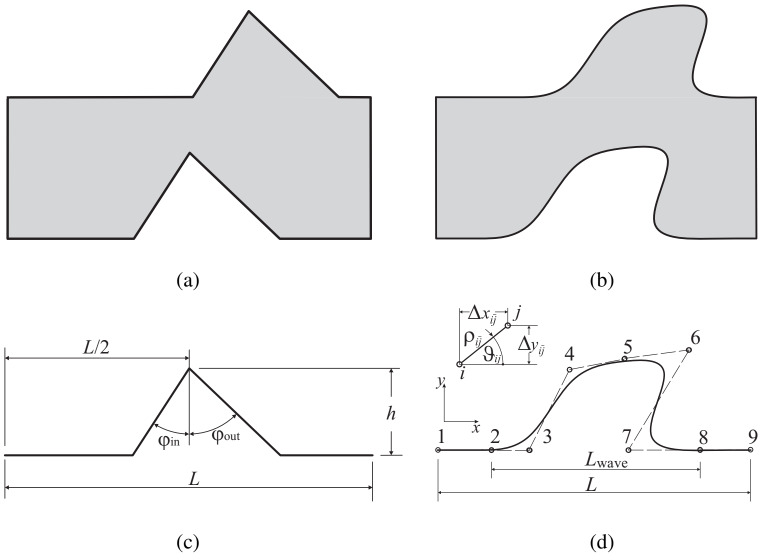


Figure 3.5 Periodic channels and geometrical parametrization. (a) Linear-piecewise channel; (b) NURBS channel; (c) Linear-piecewise parametrization; (d) NURBS parametrization and control points.

span is needed. This, however, has the drawback that it increases the number of edges in the computational domain, with possible difficulties in the mesh generation process.

In the following it will be described how the two different geometrical profile, the linear-piecewise and the NURBS, depicted in figures 3.5(a) and 3.5(b), respectively, have been parametrized. Recall that parametrization means describing a system using a discrete set of variables, and that degrees of freedom of the system (DOFs) and design variables are equivalent concepts. Subsequently the channel construction, similar in both cases, will be treated.

3.3.1 2D Linear piece-wise parametrization

The linear-piecewise wall profile of the channel is characterized by a small number of DOFs. The variables introduced in this wall parametrization are presented in figure 3.5(c) and summarized in table 3.2. There are four DOFs requested for describing the wall profile. The profile, realized defining a set of points and connecting them with straight lines, is constructed, for convenience, in order to have the edge of the corrugation centered on the wall.

Table 3.2 Variables defining the linear-piecewise channel.

Variable	Symbols	Range	Basis
module length	L	$[0.8; 2.0]$	501
corrugation height	h	$[0.0; 0.5]$	501
forward edge angle	φ_{in}	$[10°; 60°]$	501
backward edge angle	φ_{out}	$[10°; 60°]$	501
translation of upper wall	$transl$	$[-0.5L; 0.5L]$	501

3.3.2 2D NURBS parametrization

In order to generate the NURBS channel during the optimization process, let's start by defining first the wall profile. As a good compromise between the number of DOFs and the geometrical complexity, a 9 control-points periodic cubic NURBS has been chosen, to ensure the periodicity of the channel itself. As already described, a large number of variables allows to describe minutely the profile, but an excessive number of DOFs would make the optimization process quite difficult and expensive. For this reason it has been also decided to fix to a unitary value the curve's weights. This, together with the uniform knots distribution we adopted, makes our NURBS curve practically equivalent to a B-Spline curve.

As depicted in figure 3.5(d) both the first and the last three aligned control points are needed to maintain the entrance and the exit of the profile parallel to the x direction. The remaining ones give freedom to the wavy section. The parameters required for the definition of the lower profile are explained in table 3.3 and numbered as in figure 3.5(d). The symbol Δ means the difference between two points' coordinates, while $\rho_{\overline{ij}}$ is the module and ϑ_{ij} the phase of a polar-coordinates system centered on the point i. The phase is positive counterclockwise with respect to the positive direc-

Table 3.3 Control points for the NURBS-profile and parameters required.

Point	x	y	DOFs
1	0	0	0
2	$(L - L_{wave})/2$	0	0
3	$x_2 + \Delta x_{\overline{23}}$	0	1
4	$x_5 + \rho_{\overline{54}} \cos \vartheta_{54}$	$y_5 + \rho_{\overline{54}} \sin \vartheta_{54}$	2
5	$x_1 + \Delta x_5$	$y_1 + \Delta y_5$	2
6	$x_5 + \rho_{\overline{56}} \cos \vartheta_{56}$	$y_5 + \rho_{\overline{56}} \sin \vartheta_{56}$	2
7	$x_8 - \Delta x_{\overline{87}}$	0	1
8	$x_2 + L_{wave}$	0	1
9	L	0	1

tion of the x axis. The profile is again constructed, for convenience, in order to have the wavy part centered on the wall. Therefore the value of parameter x_2 is equal to $(L - L_{wave})/2$. The number of DOFs is now 10.

3.3.3 Channel construction

In figures 3.5(a) and 3.5(b) both types of channel are depicted, i.e. linear-piecewise channel and NURBS channel, and they are constructed in the same way. Once the lower wall profile has been obtained, the upper one, as illustrated in figure 3.1, is made by a simple translation in y-direction of the former, in order to obtain the height of the channel, followed by a translation in x-direction. This guarantees the realistic desire to construct the channels, of e.g. a finned heat exchanger, by simple juxtaposition of identical wavy plates. This action introduces in both cases another DOF, which defines the x translation of the upper profile, variable *transl* in tables 3.2 and 3.5. In order to avoid linear-dependent channels, the translation range is bounded to one half-period, both in positive and negative x direction. In order to guarantee smoothness of the channel walls (see Appendix), the lower wall is made up of one curve, repeated periodically three times, and an identical pattern is applied to the upper one. The y translation is instead fixed: in this way the average height of the channel is set to 0.5, that is half of the non-dimensional hydraulic diameter (D_h). Finally the periodic duct module is cut by two straight lines, representing the inlet and the outlet section, as sketched in figure 3.1. Overall, we are left with 5 DOFs for the linear-piecewise model, and 11 DOFs for the NURBS channel.

3.3.4 3D extension

Once the two-dimensional results have been obtained, it has been tried to encourage further mixing in the flow, and therefore higher heat transfer rate, by forcing a non-zero value of the z-component of the velocity vector [38]. Three dimensional analysis has been performed by simple extrusion from two dimensional modules. For this purpose, the best two-dimensional optimized channels, in term of low friction factor and high Nusselt number, have been selected first, and one new additional variable was introduced, that is the extrusion angle. The constructing procedure is described in figure 3.2. Due to the exploratory nature of this study, the extrusion length is fixed to the same value of channel height.

3.4 Optimization process

The parameters that influence the performances of an heat exchanger, and in particular of the single repeating module under study, are its overall heat transfer coefficient and the pumping power required to supply a desired mass flow. In a dimensionless perspective, the role of these two parameters is played by the dimensionless friction factor and the Nusselt number, introduced earlier in this chapter. both quantities are function of the geometric variables:

$$[f, Nu] = g(\mathbf{X}) \tag{3.34}$$

where vector **X** is constituted by the degrees of freedom of the chosen parametrization, either linear piecewise or NURBS. An analytical function g is unknown, but in a limited number of geometrical configuration As stated in chapter 1, there is an ample variety of numerical techniques to perfor optimization tasks. Evolutionary algorithm are within the most robust ones, and have been used in this optimization process. In particular, MOGA-II algorithm, whose features have been explained in chapter 1, has been employed.

To perform the optimization task, the software modeFRONTIER [18] commercialized since 1999 by EST.TEC.O. s.r.l. Company, has been chosen.

The optimization process follows these tasks, as sketched in figure 3.6:

- Automatically the optimization software generates a set of numbers, i.e. the geometrical design variables, representing the shape of a channel (called *individual*).
- These variables are written in an input file, which is sent to the CFD solver. This, in turn, computes the flow and thermal fields, and from these, it evaluates the friction factor and the Nusselt number, which represent the objective functions.
- The numerical values of the objective functions are sent back to the optimizer, which generates another set of geometrical parameters.

The results are expressed in terms of the ratios Nu/Nu_0 f/f_0, where Nu_0 and f_0 are the Nusselt number and the friction factor for a parallel plate channel, taken as reference geometry.

3.5 Results

3.5.1 Linear-piecewise optimization

The first analysis has been conducted on the linear-piecewise geometry type. The variables are small in number and clearly related to the shape of the channel: the depth of the asperity, the forward side angle, the backward side angle, the length of the channel and the translation of the upper wall.

The first step in an optimization process is the definition of a convenient starting point. In an evolutionary optimization process this means choosing a well-distributed set of initial individuals (DOE). The Full Factorial algorithm [18] with three levels has been used to achieve the task. This method gives the best homogeneous distribution of the samples. The number of Full Factorial samples is 243 and the parameters

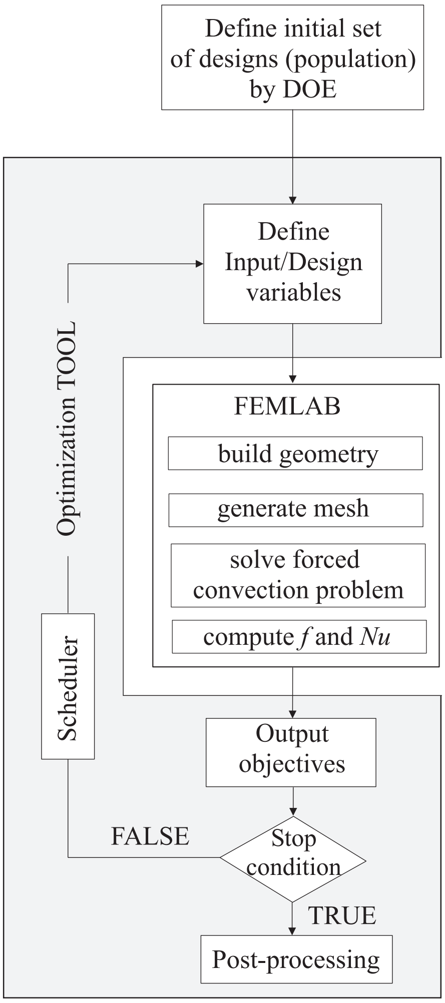


Figure 3.6 Optimization work-flow.

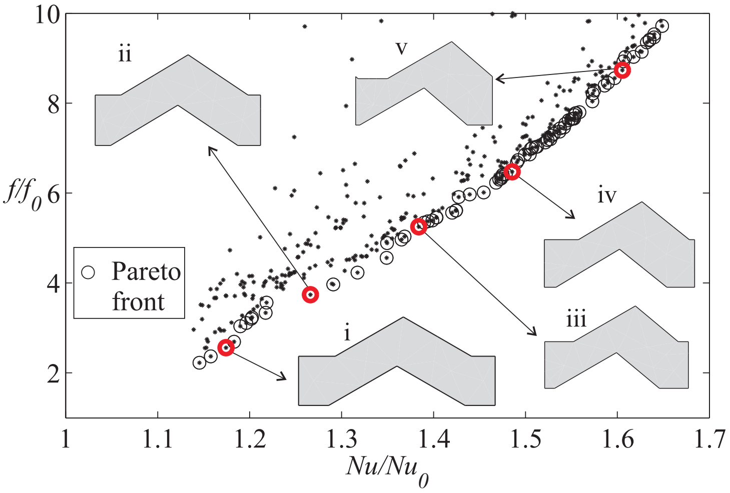


Figure 3.7 Pareto front along the linear-piecewise optimization process.

ranges are given in table 3.2. In a GA optimization the size of the population within the design space affects the convergence ratio. After having performed the numerical simulation on this set, its Pareto front has been chosen as the initial population of multi-objective optimization with GA. This preliminary Pareto front is a good selection in order to limit the number of starting channels. The optimization has been realized with MOGA-II along 30 generation of 20 individuals each: so the total number of designs evaluated is 600. The results are sketched in figure 3.7, where the Pareto front is highlighted. Due to the simplicity of the parametrization, the dominant set can probably be taken as the limit performance of this kind of geometry. In fact, further optimization process has not given appreciable improvements on design objectives.

Analyzing the shape of the channels along the Pareto front, sketched for convenience in the same figure, it results that there is no sensible fluctuation for variables

Table 3.4 Values of the design variables for the selected linear-piecewise channels.

	ID i	ID ii	ID iii	ID iv	ID v
L	2.00	1.64	1.54	1.58	1.47
h	0.400	0.400	0.383	0.400	0.400
φ_{in}	60.00°	59.04°	60°	59.04°	60.00°
φ_{out}	60.00°	56.04°	50.16°	51.24°	49.32°
$transl$	0.066	0.102	0.202	0.236	0.298

Table 3.5 Variables' ranges for the first NURBS optimization

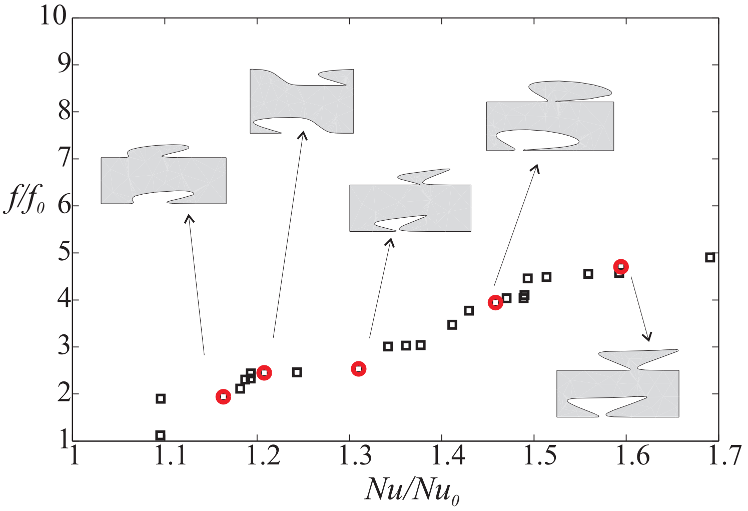
Parameter	Range	Basis
L	[1.00; 2.50]	1001
L_{wave}	$[0.15\,L; 0.85\,L]$	1001
Δx_{23}	$[0.01\,L; 0.20\,L]$	1001
ρ_{54}	[0.05; 1.20]	1001
ϑ_{54}	[70°; 300°]	501
Δx_5	$[0.30\,L; 0.75\,L]$	1001
Δy_5	[0.00; 0.50]	1001
ρ_{56}	[0.05; 1.10]	1001
ϑ_{56}	[−120°; 150°]	501
Δ_{87}	$[0.01\,L; 0.80\,L]$	1001
$transl$	$[-0.500\,L; 0.500\,L]$	1001

like φ_{in} and h, which remain close to 60° and 0.4 respectively. What makes the difference is the translation of the upper profile, responsible of the development of the separation bubble induced by the corrugation. In table 3.4 the values of the design variables required for the definition of the channels, marked in figure 3.7 for illustrative purpose, are presented.

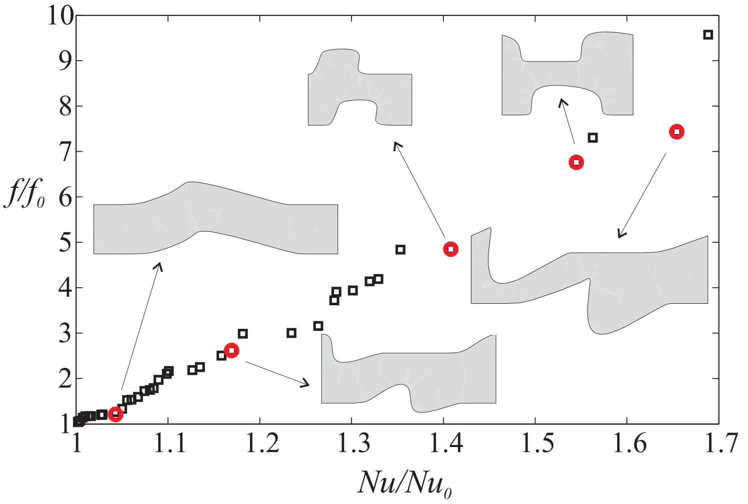
3.5.2 NURBS optimization

Having performed an optimization process on linear-piecewise model, The attention has been focused on on the more complex NURBS based channels. As already stated, the increased number of degrees of freedom causes a more expensive optimization task.

In contrast with linear-piecewise optimization, the Full Factorial algorithm has not been used as first examination of the design space. There being 11 degrees of freedom, the number of individuals to be computed would have rise to 3^{11}, that means 177147, with a three levels Full Factorial. The Sobol algorithm [18] has been chosen to define an initial population of 50 individuals. The Sobol algorithm uses a quasi-random strategy which uniformly distributes such number of parameters in the best possible way. The optimization algorithm chosen was again MOGA-II. The optimization has started allowing great freedom to the design variables, and no constraint has been imposed. In table 3.5 the summary of design variables ranges and the number of steps (basis), which notches them to discrete form, are presented. The choice of the variables and their range might dramatically affect the convergence rate to a good solution. In our case the generation of input strings leading to incoherent geometries has to be avoided as much as possible. One strategy is to scale the x-direction Cartesian variables to the length of the channel L. Therefore all parameters are proportional to each other.

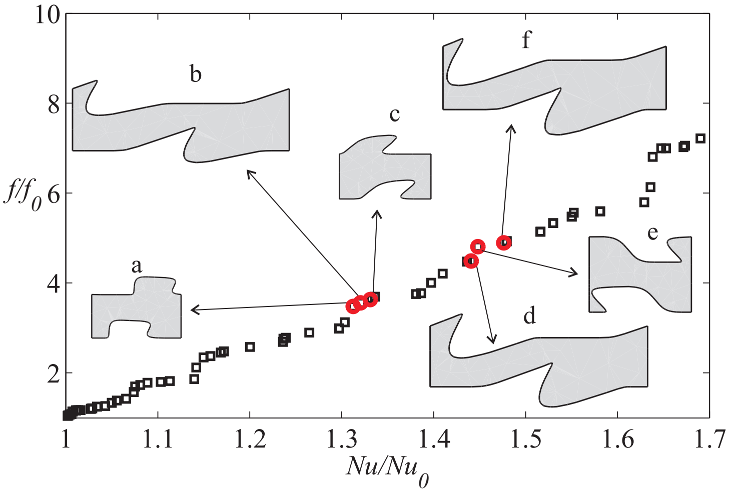


(a)



(b)

Figure 3.8 Pareto fronts comparison between different optimization process: (a) NURBS Pareto front for the first MOGA; (b) first MOGA with constraint; (c) last Pareto front.



(c)

Figure 3.8 . . . Continued.

Figure 3.8(a) shows the Pareto front after this first optimization stage. Five channels, representative of different combination of the two objectives, are highlighted. From figure 3.8(a) it is clear that all the individuals selected have in common the presence of closing bends in wall profile. The same happens for almost all the other channels. Taking into account the industrial feasibility, though only from a methodological point of view, these channels are very far from being realized, for example, in a pressing process. Therefore a sort of *fabricability check* has been implemented, and its aim was to discard those geometries whose wall profile could not be obtained by moulding. This constraint can be summarized as follows. During the construction procedure, after the lower wall profile has been drawn, it is *brushed* along x direction by straight lines which progressively change their slope ([−45°; +45°] respect to y axis), figure 3.9. The channel is declared feasible if and only if, there exists at least one direction at which all the lines encounter the wall profile no more than once. The simulation has been restarted with the same parameters, but introducing the fabricability check just described. The results are depicted in figure 3.8(b). The presence of such a constraint makes the optimization process closer to the channel shapes more common in practice. Its observation reveals the decreasing performances of the second set, as it comes out from a comparison between figures 3.8(a) and 3.8(b). As already anticipated the MOGA algorithm is well suited for truly multi-objective problems. It is robust, albeit somehow slow for increasing number of objective functions or design variables.

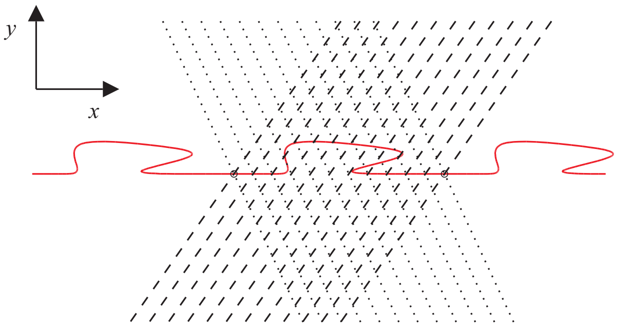


Figure 3.9 Fabricability check.

Once an high quality Pareto front has been obtained, one can choose slightly different strategies in order to improve the fitness of channels. One way is to transform the multi-objective problem into a single-objective one by means of a weighted function, involving objectives. The modeFRONTIER software package combines several different objectives into a unique monotone function using preference relations between designs, which can be set by the user. The mono-objective optimization, performed using MOGAII and SIMPLEX [1] algorithms, makes the process faster.

Having two starting objectives, imposing different relations between designs, a different weight distribution on f and Nu could be given and a different ranking to each alternative would be assigned. Using MCDM, two kind of utility functions were created, the first privileges the increase of Nusselt number, whereas the other is more focused toward the reduction of the friction factor. In this way, after about 2500 designs evaluated, the results summarized in figure 3.8(c) have been obtained.

In the same figure two sequences of three channels each one, having almost the same performance metrics, are marked. They represent different arrangements of geometries in the Pareto front. After the NURBS optimization process it has been recognized that two different families of channels shapes belong to the part of the Pareto front characterized by high values of f and Nu, and they are shuffled. Although various kind of corrugations are present, the main difference between the two type, the one called S for short (ID a, ID c, ID e), the other L for long (ID b, ID d, ID f), is the length of the module. The average length S channels is 1, while the one of L is 2, so the ratio between length of channels S and channels L is 0.5. This is an important features of the optimization, because it shows the non-univocity of the solution, i.e. similar performances can be reached by different geometries [39].

In figure 3.10 the design database has been filtered and divided into two categories. The blue ones are the shorter channels, while the red ones are longer. Pareto domi-

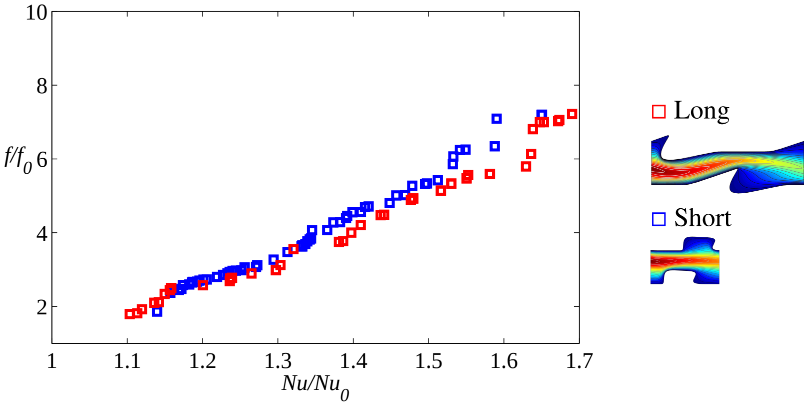


Figure 3.10 Two distinct Pareto front for the different type of geometries.

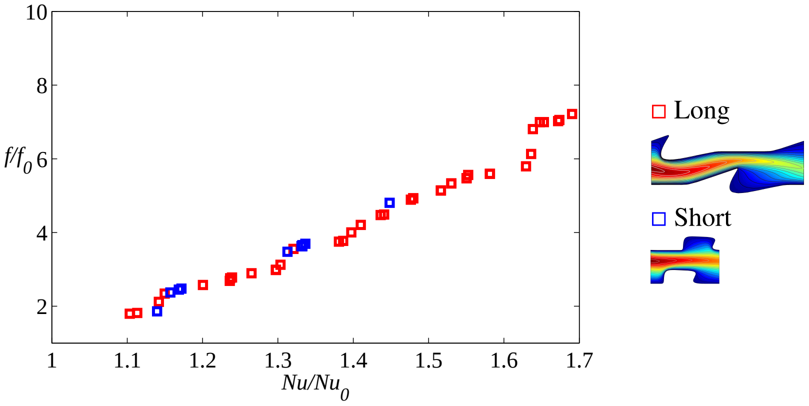


Figure 3.11 Overlapping of the solution.

nance applied to both the subset of design shows there are clearly two fronts made of completely different geometries that overlap.

From figure 3.11 it is clear that the unfiltered front is mainly made by long geometries, with superpositions of short channels. Therefore figure 3.8(c) shows two examples of this phenomena. It should be noted that at low values of f and Nu, the shape of designs is close to parallel plate channel, so the different length does not affect the form of the wave.

The six channels marked in figure 3.8(c) are shown in figure 3.12, together with the streamlines and non-dimensional temperature field. They are scaled with the same average height in order to be directly compared. In table 3.6 the values of control points and of upper wall translation for the six channels are listed.

3.5.3 Linear-piecewise versus NURBS

In figure 3.13 results obtained by the linear-piecewise and NURBS optimization are compared, The one *St*, marked by stars, represents channels with linear-piecewise wall profile; the other *Sq*, marked by squares, represents channels with smooth NURBS wall profile. The higher geometrical complexity and computational costs of NURBS channels is well counterbalanced by the significant performance improvement over the simpler channels with linear-piecewise walls. The greater number of variables makes the convergence slower and an asymptotic limit of the front was not reached during the optimization of NURBS profile channel.

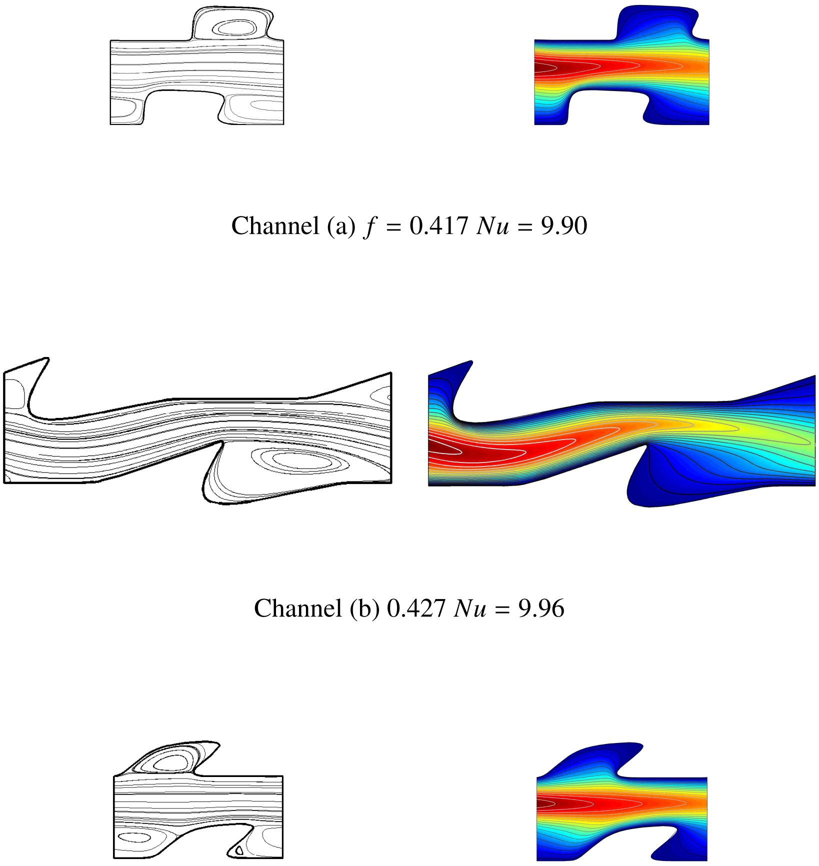
3.5.4 3D analysis

This part of the work is only a proof-of-concept in 3D applications. a true optimization has not been performed on NURBS channels but, due to time limitation and computing resources available a parametric analysis has been done in order to verify the applicability of the method to 3D problems. For the same reasons the Reynolds number has been reduced from 200 to 100, in order to guarantee an adequate level of numerical accuracy. Therefore the 2D Pareto fronts have been re-computed at this reduced value of the Reynolds number, in order to compare the performances of the 2D channels with the 3D ones. The 3D ones have been obtained, as indicated in figure 3.2, by extrusion, at different angles, of selected 2D channels. In figure 3.5.4 and 3.5.4 the results obtained are presented. In particular, the results for the simpler channels with linear-piecewise walls are shown in figure 3.5.4 , while the objective functions for the NURBS-based channels are shown in figure 3.5.4. From the figures it is evident that, for both type of channels, i.e. linear-piecewise and smooth, the Nusselt number in general tends to increase, without a sensible increment in the friction factor, for the 3D channels when the extrusion give rise to 3D flow pattern.

This trend is due to the presence of secondary motions, that enhance the heat transfer at the walls. In table 3.7 the results for one of the NURBS profiled extrusions are reported, and it is again clear the positive effect of secondary motions and longitudinal

Table 3.6 Control points for the selected NURBS-based channels.

	ID a	ID b	ID c	ID d	ID e	ID f
(x_1, y_1)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
(x_2, y_2)	(0.160, 0)	(0.354, 0)	(0.179, 0)	(0.341, 0)	(0.113, 0)	(0.347, 0)
(x_3, y_3)	(0.207, 0)	(0.457, 0)	(0.252, 0)	(0.487, 0)	(0.171, 0)	(0.480, 0)
(x_4, y_4)	(0.190, 0.220)	(1.183, 0.251)	(0.409, 0.161)	(1.149, 0.248)	(−0.105, 0.249)	(1.068, 0.209)
(x_5, y_5)	(0.410, 0.199)	(0.196, 0.267)	(0.612, 0.207)	(1.152, 0.268)	(0.208, 0.279)	(1.111, 0.262)
(x_6, y_6)	(1.348, 0.198)	(0.893, −0.234	(0.965, 0.224)	(0.827, −0.362)	(0.551, −0436)	(0.696, −0.440)
(x_7, y_7)	(0.768, 0)	(1.830, 0)	(0.580, 0)	(1.853, 0)	(0.685, 0)	(1.890, 0)
(x_8, y_8)	(0.554, 0)	(1.950, 0)	(0.843, 0)	(1.880, 0)	(0.909, 0)	(1.916, 0)
(x_9, y_9)	(0.870, 0)	(2.304, 0)	(1.023, 0)	(2.221, 0)	(1.023, 0)	(2.263, 0)
transl	0.290	−1.039	−0.204	−0.920	−0.268	−0.994

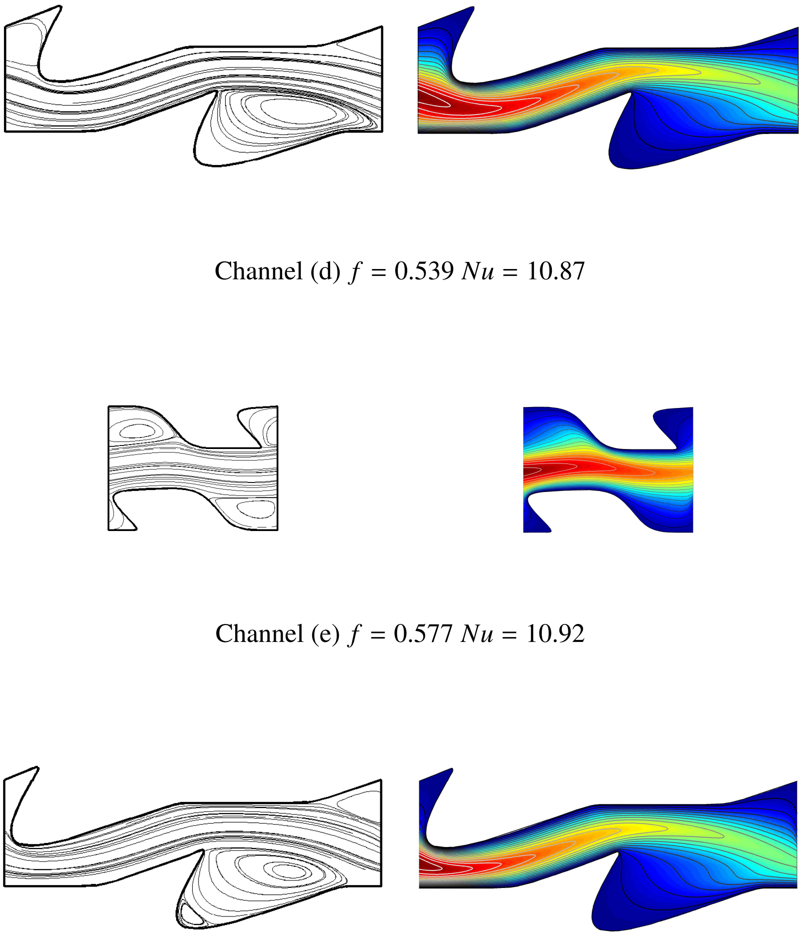


Channel (a) $f = 0.417$ $Nu = 9.90$

Channel (b) 0.427 $Nu = 9.96$

Channel (c) $f = 0.436$ $Nu = 10.04$

Figure 3.12 Selected NURBS-based channels: fluid-dynamic (left) and thermal (right) fields. These six channels correspond to designs marked in figure 3.8(c).



Channel (d) $f = 0.539$ $Nu = 10.87$

Channel (e) $f = 0.577$ $Nu = 10.92$

Channel (f) $f = 0.587$ $Nu = 11.13$

Figure 3.12 . . . Continued.

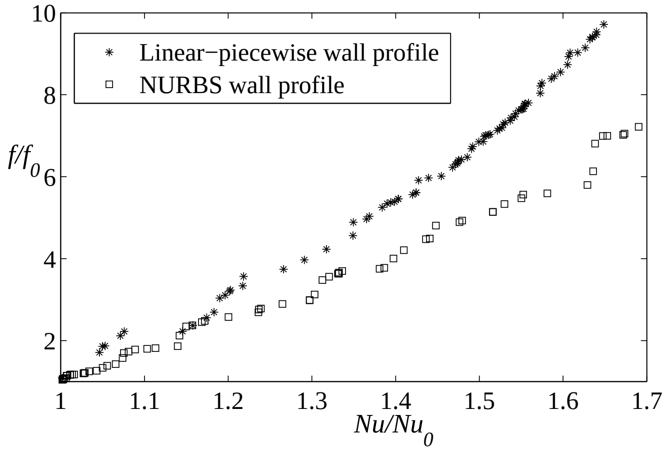
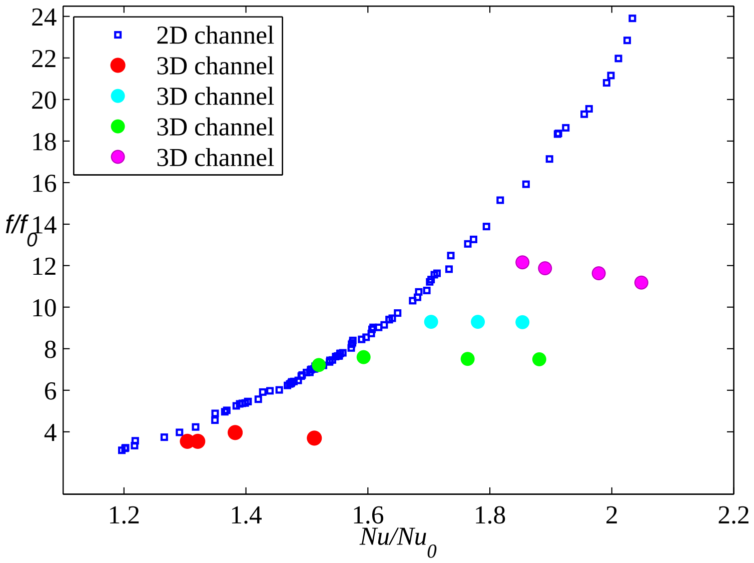


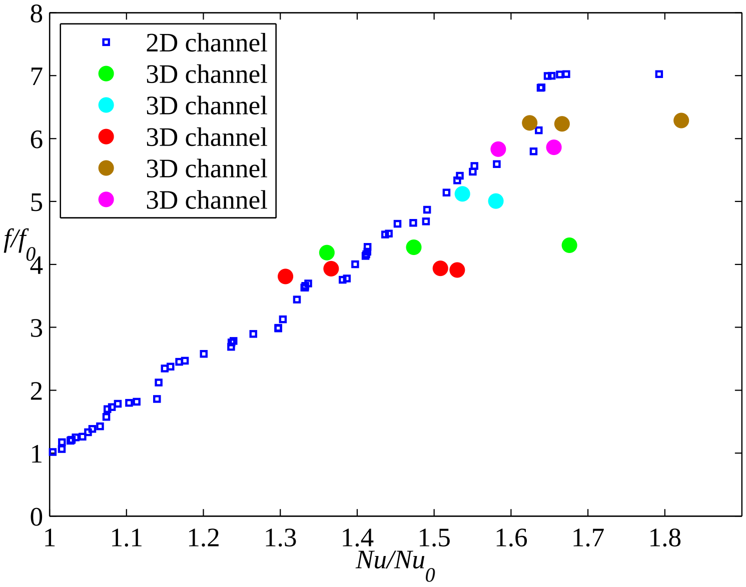
Figure 3.13 Pareto fronts comparison between linear-piecewise and NURBS-based channels.

Table 3.7 Variation of friction factor and nusselt number versus extrusion angle for one design

$\mathbf{f/f_0}$	$\mathbf{Nu/Nu_0}$	**Extrusion angle**
3.807	1.307	0°
3.932	1.366	20°
3.937	1.508	30°
3.909	1.530	40°



(a)



(b)

Figure 3.14 Pareto fronts comparison: a) linear-piecewise profile; b) NURBS profile

steady vortexes on the heat transfer rate. For illustrative purposes the secondary flow pattern, for two channels obtained with a 20 and 40 degrees extrusion respectively, are depicted in figure 3.5.4.

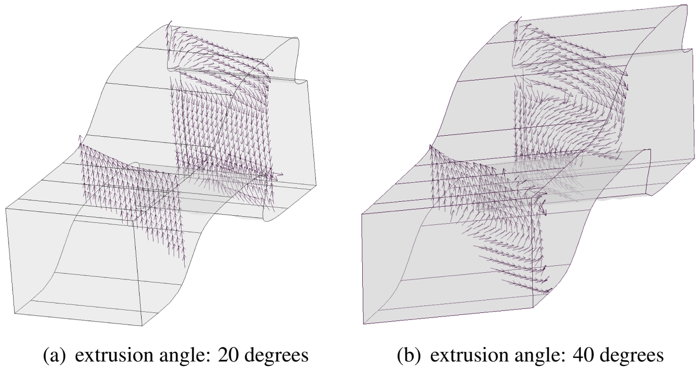

(a) extrusion angle: 20 degrees (b) extrusion angle: 40 degrees

Figure 3.15 3D Secondary vortexes

Finally a MOGA optimization has been led on extruded linear-piecewise channels for a short number of generations. The Pareto front of this optimization is compared in figure 3.16 with the 2D linear-piecewise and 2D NURBS fronts. Though the 3D linear-piecewise front is rather sparse because of a small number of individual processed, the heat transfer augmentation due to secondary motions is clearly visible

3.6 Comments

In this chapter the multi-objective shape optimization of periodic wavy channels, which represent a fundamental building block of a large variety of heat exchangers and other heat transfer devices has been performed. The geometrical model used to describe the channels makes use of NURBS, a powerfull tool to draw free-shaped curves and surfaces by approximation of set of points with the minimum amount of information, as described in chapter2. The numerical model of the channel has been solved by means of COMSOL/FEMLAB, a general purpose FE solver, coupled with modeFRONTIER, which is an optimization software.

To carry out the optimization it has been made use to evolutionary techniques, that have the capability to solve truly multi-objective problems. They are very robust and can be applied to almost any kind of problems because they do not require any constraint on the nature of the problem they deal with.

The use of smooth corrugations obtained with NURBS representation has been compared in a 2D configuration with optimized solutions of linear piecewise corruga-

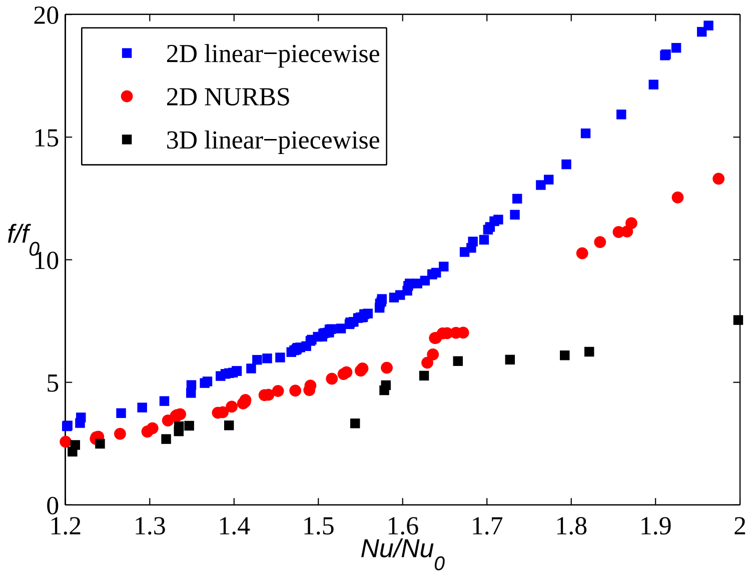


Figure 3.16 Pareto fronts: 2D-3D linear-piecewise and 2D smooth NURBS profile

tions, which are more commonly found in industrial applications. Smooth geometry have shown much more performing than linear piecewise, thus justifying the bigger effort in the optimization process, due to a higher number of degrees of freedom.

The use of multiobjective evolutionary techniques, and in particular the use of MOGA-II algorithm, has led to an unexpected conclusion that there is a completely separate type of geometries achieving the same performances. Two families of channels, the first with a module length close to the channel height, and the second with a module length of about the double of the channel height. This non univocity in the solution space has been already noted by other authors [39], and is a peculiarity that can rarely be achieved with classical optimization processes. The robustness of genetic algorithms, which mimic the evolution of living organisms in nature, evolving an initial population towards the best possible fitness, is capable of preserving multiple good solutions.

The 2D results have been extended, though in an exploratory fashion, to the 3D case, showing that the presence of secondary motions highly enhances heat transfer, without affecting the pressure losses.

Chapter Four

Inverse heat transfer problems

A heat transfer problem is fully defined by:

- A set of governing partial differential equations;
- Physical properties of the matetials;
- A complete set of initial and boundary conditions (BCs);
- A known geometrical domain and distribution of heat sources.

If any of this information is unavailable or incorrect, the porblem is *ill-posed*, and it cannpt be solved in a direct way, thus the epithet of *Inverse Heat Transfer Problems* (IHTP), where some assumptions have to be made to get the desired information.

There are situations in which, for technological reasons, it is not possible to use sensors to measure temperatures or fluxes on certain boundares, such as those of combustion chambers, over small electronic chips or in the coolant flow pasages of a turbine blade [40]. In this case one is force to find solutions for ill-posed problems where geometry and heat sources are known, but boundary condition are unavailable on part of the boundary. To get a solution, overspecified BCs have to be imposed on the known part of the boundary.

The same applies in cases where heat source distribution is partialy or totally unknown, as in chemical reactions in combustion processes. To get information on the sources distribution, both temperature and flux have to be imposed on the boundary, or at least a part of it.

A third class of inverse problems arises when one wishes the transfer phenomenon behave in a predefined way. Is such cases boundary conditions are usually overspecified and solutions can be obtained only if the geometrical domain can be appropriately modified. This kind of IHTP is a subset of *shape optimization* problems, which are problems where the goal is to find a shape (in two or three dimensions) which is optimal in a certain sense, while satisfying certain requirements.

Shape optimization is an infinite-dimensional optimization problem in the sense that the input variables of such problems are continuous entities (curves or surfaces) that cannot be determined by a finite number of degrees of freedom. Therefore the issue of a well conditioned geometrical model is of ultimate importance.

Common approach when solving inverse problems is to define the geometry and the boundary conditions, solve the governing equations for the direct problem, evaluate the results and make some changes in a cut-and-try way. Problems of this type are called analysis problems. But this methodology brings to week improvements with a large number of decision variables. Moreover, while searching more than one objective optimization, a cut-and-try method becomes impracticable.

The most used approach in heat transfer optimization problems is the sensitivity analysis. A two dimensional shape optimization for the Joule heating of solid bodies is described by Meric [41], who used the adjoint variable method and the material derivative technique, with a finite element (FE) discretization of the non-linear primary problem and the linear adjoint problem. Cheng and Wu [42], and later Lan et al. [43], considered the direct design of shape for two-dimensional conductive problems. A boundary fitted method was used to discretize the problem, and a direct sensitivity analysis to minimize the objective function. This methodology, based on the conjugate gradient method, was more recently extended to the conjugate (conduction + convection) heat transfer problem [44].

In this work, focused upon inverse (sometimes called direct) design of shape, we show that the combination of a general unstructured, adaptive FE solver, like FEMLAB [36], and a modern, multiobjective optimization system, like modeFRONTIER [18], constitutes a powerful and flexible tool for the inverse design of shape in heat transfer. The study follows from a preliminary work, which was limited, for the inverse design of shape, to two-dimensional conduction problems [42].

4.1 2D Conductive problem

In this case it has been attempted to reproduce the two-dimensional direct design of shape considered in [42, 44] and also in [45], where a gradient based method has been applied to solve the IHTP on a geometry model based on linearly interpolated points.

The numerical solution of conductive heat transfer is less CPU time consuming than conjugate heat transfer. Thus this problem is solved in order to test a series of geometrical parametrization to be applied to the second case.

4.1.1 Problem statement

A thin substrate of length L and with specified temperature $T_w(x)$ is embedded in an omogeneous and isotropic body of conductivity k_s, as sketched in figure 4.1. Heat flows from the substrate to the outer surface, and then into a fluid ambient with known temperature T_a. The heat transfer coefficient h is prescribed, therefore the problem reduces to a conductive transfer evaluation.

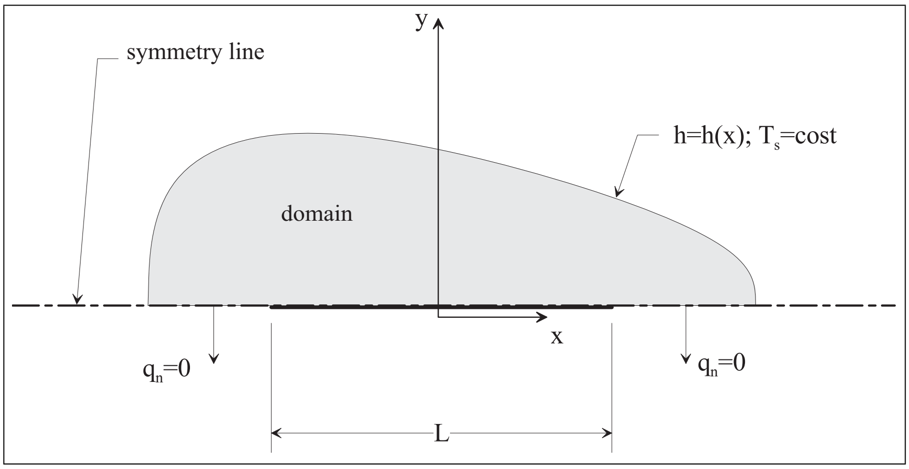


Figure 4.1 2D conductive problem: general scheme

The energy conservation equation can be expressed for an isotropic conductive medium with constant thermophysical properties as:

$$\rho_s C_s \frac{\partial T}{\partial t} = k_s \nabla^2 T + q^* \tag{4.1}$$

dove ρ_s is the density of the solid body, C_s is the specific heat, T the temperature, and q^* an internal heat source. Equation 4.1 can be made dimensionless introducing some reference quantities, in accordance with Buckingham theorem:

- L as length reference, that is a proper geometrical characteristic;
- $\Delta T = T_a - T_L$ as temperature reference, where T_a is the ambient temperature and T_L is a known temperature for the solid medium, that yelds to the dimensionless expression:

$$\theta = \frac{T - T_\infty}{T_L - T_\infty} \tag{4.2}$$

- $Bi = hL/k_s$, the Biot number that makes thermal heat transfer conditions dimensionless at the boundaries.

The problem is considered symmetric respect to the x direction, thus half of the geometry is actually modeled. Transient phenomena are not taken into acconunt and no heat sources are present apart from the thin substrate. In such conditions, the dimensionless energy conservation equation becomes:

$$\nabla^2 \theta = \frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} = 0 \tag{4.3}$$

Boundary conditions are imposed as follows, where **n** is the outward normal versor:

$$\theta = \theta(X) \tag{4.4}$$

along the heated substrate;

$$\mathbf{n} \cdot \nabla\theta = 0 \tag{4.5}$$

on the symmetry line;

$$-\mathbf{n} \cdot \nabla\theta = Bi(x)\,\theta \tag{4.6}$$

on the external surface.

the boudary condition of imposed temperature at the thin substrate has the following linear distribution:

$$\theta = 1.25 - 0.5 \times x; \tag{4.7}$$

where x ranges in the interval $[-0.5, 0.5]$ The extreme values for the temperature are 1 and 1.5.

The Biot number has been given both a constant unitary value and subsequently a quadratic distribution to better reflect the actual transfer behaviour when convective conditions are imposed:

$$Bi(x) = 3 - \sqrt{2x + 4} \tag{4.8}$$

Two different surface temperature have been searched:

- Case **A**: $\theta_s = 0.6$
- Case **B**: $\theta_s = 0.9$

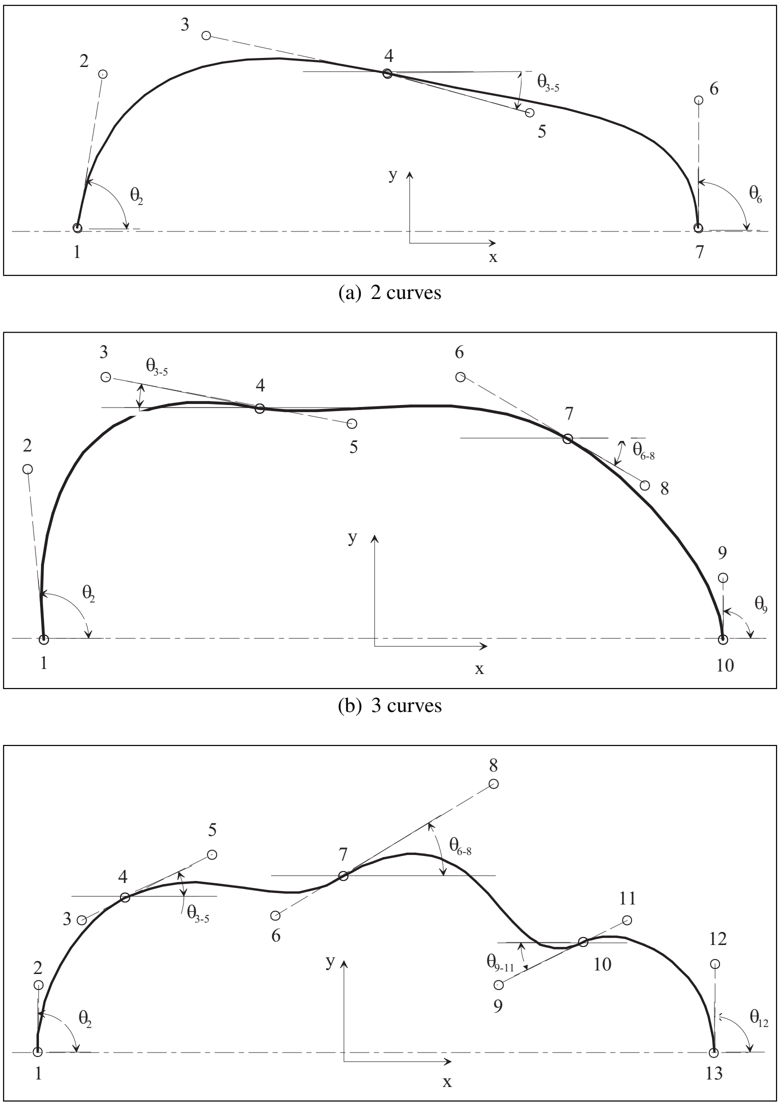
4.1.2 Geometry Modelling

As underlined at the beginning of this chapter, shape optimization problems are infinite-dimensional, thus the choice of a good parametrization is not a trivial task. Depending on the (usually unknown) optimal shape, the model has to be complete enough as to match the desired target. Yet if it is overdeveloped respect to the final shape, this may lead to multiple modeling solutions giving the same shape, affecting the optimization process.

Three different geometrical parametrizations have been optimized in order to test their suitability to this class of problems. Respectively two, three, and four third order Bézier curves have been used, as sketched in figure 4.2.

A third order Bézier cuve is univocally characterized by four points. This means that in a 2D space there are eight independent scalar variables. It has been decided to use smooth curves to model the geometry. Geometrical first order continuity G^1[1] is imposed. Consider the 2 curves shape in figure 4.2(b), to reach a G^1 condition, points 3, 4, and 5 have to be aligned. This shape is characterized by eleven degrees

[1] An introduction to shape representation is given in chapter 2



(a) 2 curves

(b) 3 curves

(c) 4 curves

Figure 4.2 Geometry models

of feedom, as summed up in table 4.1, where endpoints coordinates are expressed in reference to the global coordinate system $(x, 1)$ and internal points in polar coordinates relative to endpoints, as cleared in figure. In table 4.2, instead, the degrees of freedom

Table 4.1 Points and variables for 2 joined curves

point	variables
1	x_1
2	ρ_2, ϑ_2
3	ρ_3, ϑ_{3-5}
4	x_4, y_4
5	ρ_5, ϑ_{3-5}
6	ρ_6, ϑ_6
7	x_7

for the 3 curves shape of figure 4.2(b) are presented. In the optimization process the values of the design variables are assigned in a weighted random way within a given range. In order to control the endpoints distribution, the abscissas of points 4 and 7 will be given as a fraction of 1 and 10. Eventually, in table 4.3 the the degrees of freedom for the 4 curves shape of figure 4.2(b) is shown. Point 7 sticks to the y axis.

Table 4.2 Points and variables for 3 joined curves

point	variables
1	x_1
2	ρ_2, ϑ_2
3	ρ_3, ϑ_{3-5}
4	x_4, y_4
5	ρ_5, ϑ_{3-5}
6	ρ_6, ϑ_{6-8}
7	x_7, y_7
8	ρ_8, ϑ_{6-8}
9	ρ_9, ϑ_9
10	x_{10}

4.1.3 Optimization process

In literature there is plenty of different optimization techniques. Each one of them best suits for different problems. The scope of this chapter is to implement and test the performances of a general procedure. And with this purpouse in view, the problen is considered completely unknown. Indeed, Fourier equation 4.3 together with is known to have a unique geometry satisfying the overimposed boudary conditions. The method proposed by [42], based on a gradient search is surely one of the most effective.

Table 4.3 Points and variables for 4 joined curves

point	variables
1	x_1
2	ρ_2
3	ρ_3, ϑ_{3-5}
4	x_4, y_4
5	ρ_5, ϑ_{3-5}
6	ρ_6, ϑ_{6-8}
7	y_7
8	ρ_8, ϑ_{6-8}
9	ρ_9, ϑ_{9-11}
10	x_{10}, y_{10}
11	$\rho_{11}, \vartheta_{9-11}$
12	ρ_{12}
13	x_{13}

The geometrical model proposed in this study does not fit a gradient search in a good way. Every design variable affect the geometry in a more global way than a linear piecewise modeling, which anyway suffers a series of dwawbacks such as lack of continuity and the need for a large number of points to describe complex shapes.

Moreover, gradient based algorithms are not robust. Their convergence towads an extremum depends on the starting point, and the process can get stuck in a local optimum point rather than a global one.

Evolutionary algorithms are well suited for exploring the design space of an unknown function, because of their robusness and absence of assumptions about the system under study and no restrictions on the smothness of the objective function.

Each optimization proposed in this chapter is performed using a *Genetic Algorithm* (GA), whose features have been previously introduced in chapter 1. In particular, MOGA-II, the proprietary version implemented in modeFRONTIER©, has been exploited.

The goal of the optimizations is the achievement of a geometry with determined constant surface temperature,θ_s. To reach the goal, an objective function has to be introduced. In practice the *minimum* for an error function is searched. The error is calculated in quadratic norm as follows:

$$\epsilon = \frac{\sqrt{\int_S (\theta - \theta_s)^2 \mathrm{d}S}}{\theta_s \int_S \mathrm{d}S} \cdot 100 \tag{4.9}$$

where θ_s is the desired surface temperature and dS is the elemtentar area of integration. The design variable vector is composed of the curves parameters. The functions to be optimized for the three different geometric models are[2]:

$$
\begin{aligned}
\epsilon &= f_2(x_1, \rho_2, \vartheta_2, \rho_3, x_4, y_4, \rho_5, \vartheta_{3-5}, \rho_6, \vartheta_6, x_7) \\
\epsilon &= f_3(x_1, \rho_2, \vartheta_2, \rho_3, x_4, y_4, \rho_5, \vartheta_{3-5}, \rho_6, x_7, y_7, \rho_8, \vartheta_{6-8}, \rho_9, \vartheta_9, x_{10}) \\
\epsilon &= f_4(x_1, \rho_2, \rho_3, x_4, y_4, \rho_5, \vartheta_{3-5}, \rho_6, y_7, \rho_8, \vartheta_{6-8}, \rho_9, \ldots \\
&\quad \ldots x_{10}, y_{10}, \rho_{11}, \vartheta_{9-11}, \rho_{12}, x_{13})
\end{aligned}
\tag{4.10}
$$

that is generally neither continuous nor connected. Robustness of a GA is affected by a series of parameters. The nature of relations f_i can influence the process. But being them unknown, this is usually an non assessable parameter. The probability associated to the genetic operators, which will be manteined constant throughout the optimization processes as in table 4.4 can influence robustness and convergence rate. Then, the numerousness of the evolving population can influence the process effectiveness. A rule usually followed to decide an initial population numerousness is :

Table 4.4 Genetic operators probability

Operator	value
Selection	0.05
Mutation	0.1
Cross-over	0.45
Dir. Cross-over	0.5

$$
n = max\,(2 \cdot \mathit{(number\ of\ objectives)} \cdot \mathit{(number\ of\ variables)}, 16) \tag{4.11}
$$

The design space is initially sampled with *sobol* algorithm, in order to get a uniformly distributed initial population.

Once a good convergence is obtained, the solution is improved using SIMPLEX algorithm, which is best suited for refinement purpouses.

4.1.4 Results

Two curves parametrization

The two curves shape has been tested with a constant value of the Biot number, $Bi = 1$. an initial population of 40 individuals has been evolved for 15 generations with an initial design space as in table 4.5. In figure 4.3 the history chart of the optimization is presented, while in figure 4.4 the evolution of the shape during the optimization is shown.

During the optimization task the design variable tend to concentrate their values in smaller intervals than the possible span. A second MOGA optimization is performed

[2] the subscript corresponds to the number of curves

Table 4.5 MOGA first run design space

variable	**inf.**	**sup.**	**step**
x_1	-1.2	-0.501	1E-3
ρ_2	1E-3	1.0	1E-3
ϑ_2	10.0	170.0	0.5
x_4	-0.5	0.5	1E-3
y_4	0.0	1.0	1E-3
ρ_3	1E-3	1.0	1E-3
ϑ_{3-5}	-60.0	60.0	0.5
ρ_5	1E-3	1.0	1E-3
ρ_6	1E-3	1.0	1E-3
ϑ_6	10.0	170.0	0.5
x_7	0.501	1.0	1E-3

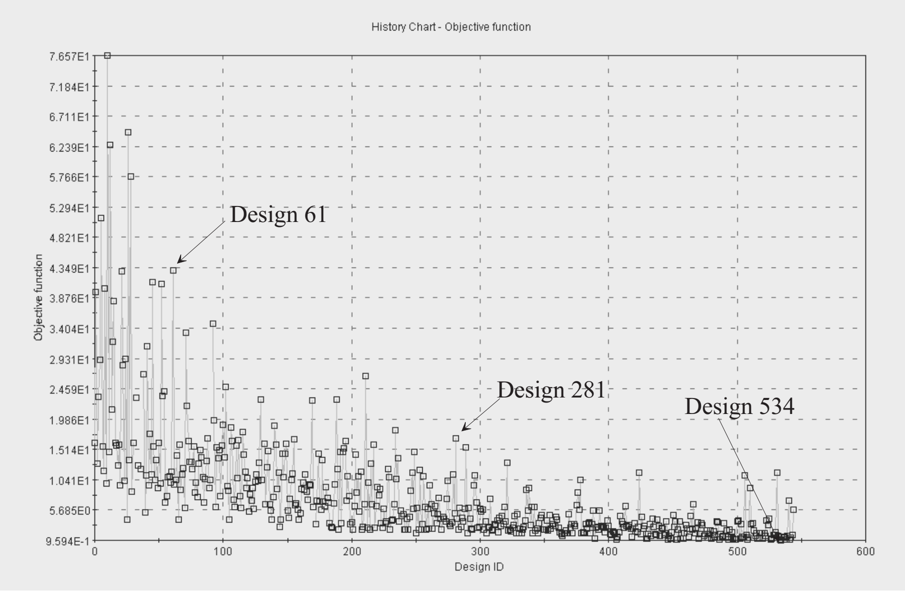
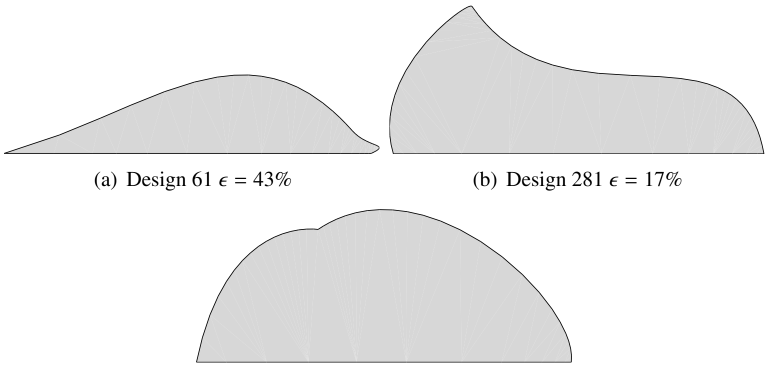


Figure 4.3 MOGA primo lancio, θ = 0,6: *design chart*



(a) Design 61 ϵ = 43%

(b) Design 281 ϵ = 17%

(c) Design 534 $\epsilon = 0,9\%$

Figure 4.4 MOGA first run, θ = 0,6: *design shapes*

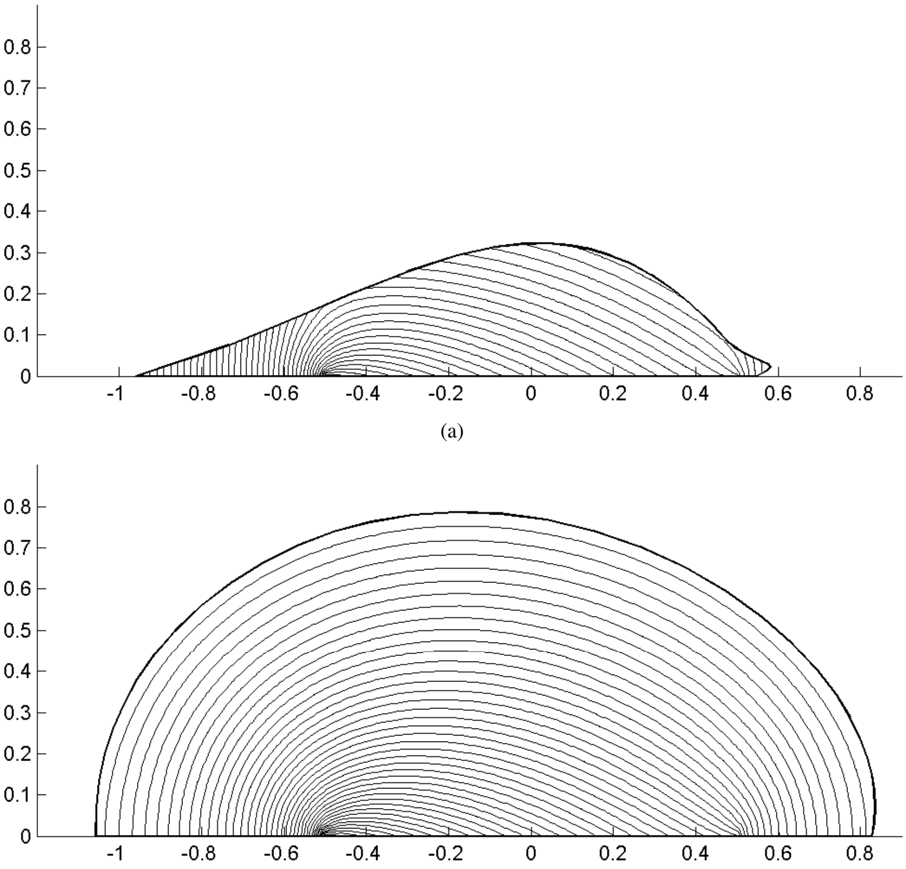
with a shrinked domain space. The initial population is again of 40 individuals. The best of the fisrt optimization plus 39 generated by the sobol algorithm.

At the end of the optimization with genetic algorithm, SIMPLEX is used to refine the solution. The optimization process is summed up in table 4.6, while the best shape is depicted in figure 4.5.

Table 4.6 Optimization evolution

	ϵ value		
σ_s	GA 1	GA 2	SIMPLEX
0.6	0.959	0.761	0.160
0.9	1.244	0.434	0.115

Though a good result has been reached,in figure 4.5,particularly at the right end, it can be noted the shape is hardly able to follow isothermal contours. This parametrization will not be able to produce good results when the Biot number will be changend from constant to a quadraric function.



(a)

(b)

Figure 4.5 2 curves parametrization, $\theta_s = 0.6$: a) Moga first run design 61 with $\epsilon = 43\%$; b) The best design obtained with $\epsilon = 0.160\%$.

Three curves parametrization

In this case, the ouer boundary condition is eq. 4.8, where the Biot number is a quadratic function in x direction.

The optimization process has not led to good results neither for $\theta_s = 0.6$ nor for $\theta_s = 0.9$, as it can be noted in figure 4.6, where the best design for $\theta_s = 0.6$ is shown.

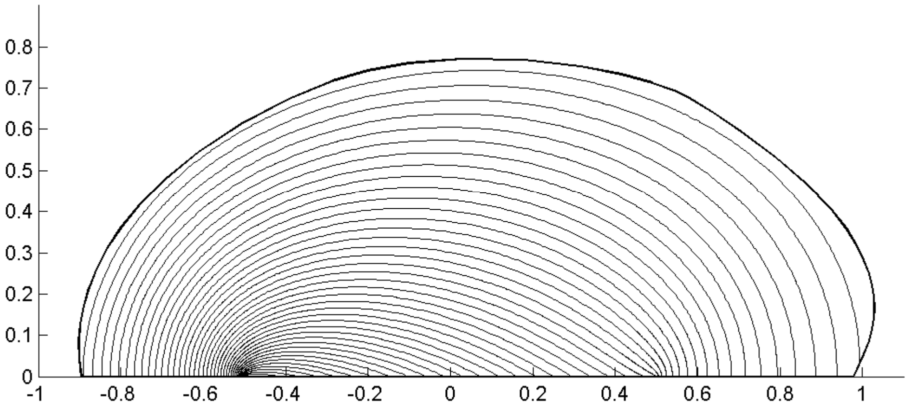


Figure 4.6 3 curves parametrization, θ = 0,6: best design

The failure of this geometric model can be abscribed to two different motivations. A first hypothesis is an objective incapacity of the curve to follow the optimal shape. This thesis is supported by the asimptotic behaviour of simplex algorithm, as depicted in figure 4.7.

Secondarily, the position of points 4 and 7 is free to move all along the surface. During successive optimization processes the values fof the degrees og freedom associated with these points are likely to gather in different zones of the design space, as figure 4.8 clearly shows for parameter x_4 in two successive MOGA optimizations.

This means that different combinations of the parameters can generate similar shapes, thus misleading and slowing down the optimition process. An example in figure 4.9, where an hypothetical optimal shape as in 4.9(a) might be approached as in 4.9(b), with a low value for the error, but with no further posible improvement.

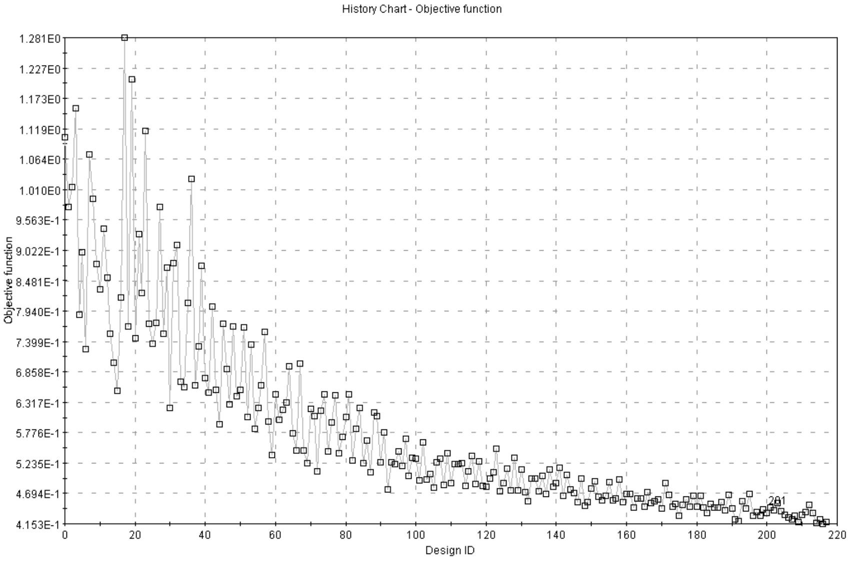
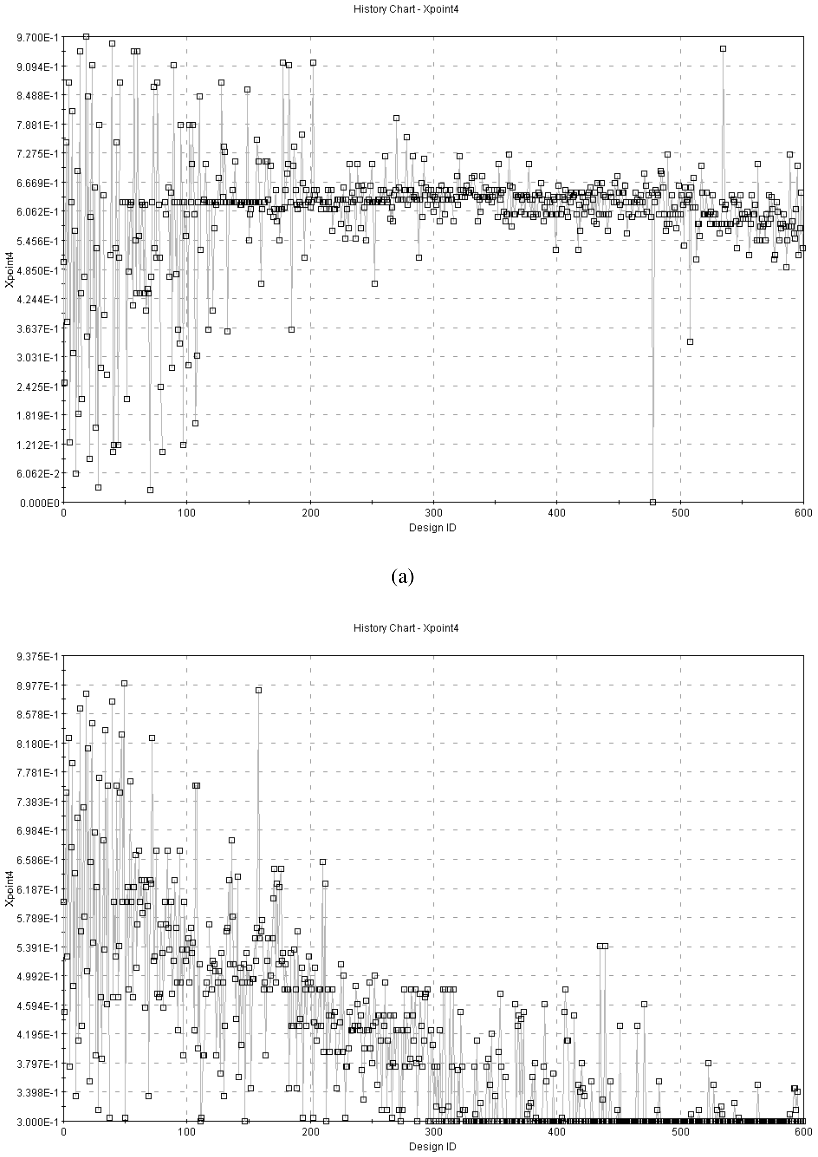


Figure 4.7 3 curves parametrization, SIMPLEX, θ = 0,6: *history chart*



(a)

(b)

Figure 4.8 3 curves parametrization, θ = 0,6, parameter x_4: a) first MOGA run; b) second MOGA run

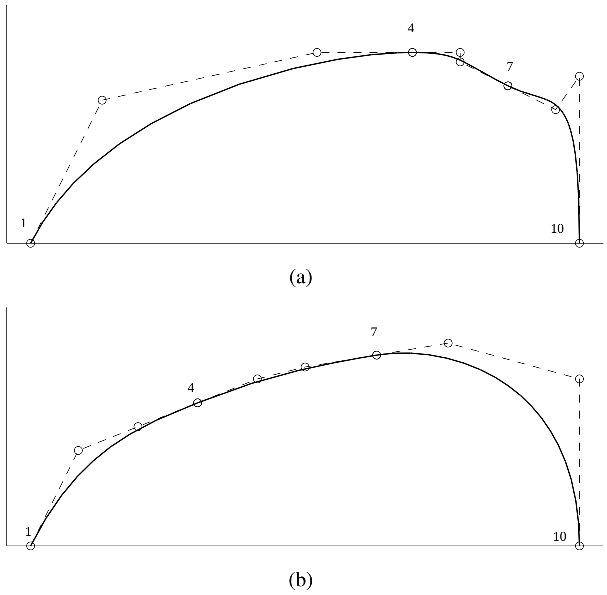


Figure 4.9 Example of a possible shape: a) hypothetical optimal shape; b) design with a good fitness but wrong points distribution

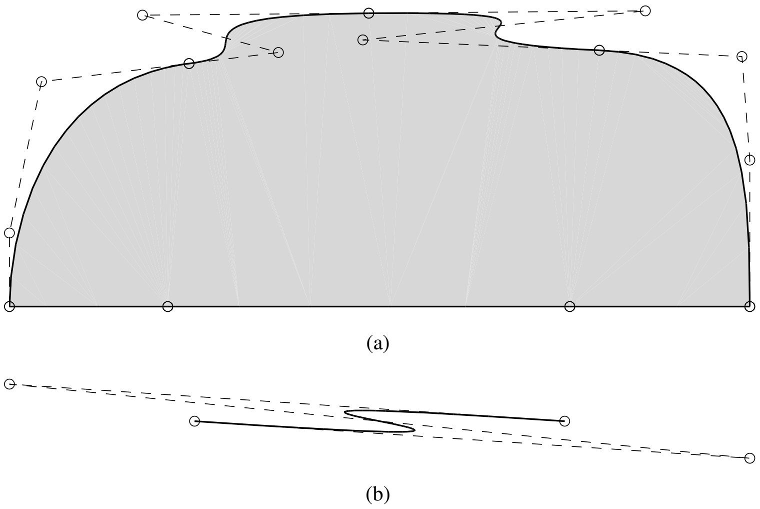


Figure 4.10 Duoble change in convexity: a) MOGA first run, best design obtained; b) the optimization process tend to crush the curve in this way

Four curves parametrization

The boundary conditions of the problem remain unchanged.

In order to avoid the problem arisen with the 3curves paraetrization, the central point of this model (point 7 in figure 4.2(c)) is stuck to the y axis. Moreover, being the problem symmetric respect to the x-axis, there is no heat flux along the axis itself. As a consequence, isothermal lines have to be perpendicular to the symmetry line.

In the previous optimizations, the values for the angles of the endpoints of the curves have been let free, and during the optimization process have tended to the optimal value of 90°. From now on, in order to reduce the number of design variables, the angles ϑ_2 and ϑ_{12} are given a constant value.

During the optimization process the shapes obtained have shown a tendency to form a double change in convexity as in figure 4.10 With the aim to avoid this problem a geometric constraint is applied to each of the four curves. the graphic reference is figure 4.11. The constraint imposed is:

$$\overline{A-B}\cos(\alpha) + \overline{C-D}\cos(\beta) \leq \overline{A-D} \tag{4.12}$$

which penalizes configurations that incline to double convexity.

Given such a constraint, the optimization process leads to optimal results, with almost isothermal shapes for both the desired θ_s, as sketched in figure 4.12 ans summed up in table 4.7.

The good results obtained with this geometry model encourage to use it in conjugate

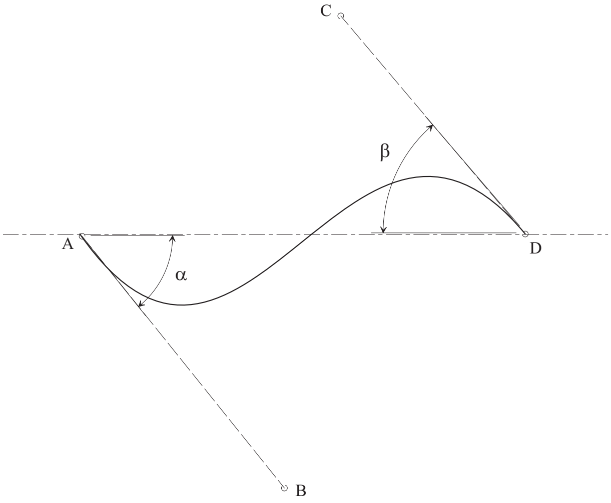
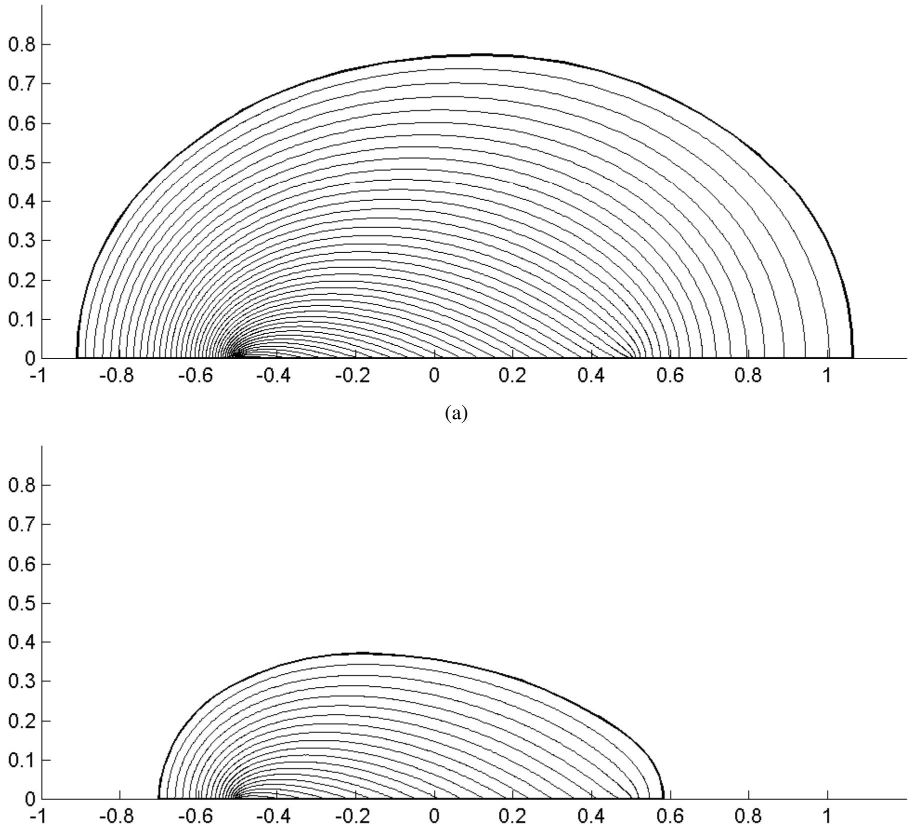


Figure 4.11 Graphic reference for the geometric constraint

optimizations, where the Biot number is not imposed any more, but the convective flow around the solid medium is sollved by means of Navier-Stokes equations.

Table 4.7 Optimization evolution

	ϵ value		
σ_s	GA 1	GA 2	SIMPLEX
0.6	1.536	0.444	0.083
0.9	1.244	0.363	0.089



(a)

(b)

Figure 4.12 4 curves parametrizations, best designs obtained: a) $\theta_s = 0.6$; b) $\theta_s = 0.9$

4.2 2D conjugate problem

This case is an extension of the 2D conductive problems just only exposed in the previous section and has been considered by Cheng and Chang in [43], as well. Dealing with conjugates problems, there is more than one domain. Typically, besides the conductive medium, there is a fluid ambient around it that dissipates its heat. In a solid body, thermal transport is still governed by eq. 4.1. Instead, in a fluid medium, the energy conservation equation becomes:

$$\rho_f c_f \left(\frac{\partial T}{\partial t} + \mathbf{u}^* \nabla T \right) = k_f \nabla^2 T + q^* + \mu \Phi \tag{4.13}$$

in which there is a term depending on fluid field ($\mathbf{u}$) and a term concerning with mechanical dissipation ($\mu\Phi$). The analysis is restricted to steady problems with zero heat sources, and a newtonian and incompressible fluid is considered, in which the dissipation term is neglected. In these conditions, and introducing dimensionless numbers as defined in 3.1.1, eq. 4.13 becomes:

$$\frac{1}{RePr} \nabla^2 \theta = \mathbf{u} \nabla \theta \tag{4.14}$$

where $Re = (\rho U L)/\mu$ is the Reynolds number and $Pr = (\mu\, c_f)/k_f$ is the Prandtl number and U is the reference quantity for the velocity.

The fluid field ($\mathbf{u}$) is obtained by solving mass consevation and momentum conservation (Navier-Stokes) equations. Forced convection under laminar and steady condition is assumed. Thus neglecting time dependent terms and buoyant forces, equations

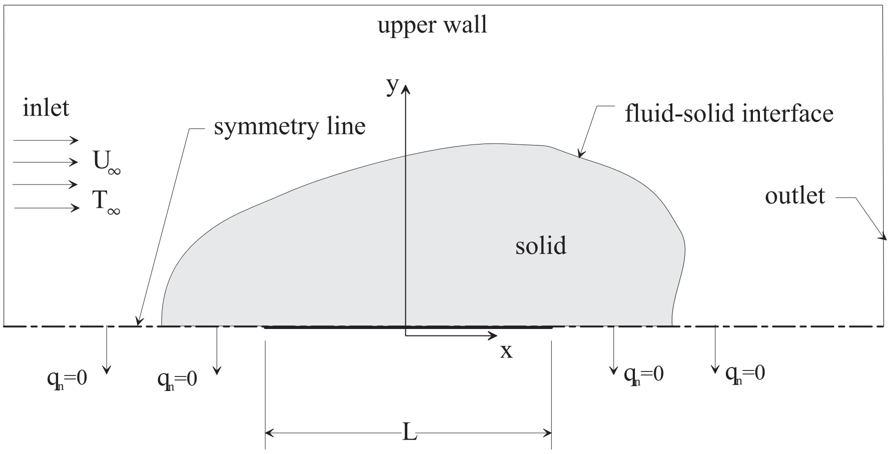


Figure 4.13 2D conjugate problem: general scheme

becomes:

$$\nabla \cdot \mathbf{u} = 0 \tag{4.15}$$

$$\mathbf{u} \cdot \nabla \mathbf{u} = \frac{1}{Re} \nabla^2 \mathbf{u} - \nabla p \tag{4.16}$$

where constant thermo-physical properties are considered.

The reference variables for the dimensionless quantities are:

- The length L of the thin substrate as metric reference;
- The undisturbed velocity U_∞ as velocity reference;
- the difference $T_L - T_\infty$ as temperature referece, where T_L is the temperature on the heated thin surface, and T_∞ is the temperature of the undisturbed fluid.
- ρU_∞^2 as pressure reference

A further parameter is to be introduced to cater for the different conductivities of the solid (k_s) and fluid (k_f) media. the heat flux conserves at the solif-fluid interface, while the temperature gradient is scaled by a factor that depends on the ratio(k_s/k_f) between the solid and the fluid conductivities. The complete sate of equation for energy conservation becomes:

$$\begin{aligned} \frac{k_s}{k_f} \nabla^2 \theta &= 0 && \text{on } \Omega_s \\ \frac{1}{RePr} \nabla^2 \theta &= \mathbf{u} \nabla \theta && \text{on } \Omega_f \end{aligned} \tag{4.17}$$

where Ω_s and Ω_f are respectively the solid and the fluid subdomains.

Thermal boundary conditions are imposed as follows:

$$\mathbf{n} \nabla \theta = 0 \qquad \text{on the symmetry line and at the outlet boundary} \tag{4.18a}$$

$$\theta = 1 \qquad \text{at the heated surface} \tag{4.18b}$$

$$\theta = 0 \qquad \text{at the inlet and upper boudary} \tag{4.18c}$$

Fluid dynamic boundary conditions are:

$$\mathbf{n}\cdot\mathbf{u}=0 \qquad \text{on the symmetry line and at the upper boundary} \tag{4.19a}$$

$$\mathbf{n}\cdot\mathbf{u}=1, \mathbf{t}\cdot\mathbf{u}=0 \qquad \text{at the inlet} \tag{4.19b}$$

$$p=0 \qquad \text{at the outlet} \tag{4.19c}$$

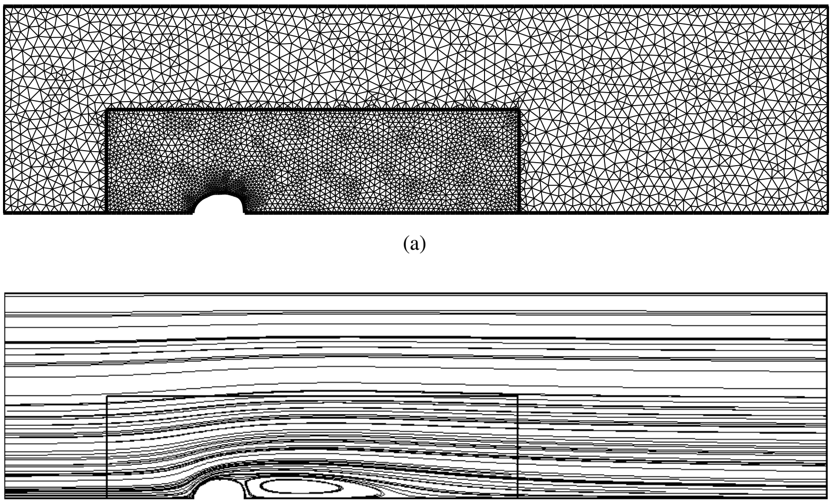
$$\mathbf{n}\cdot\mathbf{u}=0, \mathbf{t}\cdot\mathbf{u}=0 \qquad \text{on the solid-fluid interface} \tag{4.19d}$$

Prandtl number (Pr) is given the value of 0.71, corresponding to atmospheric air. Four different cases have been studied:

- Case **A**: $Re = 20$, $k_s/k_f = 1$, $\theta = 0.6$;
- Case **B**: $Re = 20$, $k_s/k_f = 1$, $\theta = 0.3$;
- Case **C**: $Re = 20$, $k_s/k_f = 5$, $\theta = 0.6$;
- Case **D**: $Re = 40$, $k_s/k_f = 5$, $\theta = 0.6$.

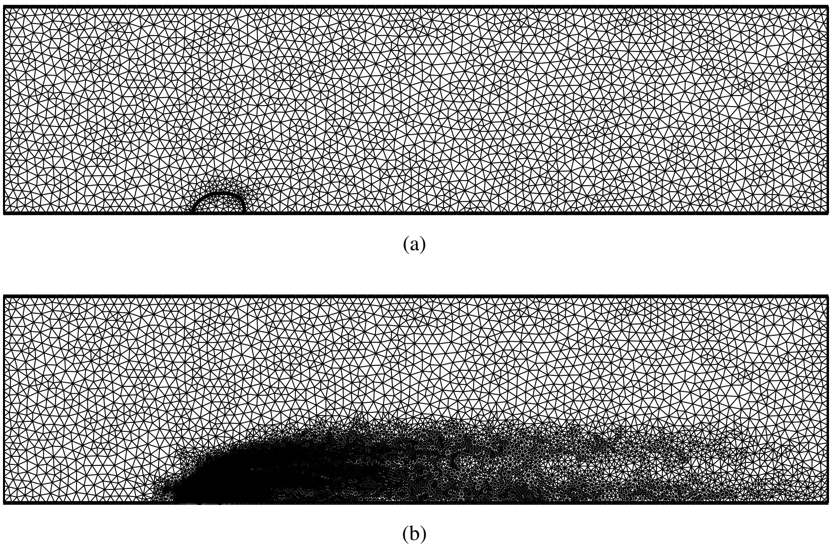
4.2.1 Direct problem solution

The fluid-dynamic and the thermal problems are de-coupled, thus their solution is split in two steps. Continuity and Navier-Stokes equations form a nonlinear system, which


(a)

(b)

Figure 4.14 Fluid-dynamic problem: a) Mesh; b) flow lines



(a)

(b)

Figure 4.15 Thermal problem: a) initial mesh; b) refined mesh

is CPU time consuming. In order to reach good compromise between accuracy and computational cost, a fixed trianguar grid with two different levels of refinement is used, as sketched in figure 4.14

Once the velocity field is obtained, the flow field is interpolated on another grid, used to solve the energy equation. COMSOL implements the mesh adaption feature, which consist in successive refinements of elements characterized by high residual error. A step of adaption is applied in the solution of the energy equation, producing the result in figure 4.15.

4.2.2 Grid and Domain independence test

In external fluid-dynamics, not only the discratization process, but even the domain dimensions have a great influence in determining trustworthy results. In order to define good domain dimensions some test on a cylinder with circular base are performed in order to obtain a value for the drag coefficient, C_d, and the Nusselt number, Nu. The tests are carried out at $Re = 20$. At first the inlet boundary is posed at distance $20L$ from the cylinder axis, being L its diameter. The outlet boundary is at $40L$ from the axis, while the upper boundary is at $20L$ the fluid dynamic mesh consist of 26500 elements, while the adapted thermal mesh consist of 70000 elements. the results obtained for the tho dimensionless numbers are:

$$C_d = 3.24$$

$$Nu = 2.46$$

The same calculation on a shrinked domain where the inlet and the upper boundary are at $10L$ and the outlet at $30L$, with a fluid mesh consisting of 7400 elements and a thermal mesh refined up to 45000 elements, the results are:

$$C_d = 3.27$$

$$Nu = 2.49$$

The twho results do not differ, so the second and more computationally economic solution has been chosen during the optimization processes.

4.2.3 Optimization process

The procedure used to carry the optimization process out is the same one expounded at point 4.1.3. MOGA-II and SIMPLEX algorithms have been used, looking for the minimum value of the error function:

$$\epsilon = \frac{\sqrt{\int_S (\theta - \theta_s)^2 \mathrm{d}S}}{\theta_s \int_S \mathrm{d}S} \cdot 100$$

Subsequently, in some of the MOGA runs the multiobjective nature of MOGA has been exploited in order to make the process easier. A second objective has been introduced: the minimization of the maximum eror on the profile:

$$\epsilon_{max} = max(|\theta - \theta_s|)_S \tag{4.20}$$

4.2.4 Results

Case A

A 45 individuals initial population is generated using sobol algorithm. After three MOGA runs performed on successively shrinked domain spaces the best result, depicted in figure 4.17(a), is far from an isothermal shape. As the history chart in figure 4.16 shows, there are no signs of convergence. Isothermal contousr in figure 4.17(a) suggest the shape should be quite simple. It has been decided to fix two variables in order to reduce the degrees of freedom of the geometrical model. Point 4 and 7 absciassas has been fixed at the middle of their original interval of definition.

After the deglees of freedom reduction, with two MOGA runs the shape in figure 4.17(b) is obtained, that is mich better configuration.

A further refinement obtained with SIMPLEX algorithm, an almost isothermal shape is obtained (figure 4.17(c)).

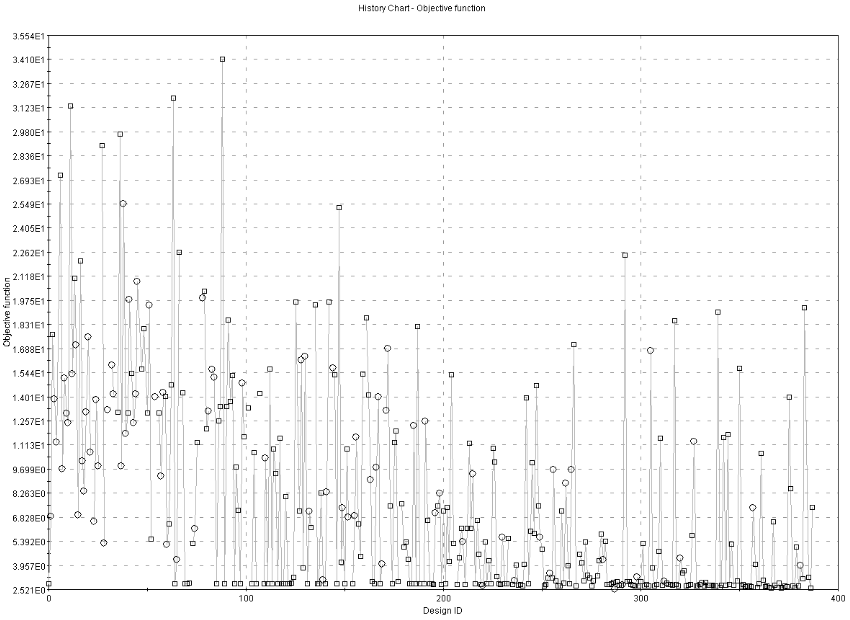
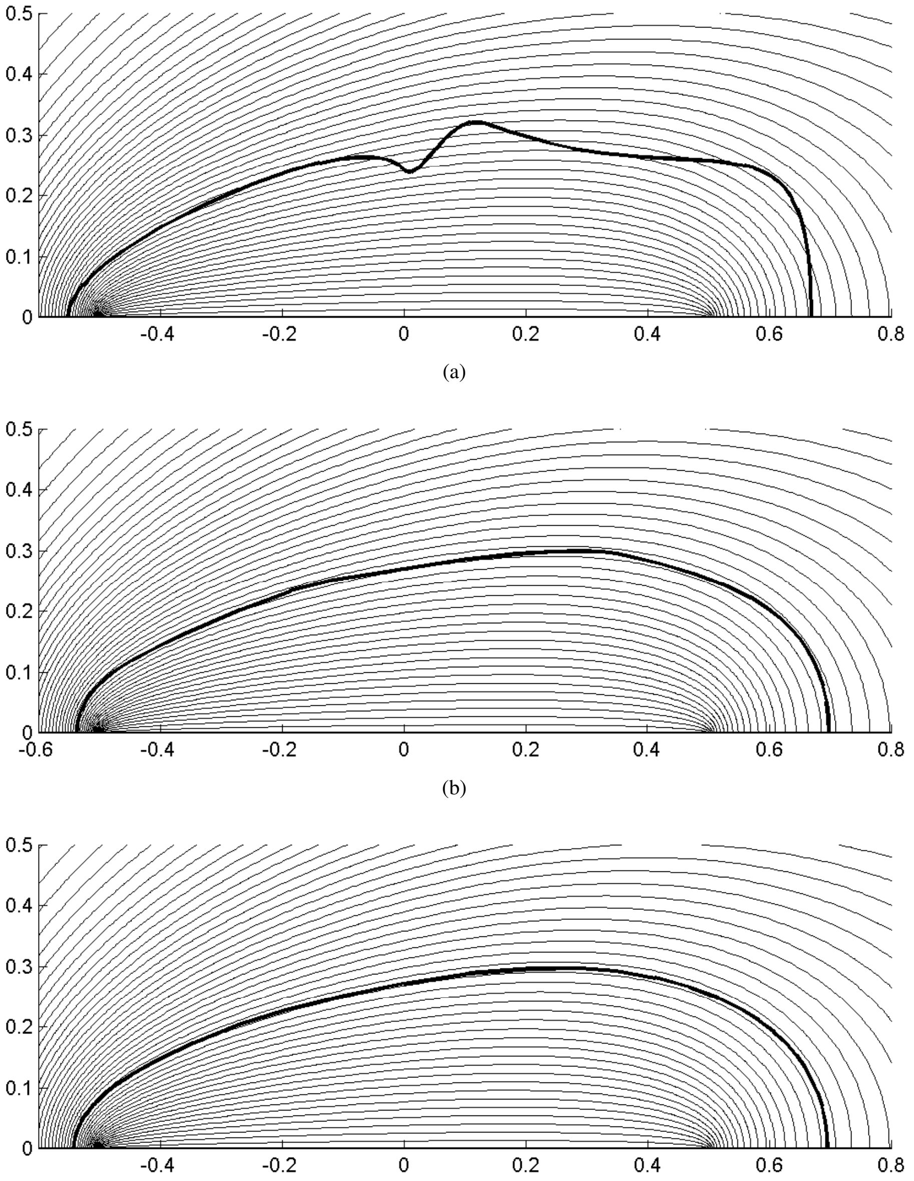


Figure 4.16 Case **A**: MOGA third run unconstrained: *design chart*



(a)

(b)

(c)

Figure 4.17 Case **A**: a) best design after 3 unconstrained MOGA optimization; b) best design after 2 constrained MOGA runs; c) simplex refinement $\epsilon = 0.144$

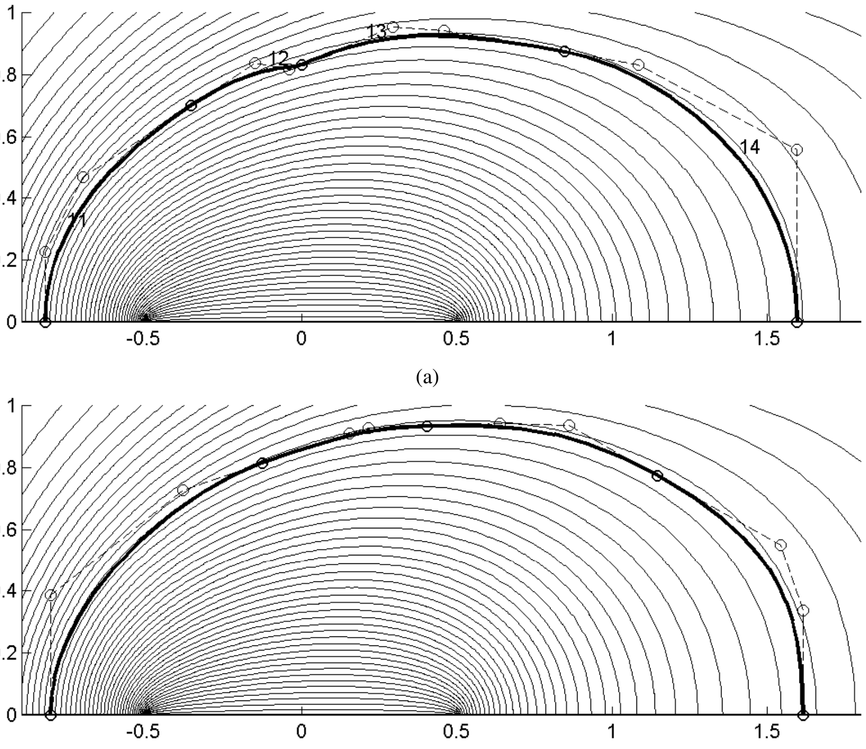
Case B

The target surfce in case **A** is close to the heated substrate. In case **B** the target value is $\theta_s = 0.3$ which should be more distant from the source.

After a series of optimization steps using both MOGA and simplex, the best obtained result is shown in figure 4.18(a). The overall result is pretty good, but the profile is developed downstream, while the midpoint 7 is stick to the y axis. The first two curves approximate a short tract and the second two draw the most of the shape. This causes some difficulties in repruducing an optimal shape.

To overcome the problem, point 7 has been released from the y axis and its abscissa has been posed at the middle of segment $\mathbf{1} - \mathbf{13}$ This operation does not introduce any further degree of freedom, but it reallocates the curves in a better way.

With this expedient the result in figure 4.18(b) is obtained, that fits the target shape much better.



(a)

(b)

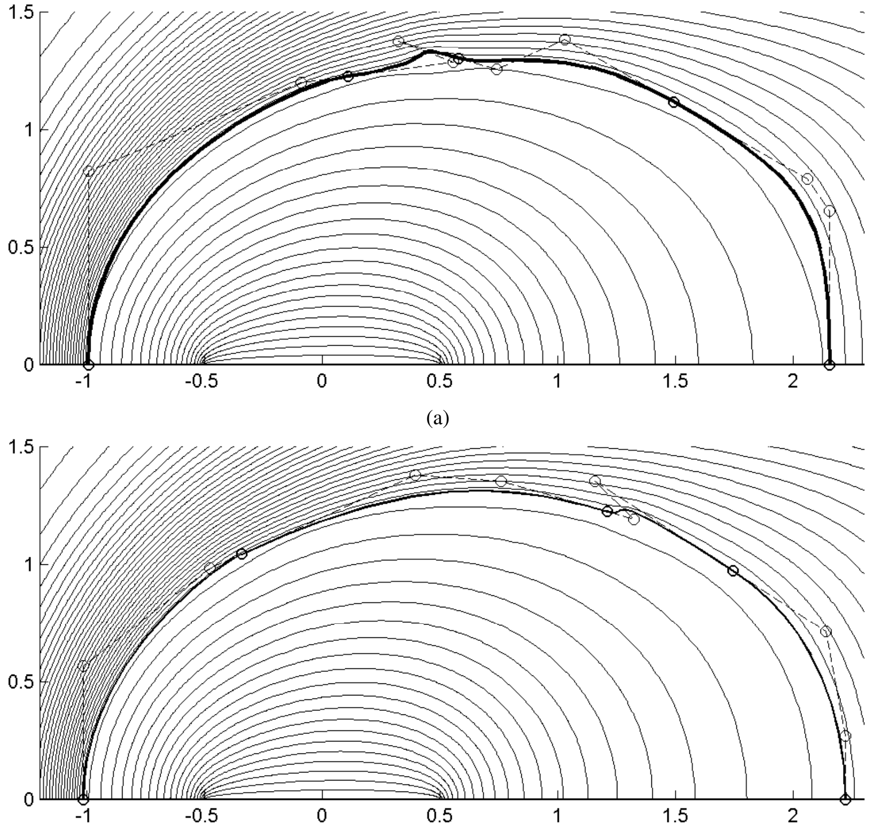
Figure 4.18 Case **B**: a) best design obtained with the original parametrization; best design obtained with the modified parametrization, $\epsilon = 0.41$

Case C

Usually in actual energy transfer situations thermal conductivity of solids is higer than thermal conductivity of the fluids in which their are embedded. In case **C** the conducibility ratio k_s/k_f is changed from 1 to 5.

A first optimization has led to the result depicted in figure 4.19(a), with $\epsilon = 0.42$. Two problem arises. The first is the contour lines suggest the optimal shape is more complex downsteam than upstream. The parametrization may not be appropriate to draw the optimal shape.

Secondarily, due to the change in the solid conductivity, there arise a loss of sensibility towards the objective function. This means that even distorted shapes present pretty isothrmal behaviour.



(a)

(b)

Figure 4.19 Case **C**: a) best design with basic parametrization; b) best design with the introductionof parameter x_7

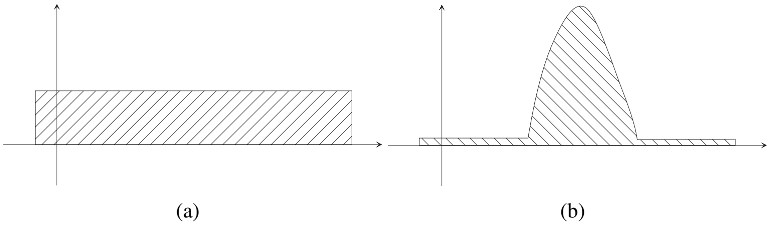


Figure 4.20 examples integral functions: a) regular shape with a constant offset; b) twisted shape with the same integral offset

In order to overcome the fisrt problem a new degree of fredom is introduced. The abscissa of point 7 is free to move between the abscissas of point 1 an 13, that are the endpoints of the curve.

With the new parameter introduced, the best result obtained ($\epsilon = 0.20$) is depicted in figure 4.19(b). The shape has improved, but the lack of sensibility towards the objective function still remains.

Case D

Due to the loss of sensibility towards the objective function both in case **C** and even more in case **D** the optimization process is slow and disturbed.

After four sucessive MOGA runs, for a total number of 2700 evaluated designs, the best design obtained has an error $\epsilon = 0.44$.

In oder to tackle this problem the multiobjetive feature of MOGA has been exploited in order to speed the convergence rate up. Multi objective problems usually deal with conflicting goals, leading to a multiplicity of good solutions as the problems exposed in the previous chapter. In this case, the second objective intriduced is an infity norm, so the two objectives are:

$$\epsilon = \frac{\sqrt{\int_S (\theta - \theta_s)^2 \mathrm{d}S}}{\theta_s \int_S \mathrm{d}S} \cdot 100$$

$$\epsilon_{max} = max(|\theta - \theta_s|)_S$$

which are not conflicting at all. Both objective functions, while minimized, have as a goal the same isothermal surface. What differs is the way in which the goal is evaluated.

Keep in mind figure 4.20, where two different functions with the same integral value are presented. They can be seen as the value of $\theta - \theta_s$ versus the curvilinear coordinate S. While a quadratic norm has the function to draw up the surface to the optimal shape in an integral sense, giving the same fitness to both (a) and (b) configurations, the

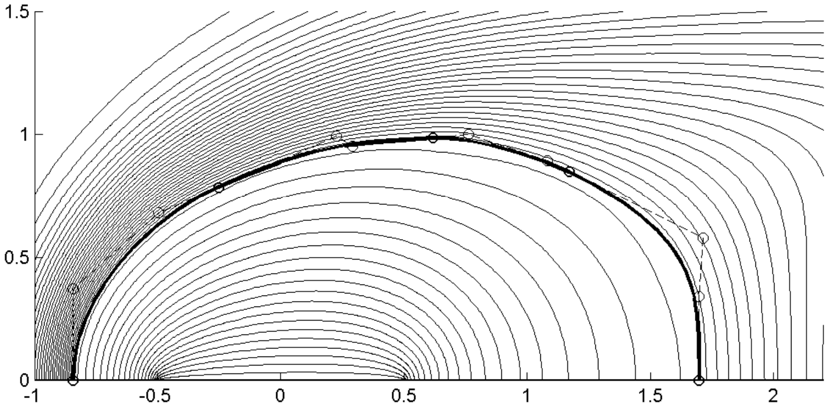


Figure 4.21 Case **D**: best design obtained, $\epsilon = 0.15$

second objective gives a higher fitness to more regular configurations, thus preferring configuration (a).

Introduced the second objective, with only two MOGA runs a shape with error $\epsilon = 0.19$, that refined with SIMPLEX gives the shape in figure 4.21. The scatter chart in figure 4.22 representing the two objectives highlights they are not conflicting,

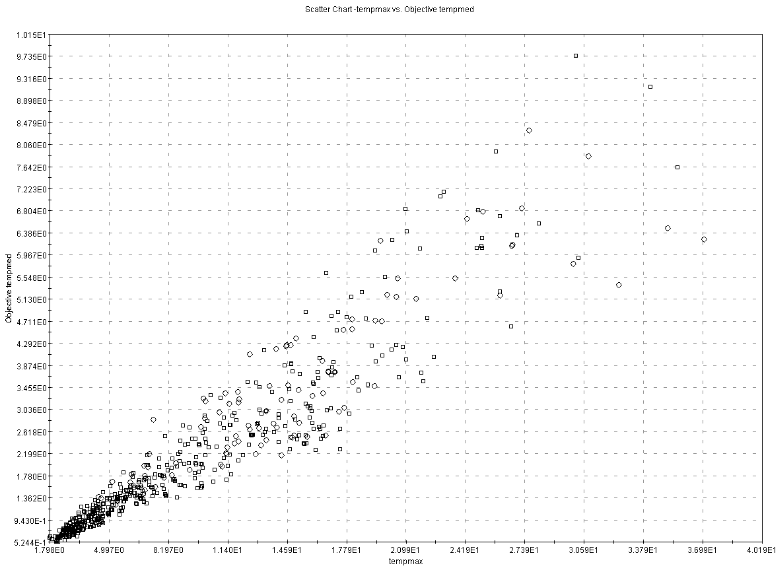


Figure 4.22 Case **D**: Pareto front for the multiobjective optimization

though they operate to stabilize the optimization procedure.

4.3 3D conjugate problem

In the preceding sections the optimization process for a two-dimensional heat transfer problem has been presented. At first a series of geometric models have been tested on a conductive case in order to evaluate their feasibility in reproducing optimal shapes. The four curve parametrization has shown the best in reaching the desired goal, that is to reconstruct isothermal shapes of given temperature.

In real circumstances the two dimensional assumption is not always possible. As for example in the cooling of microchips where the dissipative element does not bear a dimension that dominates the others.

In this section the extension of the two dimensional cases previously studied to a three dimensional form is presented.

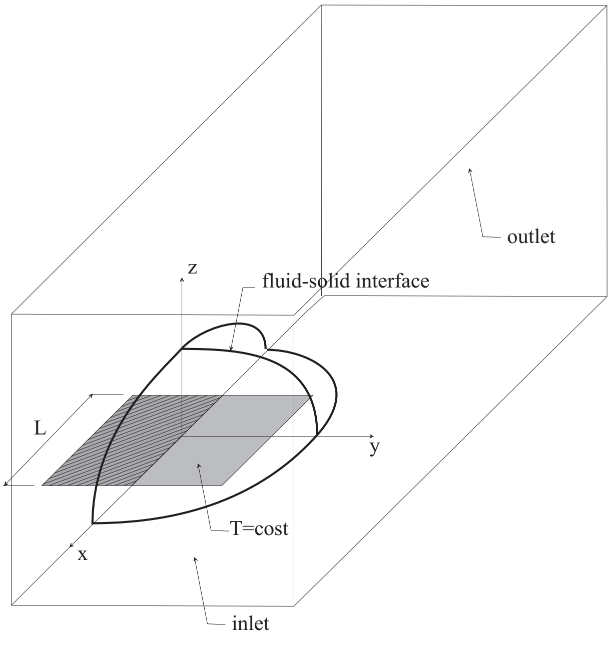


Figure 4.23 3D conjugate problem: sketch of the domain

4.3.1 Problem statement

A square thin heating source is embedded in an omogeneus isotropic conductive body, as in figure 4.23. The body is posed in relative motion in a newtonian and incompressible fluid. Thus forced convective heat trasfer dissipated heat at the solid-fluid interface. The fluid flow is considered laminar and steady state.

The differential equations that dominate the phenomenon are eq. 4.15, eq. 4.16, and eq. 4.17, as in the two dimensional case.

The thin substrate is given a constant temperatue boundary condition and the streamwise direction corespont to the x axis. Thus due to computational cost reasons the double symmetry respect to the xy and xz planes is exploited and only a quarter of the whole domain is numerically modelled. All the four faces parallel to the streamwise direction are given symmetry conditions.

Thermal boundary conditions are imposed as follows:

$$\mathbf{n}\nabla\theta = 0 \qquad \text{on the symmetry planes and at the outlet boundary} \tag{4.21a}$$

$$\theta = 1 \qquad \text{at the heated surface} \tag{4.21b}$$

$$\theta = 0 \qquad \text{at the inlet} \tag{4.21c}$$

Fluid dynamic boundary conditions are:

$$\mathbf{n}\cdot\mathbf{u} = 0 \qquad \text{on the symmetry planes} \tag{4.22a}$$

$$\mathbf{n}\cdot\mathbf{u} = 1, \mathbf{t}\cdot\mathbf{u} = 0 \qquad \text{at the inlet} \tag{4.22b}$$

$$p = 0 \qquad \text{at the outlet} \tag{4.22c}$$

$$\mathbf{n}\cdot\mathbf{u} = 0, \mathbf{t}\cdot\mathbf{u} = 0 \qquad \text{on the solid-fluid interface} \tag{4.22d}$$

Four different cases have been studied:

- Case **A**: $Re = 20$, $k_s/k_f = 2$, $\theta = 0.4$;
- Case **B**: $Re = 20$, $k_s/k_f = 5$, $\theta = 0.4$;
- Case **C**: $Re = 20$, $k_s/k_f = 2$, $\theta = 0.6$;
- Case **D**: $Re = 20$, $k_s/k_f = 5$, $\theta = 0.6$.

4.3.2 Geometry Modelling

If from a physical point of view the phenomenon follows faithfully the two dimensional case and it is governed by the same set of partial differential equations, the geometric modelling is non trivial task.

To draw unknown freeshape surfaces Bézier patches have been used, that are the natural three domensional extension of Bézier curves and whose main characteristic hve been introduced in chapter 2.

COMSOL/FEMLAB implements Bézier surfaces up to the third order. A bicubic surface is univocally determined by a set of 16 points. This implies that in a 3D space a patch is a function of 48 DOFs. When a single surface is not capable of reproducing the desired target surface, there is the possibility to flank more than one patch, provided that appropriate constraints are applied in order to guarantee a smooth junction aT the flanked edges.

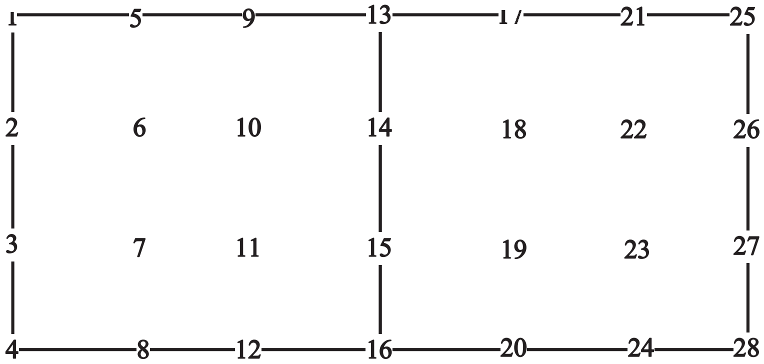


Figure 4.24 Scheme of two flanked bicubic spatches

For this study has a precursory fashion and the computational cost associated to three dimensional CFD coputations is still prohibitive in FEMLAB package, it has been tried to reduce as much as possible the number of degrees of freedom of the geometric model. The aim of the work is to test the possibility to reach coherent results with an optimization procedure that has shown reliable and robust in the other examples proposed in this thesis, with any focus neither on particularly accurate solutions nor on a rapid convergence to the goal.

The chosen parameterization consist of two patches. The general scheme of two joined patches is shown in figure 4.24, where the patches and their control points have been projected on paper. This generic surface is to be constrained as to reach the schematised surface in figure 4.23.

The first step is to collapse the upper edge of the surface to a single point by means of the following constraint:

$$\mathbf{1} \equiv \mathbf{5} \equiv \mathbf{9} \equiv \mathbf{13} \equiv \mathbf{17} \equiv \mathbf{21} \equiv \mathbf{25} \tag{4.23}$$

The second step consist in linking the right and left edges to the *xz* plane and the lowe edge to the *xy* plane:

$$Y_1 = Y_2 = Y_3 = Y_4 = Y_{26} = Y_{27} = Y_{28} = 0 \tag{4.24}$$

$$Z_4 = Z_8 = Z_{12} = Z_{16} = Z_{20} = Z_{24} = Z_{28} = 0 \tag{4.25}$$

To guarantee the smoothness of the entire geometry once constructed by double reflection, the following perpendicularity constrains must be respected:

$$\mathbf{6-2},\ \mathbf{7-3},\ \mathbf{8-4},\ \mathbf{14-13},\ \mathbf{22-26},\ \mathbf{23-27},\ \mathbf{24-28}\ \perp\ xz \tag{4.26}$$

$$\mathbf{3-4},\ \mathbf{7-8},\ \mathbf{11-12},\ \mathbf{15-16},\ \mathbf{19-20},\ \mathbf{23-24},\ \mathbf{27-28}\ \perp\ xy \tag{4.27}$$

Finally, to ensure the smoothness at the joined edge the following terns of points must be collinear and equally spaced.

$$\begin{array}{ccccc} \mathbf{2} & - & \mathbf{1} & - & \mathbf{26} \\ \mathbf{10} & - & \mathbf{14} & - & \mathbf{18} \\ \mathbf{11} & - & \mathbf{15} & - & \mathbf{19} \\ \mathbf{12} & - & \mathbf{16} & - & \mathbf{20} \end{array} \tag{4.28}$$

with such constraints the total number of DOFs reduces from 96 to 22, and points **7** and **23** do not have any.

In table 4.8 the DOFs for each point are summed up.

4.3.3 Optimization process

As in the previous studies, the optimization is performed by means of evolutionary algorithms. An initial population of 50 individuals is generated by means of the sobol algorithm. Then the population evolves by means of MOGA-II algorithm towards higer fitnesses of its individuals. The objective function to be minimized is similar to those introduced earlier in eq. 4.10

$$\begin{aligned} \epsilon = f(&x_1, z_1, \rho_{2-26}, \zeta_{2-26}, \rho_3, x_4, \rho_6, \rho_8, \rho_{10-18}, \vartheta_{10-18}, \zeta_{10-18}, \dots \\ &\zeta_{11-19}, \rho_{12-20}, \vartheta_{12-20}, \rho_{14}, \rho_{15}, x_{16}, y_{16}, \rho_{22}, \rho_{24}, \rho_{27}, x_{28}) \end{aligned} \tag{4.29}$$

where the objective is calculated with eq 4.9:

$$\epsilon = \frac{\sqrt{\int_S (\theta - \theta_s)^2 \mathrm{d}S}}{\theta_s \int_S \mathrm{d}S} \cdot 100$$

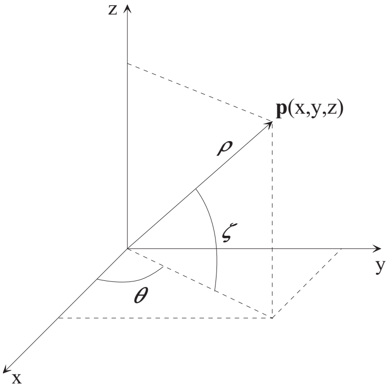
4.3.4 Direct problem solution

The computational domain used is double symmetric respect to xy and xz planes. The streamwise dimension (x coordinate) isposed equal to $40L$,$10L$ upstream and $30L$ downstream. The crossflow dimensions (y and z coordinates) are $10L$. The numerical solution for each individual in the optimization process is carried out by means

Table 4.8 Points and variables for the two joined patches

point	variables
1	x_1, z_1
2	ρ_{2-26}, ζ_{2-26}
3	ρ_3
4	x_4
5	$\equiv \mathbf{1}$
6	ρ_6
7	—
8	ρ_8
9	$\equiv \mathbf{1}$
10	ρ_{10-18}, ϑ_{10-18}, ζ_{10-18}
11	ζ_{11-19}
12	ρ_{12-20}, ϑ_{12-20}
13	$\equiv \mathbf{1}$
14	ρ_{14}
15	ρ_{15}
16	x_{16},y_{16}
17	$\equiv \mathbf{1}$
18	ρ_{10-18}, ϑ_{10-18}, ζ_{10-18}
19	ζ_{11-19}
20	ρ_{12-20}, ϑ_{12-20}
21	$\equiv \mathbf{1}$
22	ρ_{22}
23	—
24	ρ_{24}
25	$\equiv \mathbf{1}$
26	ρ_{2-26}, ζ_{2-26}
27	ρ_{27}
28	x_{28}

In this table Cartesian coordinates are referred to the axes system in figure 4.23. While in the spherical coordinate system, angle θ sweeps a plane parallel to the xy one, and angle ζ sweeps a perpendicular plane to the xy one.



Spherical coordinate are used for the points whose position refers to other points one:
2, **26**, and **14** are relative to **1**;
6 a **2**;
8 a **4**;
22 to **26**;
24 to **28**;
10 and **18** to **14**;
11 and **19** to **15**;
12, **15**, and **20** to **16**.

of COMSOL/FEMLAB software package. COMSOL is a general purpose FE based solver, which can deal with multiphysics problems, but unfortunately it lacks of algoritms dedicated to the solution of CFD problems, which are known to be very high CPU time and memory consuming. The solution of an external flow in a coupled way has shown impracticable on even pretty coarse meshes. For this reason a segregated approach has been implemented in order to solve the direct problem.

In segregated methods the time dependent term is not neglected and a series of advancing steps towards a steady solution is obtained by solving the velocity and pressure variables separately

After a steady solution has been reached for the fluid dynamic field. The velocity components are used in the advective term of energy equation to solve the temperature field with a linear solver.

The projection algoritm

In [46] Gresho gives an overview on fractional step methods and he discusses about the best boundary condition to be given to the intermediate velocities. In [47] Nobile uses an additive-correction multigrid method characterized by second order accuracy in time and space to solve two-dimensional unsteady flows.

In this work a fully implicit scheme is used with first order backward euler time discretization and no subiterations per time step. This scheme is pretty diffusive in time and is not well suited for time dependent simulations, but it represents a good compromise between stability and computational cost while searching steady state solutions.

The spatial discretization is otained by means of a FE unequal order scheme with quadratic lagrange elements for the velocity components and linear ones for the pressure.

The first pace for each time step is to guess a pressure gradient to be used in the velovity update. In fact, were the pressure distribution known, there would not be any particular dificulty in solving Navier stokes equations. Thus, the pressure fied at each new time step can be considered as composed of two componets:

$$p = p^* + p' \tag{4.30}$$

where p^* is the guessed pressure and p' is a correction term to be evaluated. the guessed pressure is usually taken the pressure at the previous time step $p^* = p^n$

Navier stokes equation is modified as follows:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \mathbf{u}^n \nabla \mathbf{u}^* = -\nabla p_n + \frac{1}{Re} \nabla^2 \mathbf{u}* \tag{4.31}$$

where subscript n is the previous time step and $\mathbf{u}^*$ is an intermediate velocity field that in general is not divergence free. The advective term is linearized using old time step velocities, thus allowing a decoupled solution of each velocity component u, v, and w.

Once obtained the intemediate velocity field, the next time step velocity is computed by means of the corerction pressure p':

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{*}}{\Delta t} = \frac{\mathbf{u}'}{\Delta t} = -\nabla p' \tag{4.32}$$

where $\mathbf{u}'$ is a correction velocity field. The aim of the correction pressure is to enforce mass conservation thas is expressed by:

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \tag{4.33}$$

thus applying the divergence operator to eq. 4.32 yields to:

$$\nabla^2 p' = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^{*} \tag{4.34}$$

that is to be solved in order to evaluate the pressure correction contribution. The correction velocities are obtained as follows:

$$\mathbf{u}' = -\Delta t \nabla p' \tag{4.35}$$

and the guesses pressure and velocity fields can be updated to obtain the next time step velocity and pressure fields:

$$p^{n+1} = p^{*} + p' \tag{4.36}$$

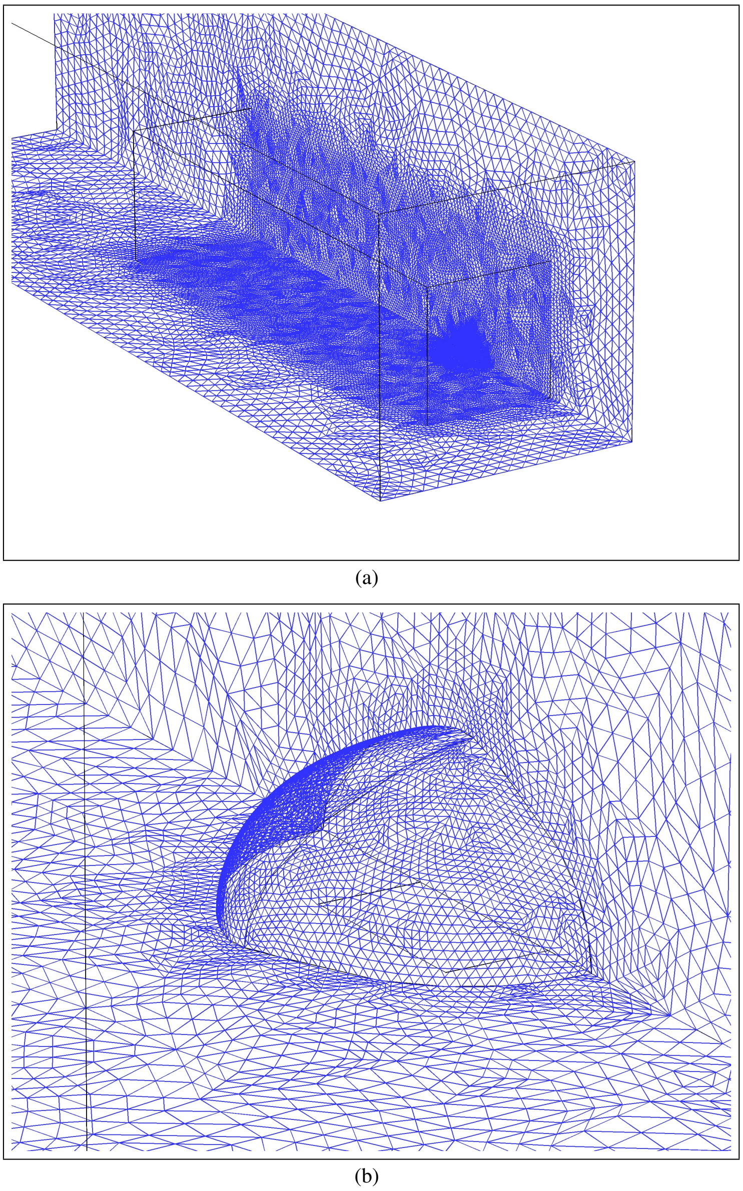
$$\mathbf{u}^{n+1} = \mathbf{u}^{*} + \mathbf{u}' \tag{4.37}$$

At each time step, the boundary conditions for the intemediate velocity field are eqs. 4.22, while omogeneous boundary conditions are imposed for eq. 4.34.

The dimensionless length of the domain in the steamwise direction is equal to 40, while the undisturbed dimensionless velocity is equal to 1. To ensure a steay solution a simutation has to be conducted for at least three times the time for a particle to overpass the domain. A total dimensionless time of 150 has been simutated, with a constant time step $\Delta t = 1$. In these conditions, each simulations has a computational cost of approximatively 4 CPU hours on an AMD Athlon® 64 Processor 3400+ with 2 Gb of RAM.

Spatial discretization

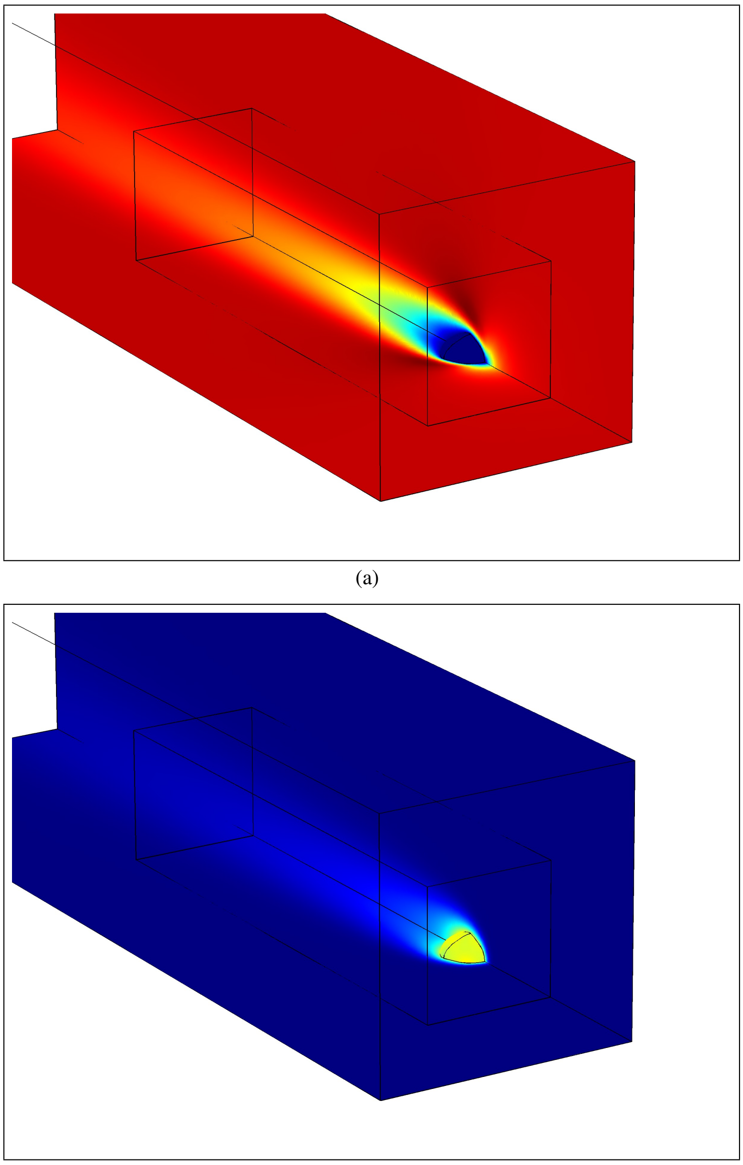
Due to computaional resources reasons, an actual grid independence test has not been made on 3D domains. A validation test of the segregated procedure has been led on a 2D configuration on a circular cylinder for a two level refinement mesh as in figure 4.14, by comparing the Nusselt number obtained for the segregaterd approach with the one obtained with the coupled approach. The overall Nussel number is 2.4943 for the coupled method, while it is 2.4981 for the segregated technique, thus showing a marked accordance.



(a)

(b)

Figure 4.25 Mesh 3D: a) global view; b) zoomed view



(a)

(b)

Figure 4.26 3D solution example: a) fluid field; b) thermal field

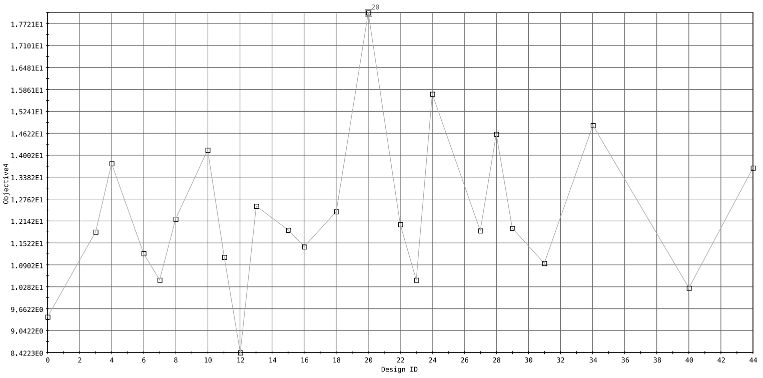


Figure 4.27 Example of initial popilation: the most of the indivuduals do not give fitness information

The mesh used in 3D calculations is of about 40000 tetrahedral elements, with two different zones of refinement. An example of surface mesh is shown in figure 4.25, where the mesh at the solid-fluid interface is highlighted in a zoomed view. An picture of the obtained velodity and temperature fields is presented in figure 4.26

4.3.5 Results

For each optimization process the initial domain space for the design variables has been set as in table 4.9, while due to high CPU time requirements a self adapting version of MOGA has been used to progressively reduce the variables spans throughout the optimization process.

Not every combination of design variables generates feaseble geometries. These individuals do not produce fitness information and are considered error designs. In addition, mesh generation or numerical solution errors occurs, thus augmenting the loss of information. In 3D optimization this aspect is particularly marked. In figure 4.27 an example of initial population is given, where more than the half of the individuals do not have fitness information. Nevertheless, the procedure shows robust and the optimization process moves forward. In such conditions where the failure of the numerical computation has high probability, gradient-based optimization techniques are unsuccessfull. For the exploratory nature of this study, the geometric model of the solid fluid interface has been kept at a minimum of complexity in order to reduce the high number of variables and the heaviness of the optimization process.

Figure 4.28 shows the history chart for the optimization of case **D** where a total number of 882 individuals have been evaluated.

Overall the optimization process has led to good solution. In figures 4.29 it is depicted an intermediate configuration obtained in case **D**, while the best individual

Table 4.9 Desaign variables definition interval

variable	inf.	sup.	step
x_1	-0.3	0.3	1E-3
z_1	0.3	1.5	1E-3
ρ_{2-26}	5E-2	1	1E-3
ζ_{2-26}	-60	60	1E-3
ρ_3	5E-2	1	1E-3
x_4	-1.5	-0.6	1E-3
ρ_6	5E-2	1	1E-3
ρ_8	5E-2	1	1E-3
ρ_{10-18}	5E-2	1	1E-3
ϑ_{10-18}	-50.0	50.0	0.5
ζ_{10-18}	-50.0	50.0	0.5
ζ_{11-19}	-50.0	50.0	0.5
ρ_{12-20}	5E-2	1	1E-3
ϑ_{12-20}	-50.0	50.0	0.5
ρ_{14}	5E-2	1	1E-3
ρ_{15}	5E-2	1	1E-3
x_{16}	-0.5	0.5	1E-3
y_{16}	0.5	1.5	1E-3
ρ_{22}	5E-2	1	1E-3
ρ_{24}	5E-2	1	1E-3
ρ_{27}	5E-2	1	1E-3
x_{28}	0.56	1.2	1E-3

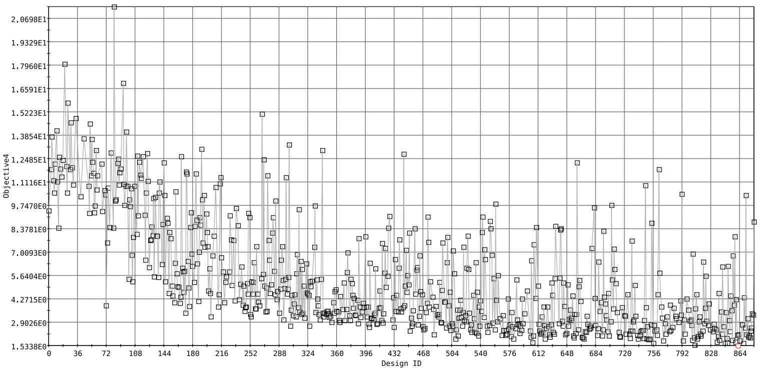


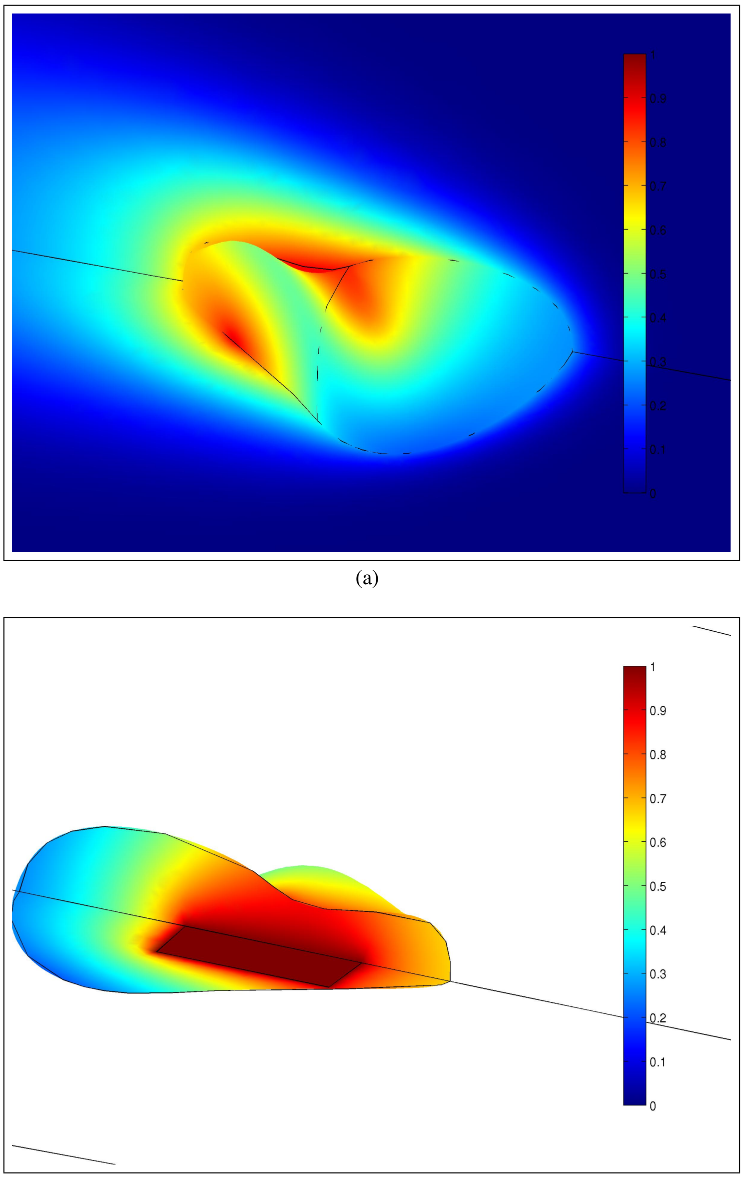
Figure 4.28 Optimization history chart: $\theta_s = 0.6$, $k_s/k_f = 5$

is shown in figure 4.30. The same for case **A** in figures 4.31 and 4.32. The only optimization in which a uniform temperarue has not been reached is case **C**, shown in figure 4.33. In this latter case, the target surface is too close to the thin substrate and the geometric model imposed has not been capable of reporducing the correct shape, settling the error to the value of 10%

Response Surface Methodology

The computational effort required to lead each optimization process has been quite high. Each single indivudual requires about 3 hors of CPU time. Thus, a whole optimization of about 800 configurations has taken about 3 months. In order to reduce this huge ammount of time a test has been made in order to verify the applicability of metamodels in virtually predicting good solutions. Four responce surface have been created for case **D** using Kriging method: the first using the first (RSM1) generation as interpolation basis and 10 k-nearest points, the second (RSM2) using the first 100 feasible evaluated design and 10 k-nearest points, the third (RSM3) using the first 100 feasible design and 50 k-nearest points, and the last one (RSM4) interpolating the fisrt half of the database and 50 k-nearest points. The whole dataset has been evaluated and compared with the actual evaluations. The results are graphed in figure 4.34. The results are not encouraging. None of the four responce surfaces is able to match the non-interpolated designs.

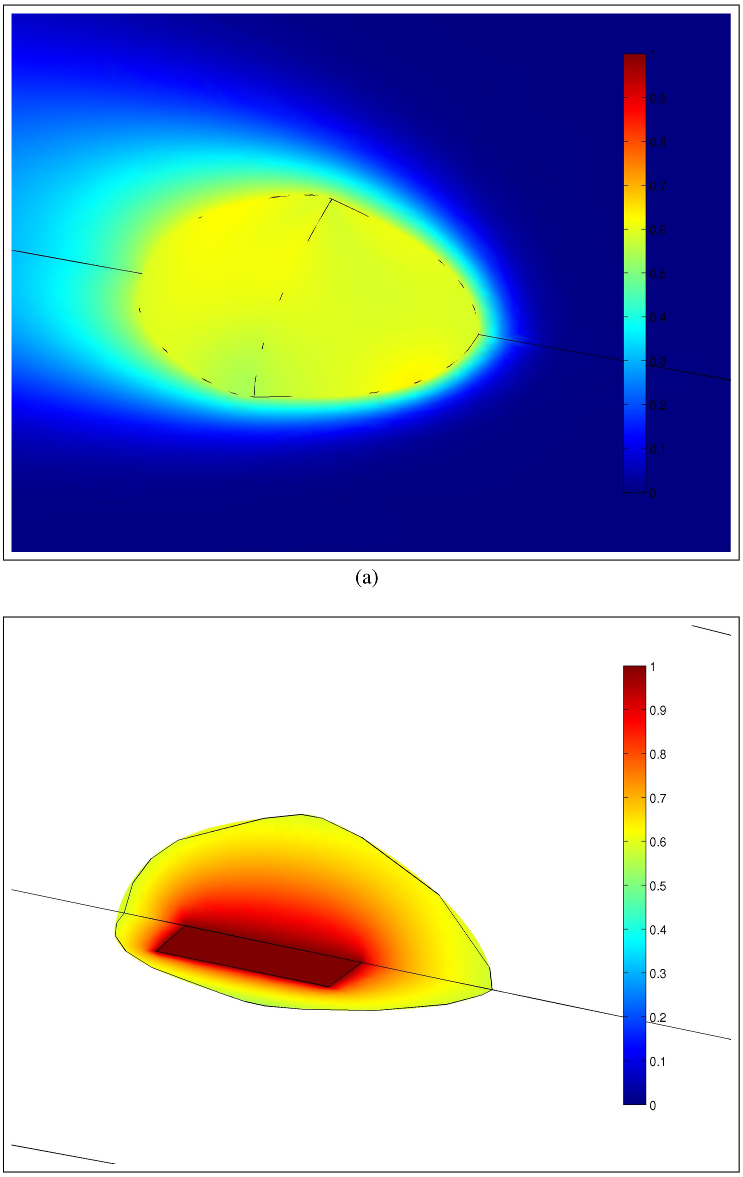
The failure of RSM, even in the fourth case in which a huge and sparse dataset has been used to create the meta-model might be due to the fact that shape optimization is an infinite-dimensional optimization problem, while a finite-dimensional geometric model is used to draw the surfaces. In particular, the design variables of the optimization are Bézier points, each of them influencing the surface shape for a wide truct, and consequently affecting the fluiddynamic and thermal behaviour in a complex way. In these conditions, the use of meta-models appears useless.



(a)

(b)

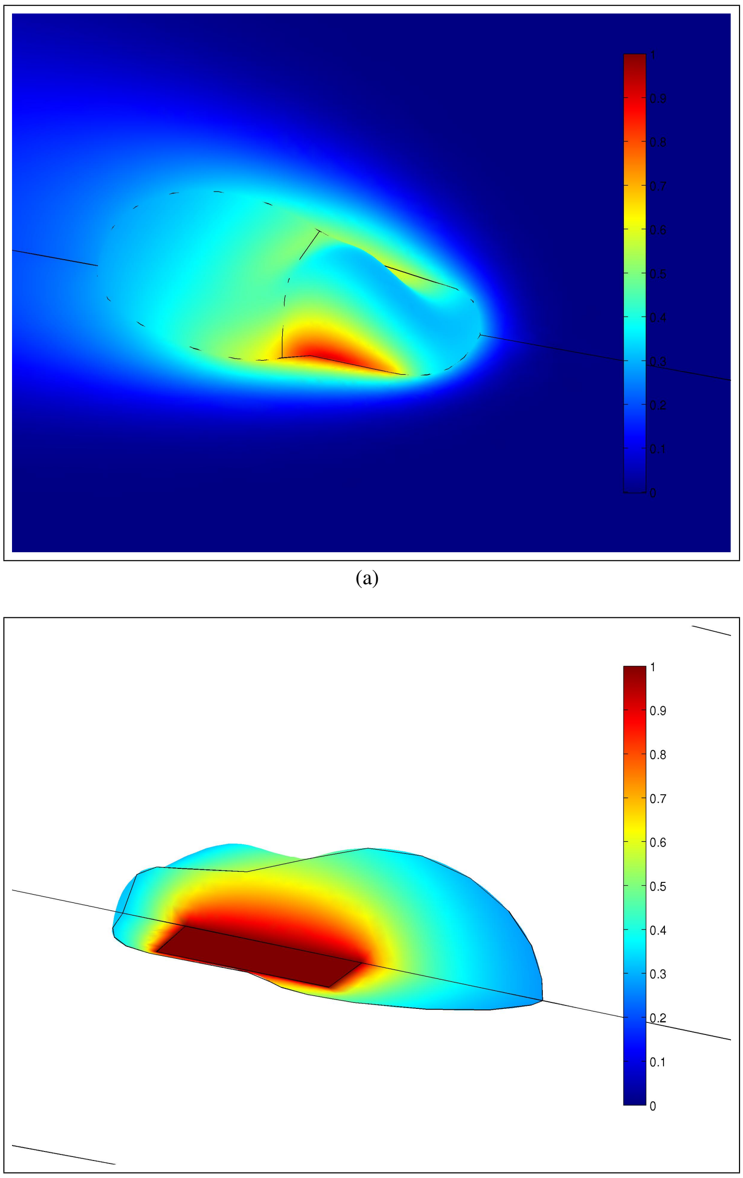
Figure 4.29 Target temperature 0.6, $k_s/k_f = 5$, bad shape: a) surface view; b) core view



(a)

(b)

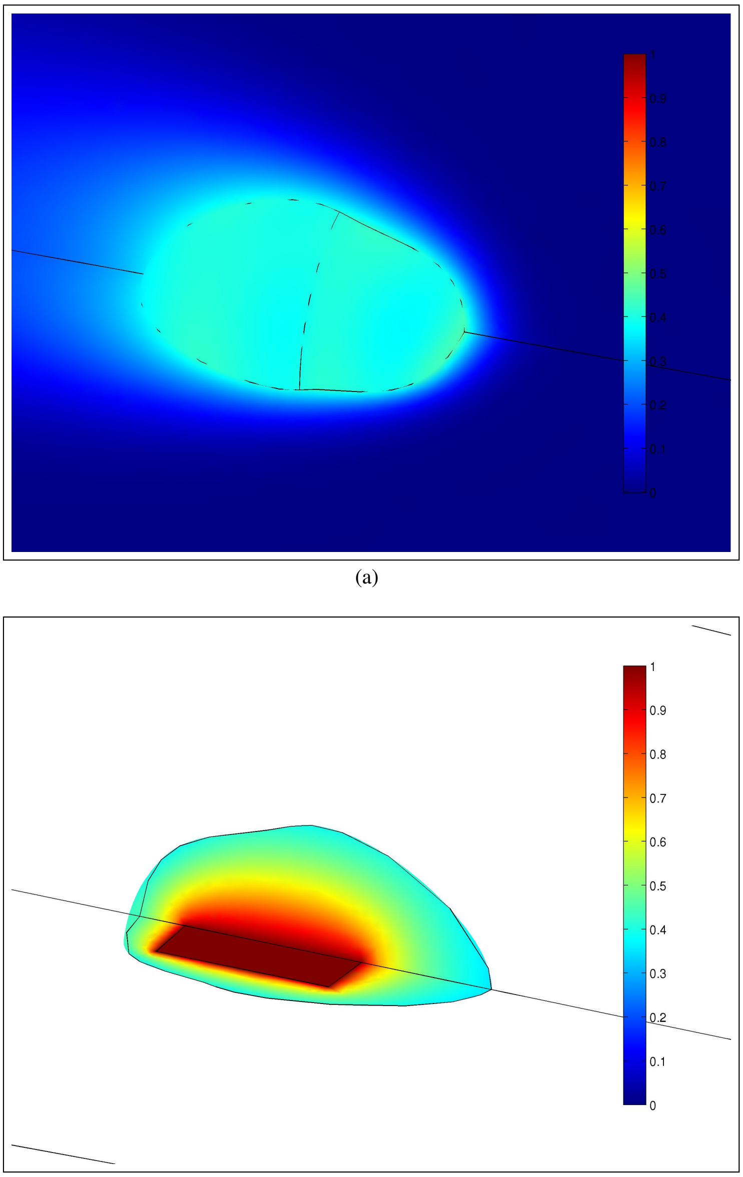
Figure 4.30 Target temperature 0.6, $k_s/k_f = 5$, best shape obtained, $\epsilon = 1.5$: a) surface view; b) core view



(a)

(b)

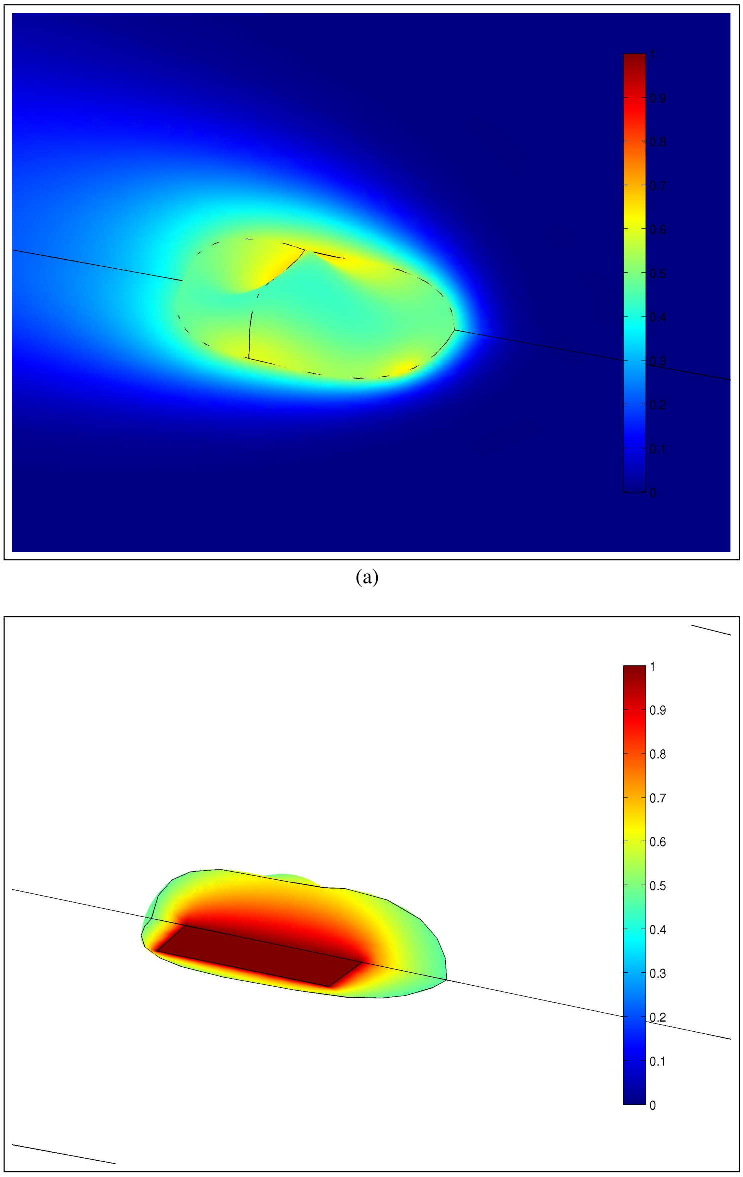
Figure 4.31 Target temperature 0.4, $k_s/k_f = 2$, bad shape: a) surface view; b) core view



(a)

(b)

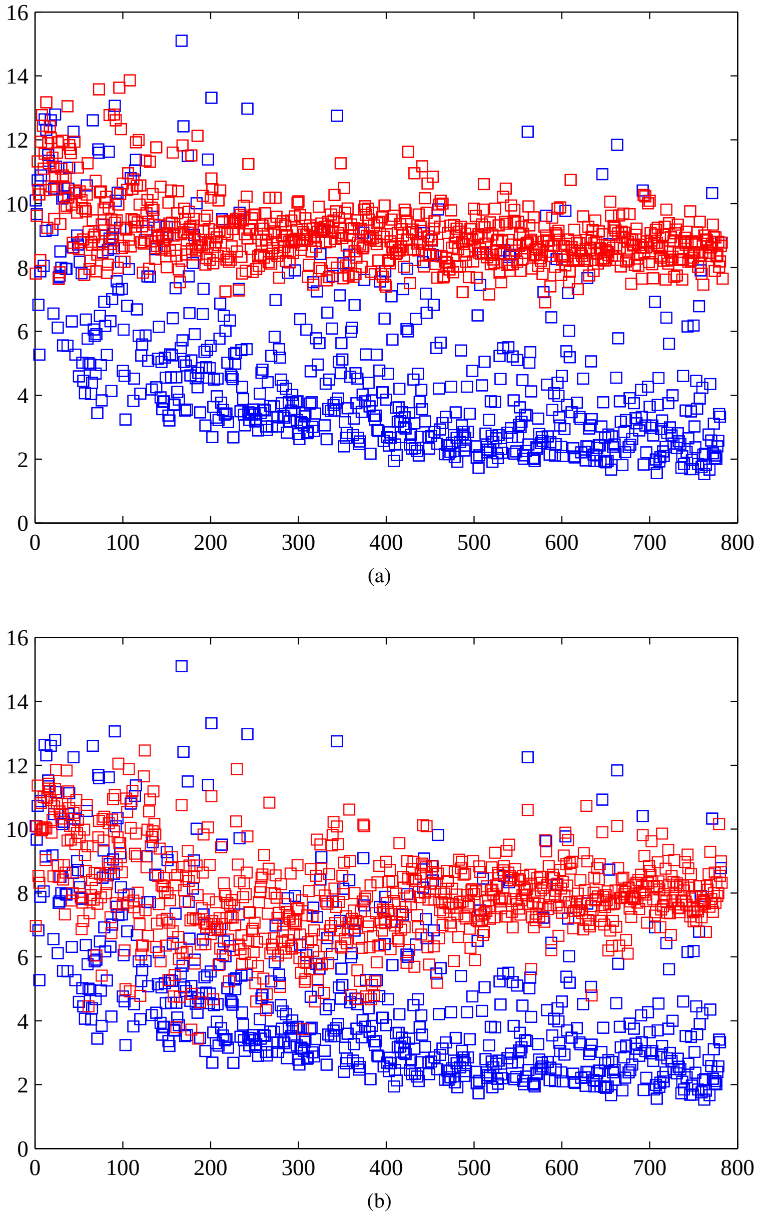
Figure 4.32 Target temperature 0.4, $k_s/k_f = 2$, best shape obtained,$\epsilon = 2.3$: a) surface view; b) core view



(a)

(b)

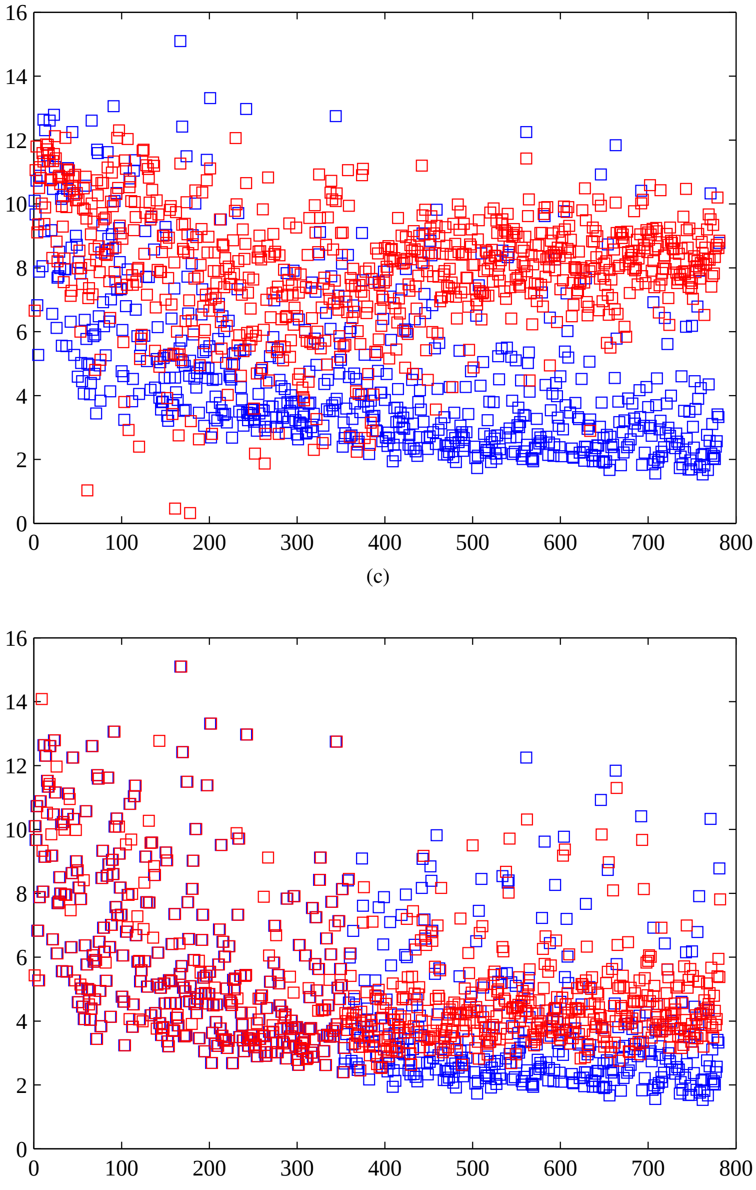
Figure 4.33 Target temperature 0.4, $k_s/k_f = 2$, best shape obtained,$\epsilon = 2.3$: a) surface view; b) core view



(a)

(b)

Figure 4.34 RSM evaluation □ are real designs and □ are virtual designs : a) RSM1; b) RSM2; c) RSM3; d) RSM4



(c)

(d)

Figure 4.34 . . . Continued

Low Consumption Buildings

Chapter Five

Night ventilation cooling techniques

A significant part of the primary energy demand in industrialized countries is due to space heating and cooling in buildings. Furthermore, especially in Europe, the use of HVAC systems is becoming highly popular. Thus, the development of efficient cooling techniques is a very important research task to prevent an uncontrolled energy consumption increase. Night ventilation is a passive cooling technique that can significantly reduce the cooling loads and energy requirements, but a trade off must be made between energy cost savings and zone thermal comfort.

In this scenario the aplication of evolutionary multi-objective techniques can be helpful in developing optimized cooling systems.

In this chapter, it is shown the coupling of modeFRONTIER, an optimization tool, and ESP-r version 10.12, a building simulation code, in order to assess the impact of different parameters on the effectiveness of night ventilation in reducing energy requirements for the climatization of an office building.

5.1 Introduction

The constant increase use of air conditioning systems in both office and residential buildings in industrialized countries is a major factor in causing high peak electricity demands during working hours. Moreover, this trend leads to a significant increase in primary energy consumption required for space climatization. Therefore energy saving strategies must be sought in order to guarantee healthy conditions with a sustainable environmental impact.

Passive cooling systems, as an alternative to mechanical cooling systems, have been used for years but in warm climates they are not able to maintain specified conditions for thermal comfort. Nevertheless natural cooling can be helpful when outer temperatures are low. This condition is met in the swing season or during the night. In the former case the zones can be directly ventilated during working hours, in the latter case the cool external air can be used as an heat sink to lower the temperature of the building's fabric during the night.

The effectiveness of night ventilation has been explored by various authors using both experimental data and building simulation codes, Blondeau [48] analyzed experimentally and numerically the effect of night cooling in summer conditions identifying the temperature difference between inside and outside air as the driving potential for energy removal, Givoni [49] reports experimental data and presents a formula for predicting the expected indoor maximum temperatures, the potential and the impact of the main parameters are investigated in [50]. Thermal performance of various ventilation strategies aimed at reducing the peak power demands, are reported in [51], among the alternatives the coupling between intensive night ventilation and daily active cooling resulted in a substantial reduction of peak loads and total energy consumption while maintaining a controlled internal temperature. In [52] Pfafferott et. al. carried a comparison between data obtained during a long-term monitoring experiment with those obtained with a building simulation program showing that the simulation program, if run with consistent input parameters and boundary conditions, can give accurate results .

The impact of night ventilation on energy consumption is affected by climate, building and control parameters. Nowadays the use of complex building simulation programs allows wide-ranging detailed analysis. Buildings can be linked to PID controlled HVAC models, Navier-stokes calculation can be perfomed to achieve detailed air movement patterns and convective heat transfer loads , moreover radiancy calculations permit radiant energy ballances and illuminance conditions analysis. The use of these softwares can become CPU time consuming in performing a whole year simulation. Moreover, due to modelling issues, the cost functions associated to the problem can be discontinuous. These considerations should be taken into account while choosing the appropriate optimization scheme. Design of buildings is a naturally multi criterion optimization task were always a trade-off has to be made between occupant's thermal comfort and operating costs. So it is natural to couple optimization tools with building simulation codes. In particular, in this work genetic algorithms have been exploited in order to assess the potential of night ventilation in two different mediterranean clitame towns: Rome and Trieste. Two representative days have been tested for each climate: one in July in full summer conditions, one in May, where there are milder temperatures.

The application of evolutionary algorithms to energy saving problems is a newborn reseach field. Nevertheless, in literature there are some interesting works on this subject. In [53] Wright et al. show a two objectives optimization on a single zone HVAC system accomplished with a multi objective genetic algorithm, Huang and Lam [54] used an adaptive learning algorithm based on Genetic Algorithms (GA) for the automatic tuning of a Proportional, Integral and Derivative (PID) controller in HVAC systems, in [55] Wetter and Wright carried out a comparison of deterministic (classical) and probabilistic optimization algorithms on non-smooth optimizations underlining

Table 5.1 building's envelope thermal characteristics

	Transmittance U W/(m^{2}K)	exposed mass m kg/m^2
External wall	0.42	88
Internal wall	1.79	56
Floor	0.6	424
roof	0.6	101
windows	2.6	–

the difficulty in reaching a good solution with gradient based algorithms.

5.2 Problem Description

In this work a Multi Objective Genetic Optimization (MOGA) has been carried on the highlighted part of the three level office building of Figure 5.1. Each of the three zones has a roof surface of 25 square meters and is 2,7 m high, building's envelope thermal characteristics are described in Table 5.1. absorptance for solar radiation and emittance are respectively 0,5 and 0,9 for internal and external surfaces, the windows have a total solar transmission of 0,4. Mediterranean climatic conditions have been used for assessing the efficiency of night ventilation, two sites has been selected, Roma and Trieste. The simulations have been performed for two characteristic days, one in July to represent conditions with high thermal loads, and one in May characterized by lower ambient temperature and solar irradiation. In figure 5.2, the daily ambient temperature profiles for Rome and Trieste are graphed. An air conditioning system has

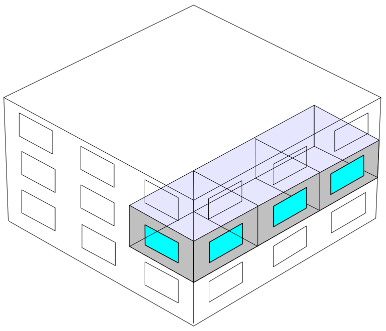
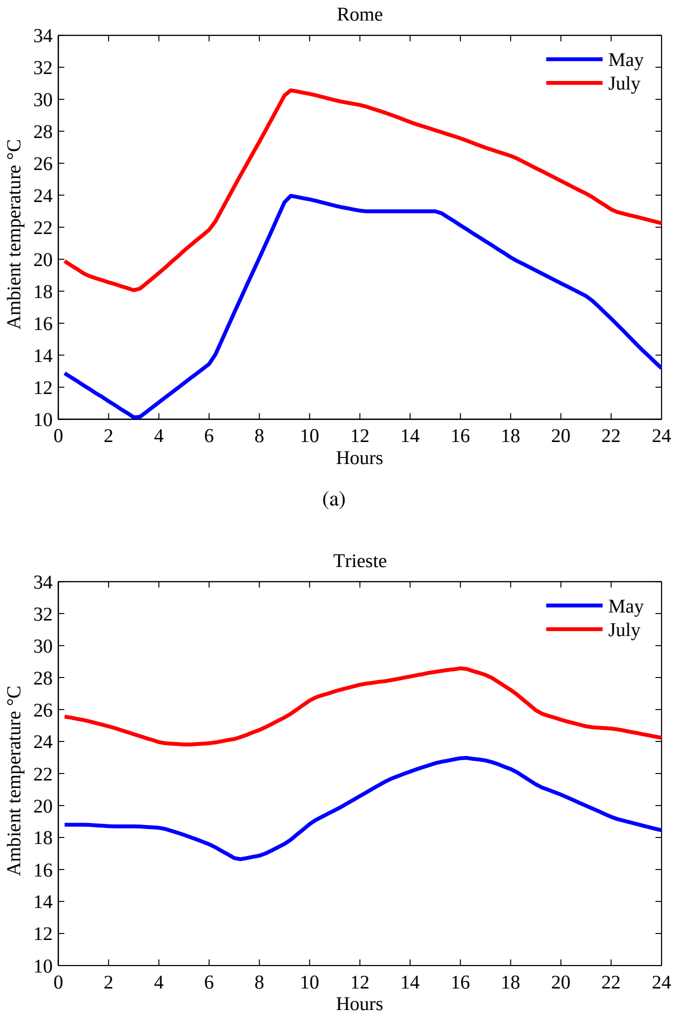


Figure 5.1 Three levels office building with the highlighted analyzed part



(a)

(b)

Figure 5.2 Daily ambient temperature profiles used in sumulations: a) Rome climate; b) Trieste climate

Table 5.2 Design variables for the optimization process

Variale	Span
ΔT	[0 - 10]
Mass flow	[0 - 0.5]
Cooling set point	[20 - 26]
Max RH	[50 - 70]
Min RH	[30 - 50]

been considered operative during working hours from 07:00 to 18:00, while outside this period only ventilation is active when the difference Δt between inside and outside air temperature is greater then a prescribed set point value. During work time the external air change rate is 1 h^{-1}, while internal sensible and latent loads are respectively 350 W and 100 W.

Volumetric flow rate and temperature difference between indoor and outdoor drive the energy removal from the building's fabric. An optimization of these parameters is required because an increment of volumetric flow rate also increases the fan electricity demand, so diurnal savings can be nullified by an increased overall cost, while high temperature differences can be obtained only for short time periods.

Night ventilation can affect healthy conditions in the served areas because of the cooler internal walls, with a favorable effect on operative temperature since comfort conditions can be obtained with higher internal air temperatures and reduced power and energy requirements. Therefore internal air temperature has been used as an optimization parmeter, and to guarantee healthy conditions inside climatized areas a 10% mean PPD constraint throughout working period has been introduced. A side effect of Ventilation is the reduction of temperatures during the first part of the day when the cooling plant starts up, during this period minimum temperatures and higher values of PPD can be encountered.

In this work an ideal controller that injects the required energy to maintain specific conditions has been used to represent the plant, this way the description of specific plant components has been avoided. The minimazion of the energy requirement for maintaining a specified temperature and relative humidity and the energy consumed by the ventilation are considered as two sparate objectives of the optimization, to compute the latter a typical specific power of 0.35 $W/(m^3h^{-1})$ has been used.

5.3 Optimization process

For each site and month considered, four exposure have been tested: east, north, west and south. Multi-objective genetic algorithm optimizations have been performed on 20 individual populations, for 30 generation and a total of 600 simulation for each case considered.

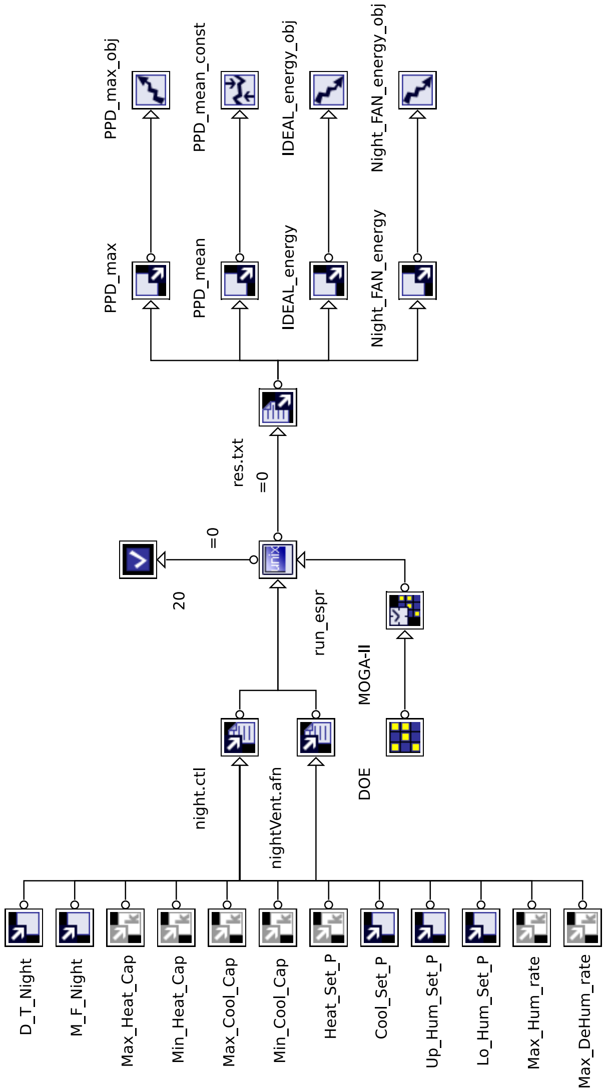


Figure 5.3 Optimization work flow

As in the rest of this thesis, modeFRONTIER [18] software has been chosen as optimization tool. A workflow of the optimization process is highlighted in figure 5.3.

The initial population has been chosen by means of Sobol algorithm on a design space defined as in table 5.2.

5.4 Results and analysis

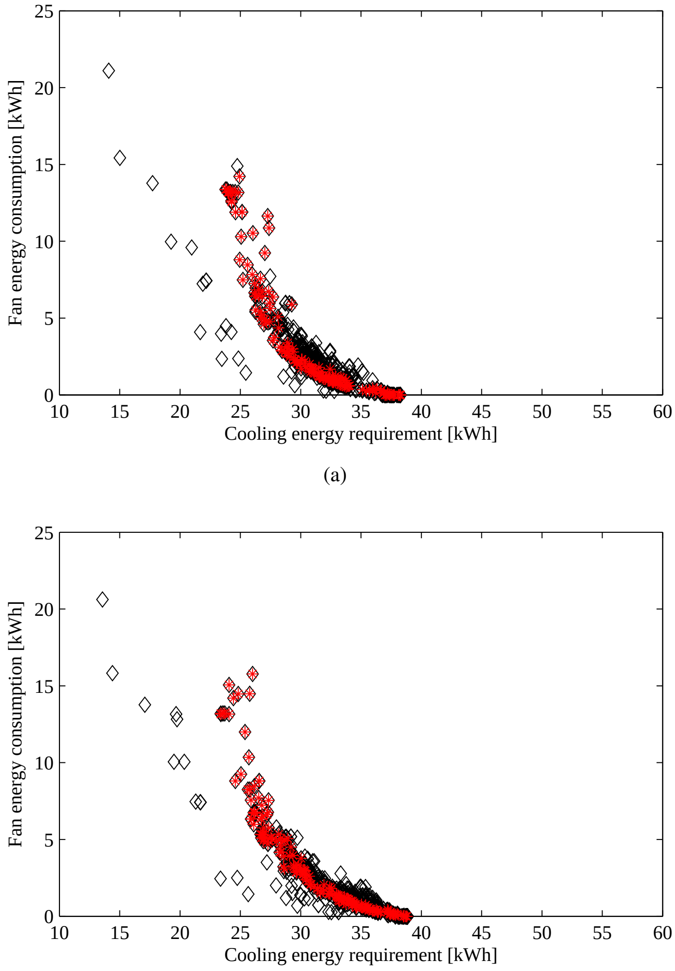
The results will be presented in the following by representing the solutions obtained and representing the solutions pertaining to the pareto front by red marks. Fig. 5.4 presents the relation between ideal cooling energy and fan energy requirement for site Rome and the four exposures in July, as expected the decrease of cooling energy is obtained with an increase in fan energy, nevertheless, independently from the exposition, the relationship between the two objectives is not linear, since a first consistent reduction of cooling energy requirement is obtained with limited night fan energy. Nevertheless to further reduce the energy required a decisive increase in fan consumption must be taken into account. The same results can be obtained for May, as reported in Fig. 5.5, but with a cooler climate the obtainable savings are more marked. To complete the plot in Fig. 5.6 the same results are presented considering the west exposure only, for Trieste, a northern Italian town with mild climate. In this case the performance of night ventilation is deteriorated, and savings in cooling energy requirement can be obtained only with an increase of fan energy. This behavior can be easily explained considering that Trieste is located on the seashore and the presence of the sea avoids large air temperature difference between night and day (see figure 5.2), so the cooling potential available for cooling the building fabric is low.

Volumetric flow rate during the night and the minimum temperature difference between internal and external air at which ventilation occurs are both parameters that affect night ventilation performance. In Fig. 5.7 the relation between volumetric flow rate and cooling energy for Rome and exposure west is presented. It is commonly accepted that an high volumetric flow during the night is beneficial, indeed the inspection of Fig. 5.7 reveals that the great part of the solutions fall between the values 0.15 and 0.3 m^3/s which represent air change rates of about 5 and 16 h^{-1} respectively, substantially lower than those utilized by other authors, for example in [51] intensive night ventilation were performed with air changes of 25 h^{-1}. Fan energy steadily increases with flow rate as reported in Fig. 5.8, hence the optimization process seeks optima with a reduced flow rate.

The temperature difference ΔT, affects both cooling and fan energy, the analysis of Figs. 5.9 and 5.10 shows that a maximum value of $\Delta t = 2$ K should be used.

5.5 Comments

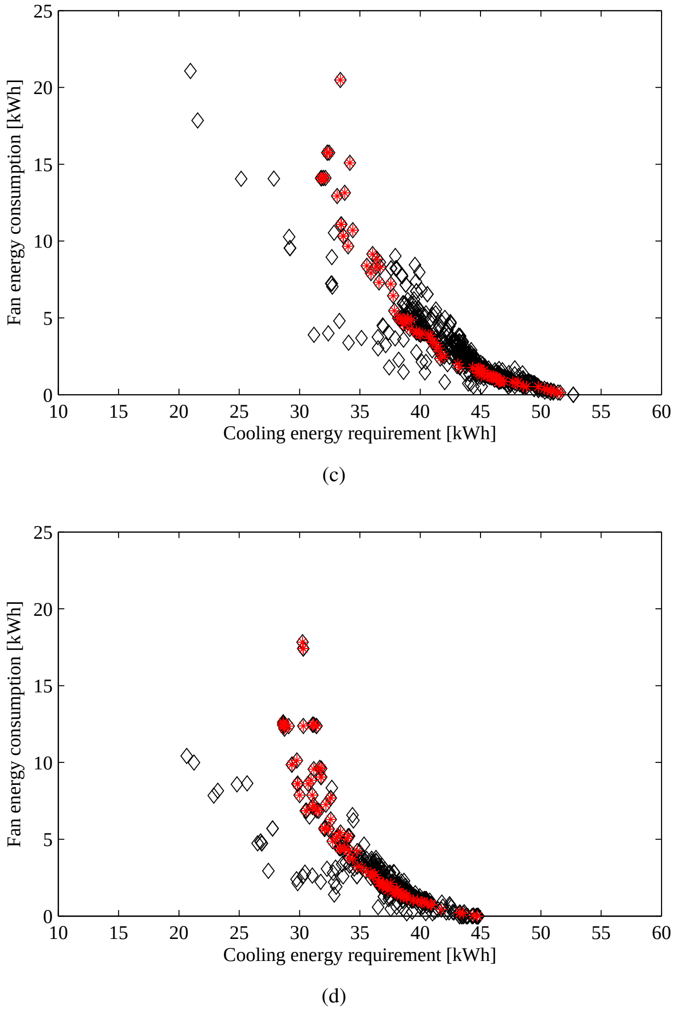
In this work a multi objective genetic optimization has been performed on a climatized part of a building with night ventilation. Different parameters have been considered with the aim of identifying optimum values to reduce the overall energy consumption. Energy saving has to be obtained while maintaining healthy indoor conditions, hence a mean PPD value has been set as a constraint, moreover maximum values of PPD have been minimized throughout the optimization process. The results show that night ventilation can be a viable strategy for reducing the overall energy requirements for building's cooling. The use of genetic optimization methods proved to be a helpful tool for analyzing problems where conflicting objectives have to be optimized simultaneously.



(a)

(b)

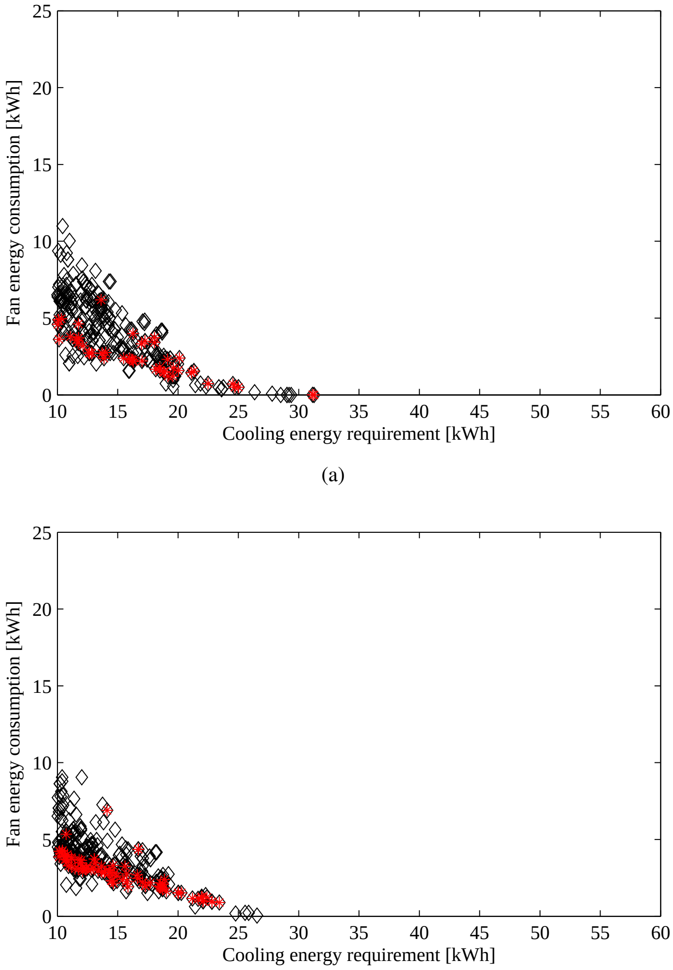
Figure 5.4 Cooling and fan energy requirement for Roma in July exposition: a) east; b) north; c) west; d) south



(c)

(d)

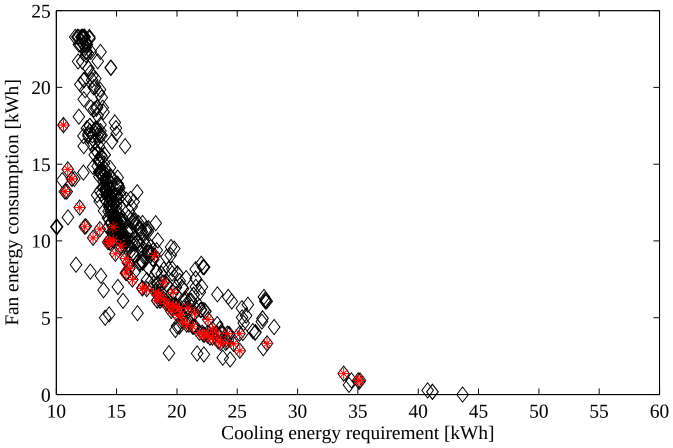
Figure 5.4 . . . Continued.



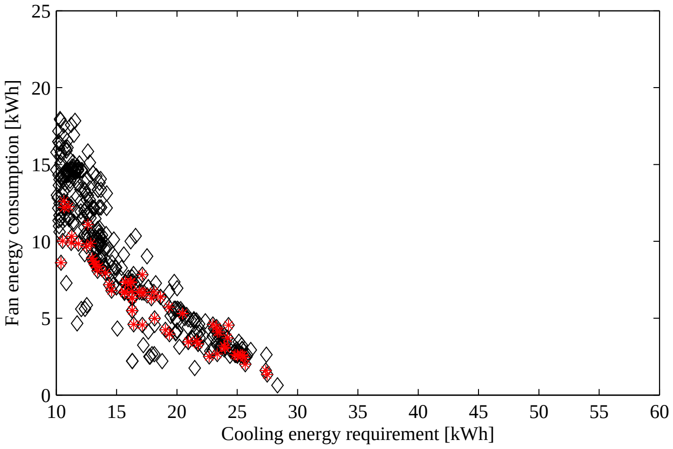
(a)

(b)

Figure 5.5 Cooling and fan energy requirement for Roma in may exposition: a) east; b) north; c) west; d) south

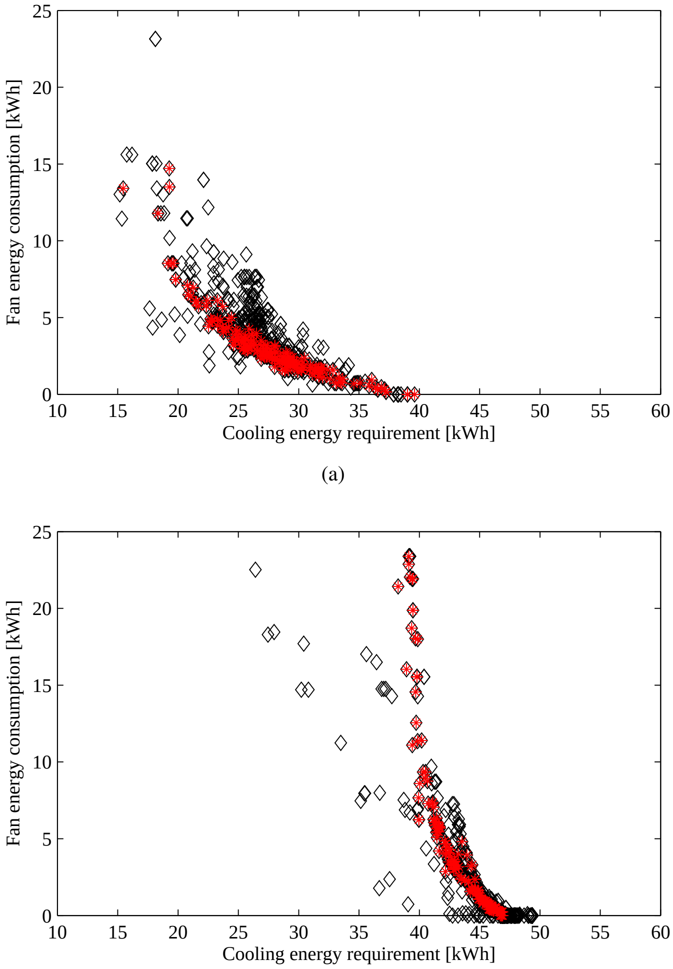


(c)



(d)

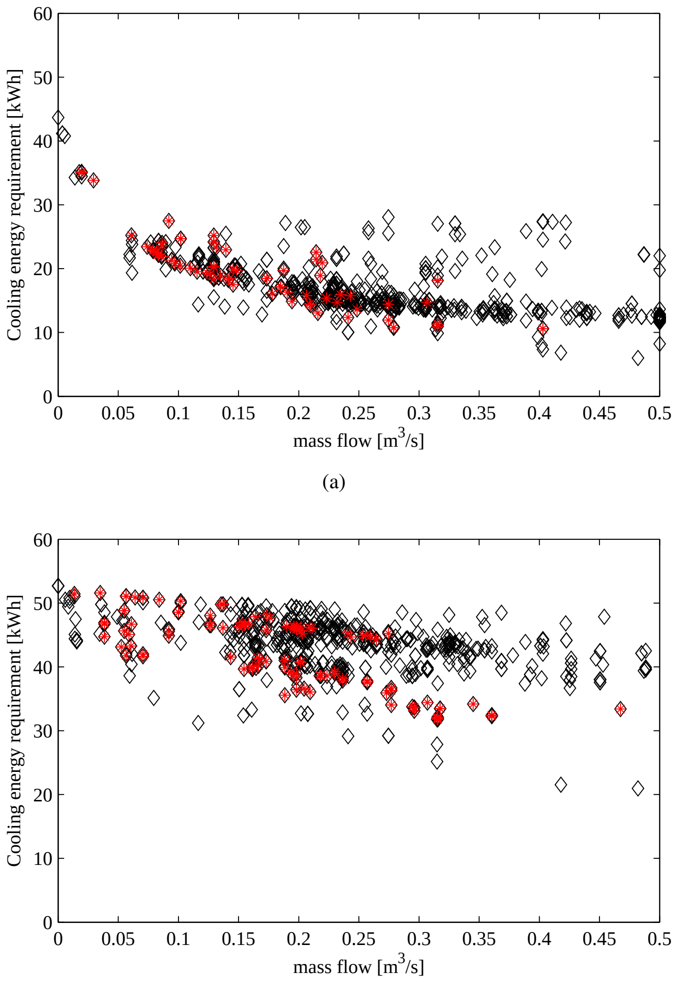
Figure 5.5 . . . Continued.



(a)

(b)

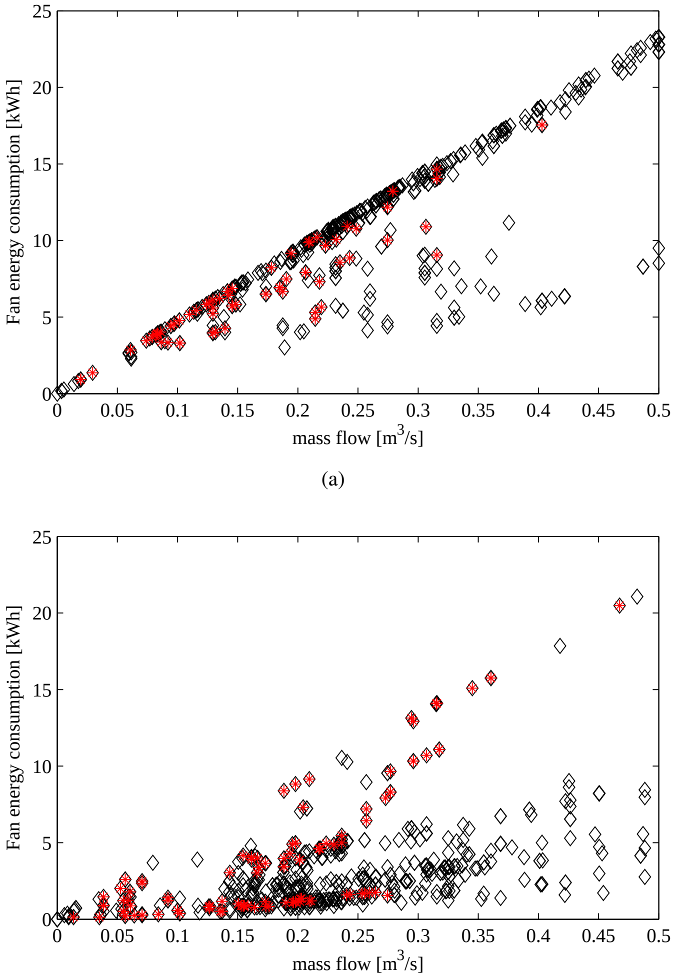
Figure 5.6 Cooling and fan energy requirement for Trieste exposition west: a) May; b) July



(a)

(b)

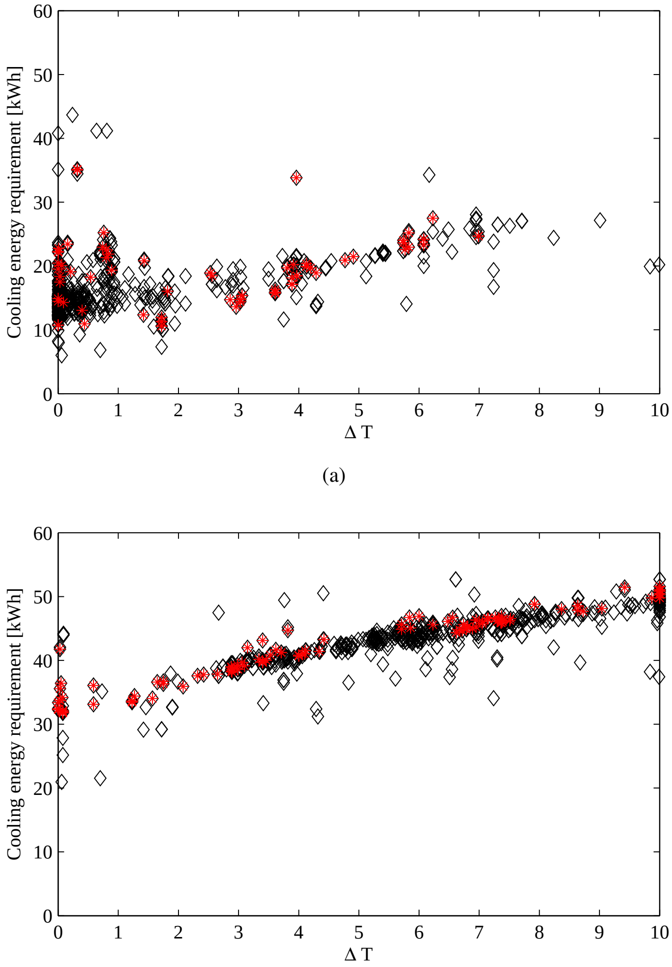
Figure 5.7 Volumetric flow rate and cooling energy consumption for Rome, exposition west: a) May; b) July



(a)

(b)

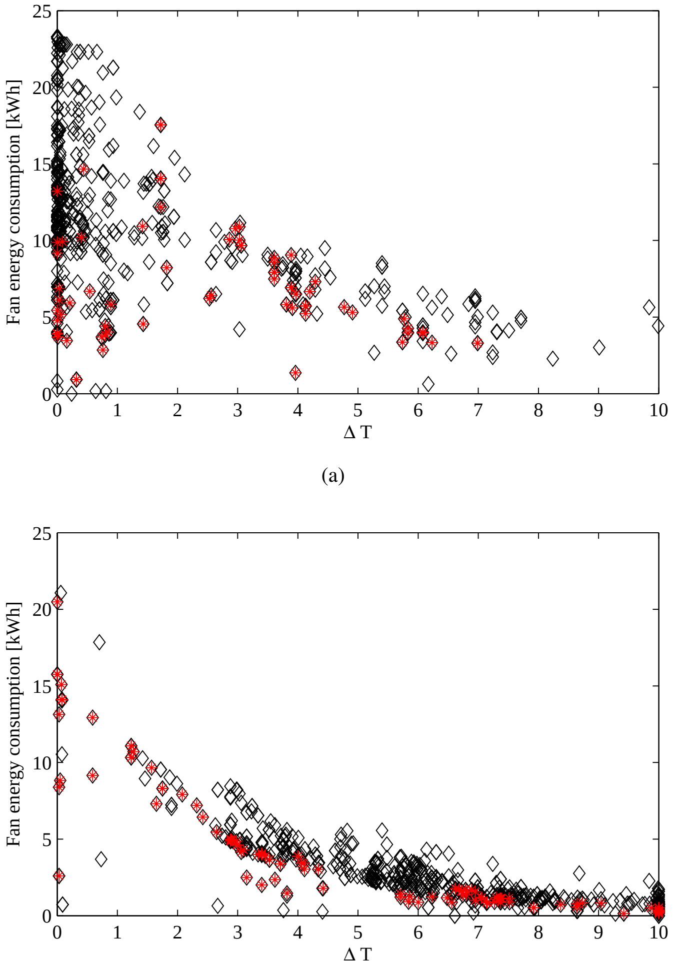
Figure 5.8 Volumetric flow rate and fan energy requirement for Rome, exposition west: a) May; b) July



(a)

(b)

Figure 5.9 Temperature difference set point and cooling energy requirement for Rome, exposition west: a) May; b) July



(a)

(b)

Figure 5.10 Temperature difference set point and Fan energy requirement for Rome, exposition west: a) May; b) July

Chapter Six

Breathing walls systems

6.1 Introduction

The development of truly sustainable, energy efficient buildings is key to tackling the threats of increasing energy consumption, diminishing fossil fuels and global warming. Today, a disproportionately high 50% of all primary energy is used in buildings, with 50% to 90% of this to maintain tolerable indoor temperatures - i.e., for space heating and cooling. With the total world consumption of marketed energy expected to increase by over 70% in the next two and a half decades [56], this continued level of energy demand to keep our buildings habitable is clearly not sustainable.

Dynamic Breathing Building (DBB) technology [57] can dramatically cut the energy demand of a building while at the same time improving indoor air quality and cleaning up the local environment. In a DBB, the external walls and roof of the building are constructed using air permeable dynamic insulation and all of the air needed to ventilate the building is drawn in through the insulation layer, which effectively becomes the air supply source, heat recovery device and (in the case of fibre-based media) filter of airborne particulate pollution.

Dynamic insulation effectively saves energy by exploiting contra-flux heat and mass transport through an air permeable medium in order to reduce the U-Value (or increase the R-Value) of the wall, with U being the overall heat transfer coefficient. This aspect of dynamic insulation has been investigated in [58, 59, 60, 61], who developed the basic theoretical understanding of the phenomenon. Independent experimental tests [62, 63] have been used to validate the basic heat transfer model, thus confirming the energy-saving potential of dynamic insulation. Systems analysis of the Optima House in Sweden, in which a dynamically insulated ceiling comprising 250 mm of cellulose insulation, 45 mm air gap and 13 mm plasterboard was used as ventilation source, suggests that a 31% reduction in energy consumption is achievable, even when air leakages through the envelope and wind induced pressure variations are taken into consideration [64].

It is possible, through careful design and material selection, to produce a multi-layer dynamic breathing wall structure where the air flow through the insulation layer is uniform (i.e., constant velocity) over the entire area of the wall [65]. Such a wall would not be very different from existing wall construction comprising an outer rain screen, external air cavity, dynamic insulation layer, internal air cavity and inner dry wall. Its primary distinguishing features are an external air vent to facilitate outdoor air being drawn into the external air cavity and an internal vent to deliver the pre-conditioned air to the room or HVAC plant.

Air layers in fully enclosed 'unventilated' cavities are assumed for design purposes to offer static thermal resistances in the range 0 - 0.23 m^{2}K/W, depending on layer thickness, orientation and direction of heat flow [66]. For a vertical wall with a 10 mm unventilated cavity thermal resistance is assumed to be 0.15 m^{2}K/W, rising to a maximum of 0.18 m^{2}K/W for cavity thicknessed of 25 mm and higher.

The present work seeks to achieve a fundamental understanding of heat transfer across the ventilated cavities of a dynamic insulation layer when air flows through the wall. This is the first investigation of its type undertaken to evaluate the thermal effects of free and/or forced thermal convection in ventilated air cavities forming part of a DBB envelope. The results include tentative estimates and suggest the inner and outer thermal resistance contributions of a ventilated cavity in multi-layer dynamically insulated wall construction vary dynamically and are significantly higher than for unventilated cavities.

The incoming ventilation air is thus effectively pre-heated in winter (pre-cooled in summer) in 3 stages as it flows through the outer cavity, dynamic insulation layer and inner cavity before entering the building. This should translate to improved heat recovery through the wall, counter-balanced by greater room-to-wall interaction with inner wall surfaces that are slightly cooler in winter and warmer in summer than the existing norm.

6.2 Principle of dynamic breathing walls

DBB is a technology aimed at reducing the conductive heat loss throughout a wall construction. Its basic concept is to draw air from outside through a permeable insulating medium, thus reducing the conductive flux. The theory is well developed in the steady, one-dimensional case. A transport term due to the air "breathing" through the permeable medium is added to the equation of conduction:

$$\rho_c c_c \frac{\partial T}{\partial t} - \kappa_c \nabla^2 T + \rho_a c_a \mathbf{u} \cdot \nabla T - \dot{q} = 0 \tag{6.1}$$

that in the steady, one-dimensional case in absence of internal heat generation is

$$\frac{\mathrm{d}^2 T}{\mathrm{d}x^2} - \frac{\rho_a c_a u}{\kappa_c} \frac{\mathrm{d}T}{\mathrm{d}x} = 0 \tag{6.2}$$

Of this equation there exists an analytical solution, that for Dirichlet boundary conditions yields the following expression:

$$\frac{T(x) - T_0}{T_L - T_0} = \frac{e^{Ax} - 1}{e^{AL} - 1} \tag{6.3}$$

where $A = \rho_a c_a u / \kappa_c$. The plot in figure 6.1(a) reveals a non constant heat flux through the wall, the system acting as a counter flow heat exchanger. The effect of DBB is to reduce the conductive heat loss at the external surface of the wall (position $x = 0$ in figure 6.1(a)), while warming up the incoming air, which receives heat from the room ambient. The heat flux at the external wall is given by:

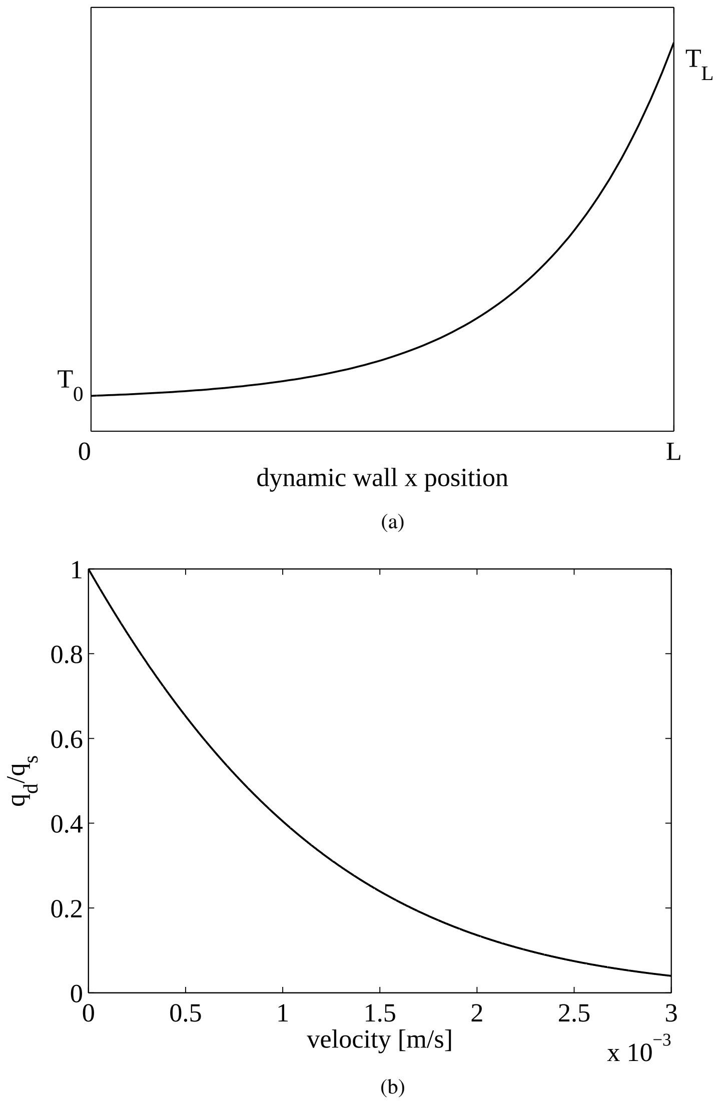
$$q_d = \frac{AL}{e^{AL} - 1} \frac{\kappa}{L} (T_L - T_0) = \frac{AL}{e^{AL} - 1} q_s \tag{6.4}$$

where q_s is the static - with no mass flow - heat flux at the same boundary conditions. The value of q_d/q_s as a function of the velocity follows the trend in figure 6.1(b) for a typical construction. Hence, even at quite low velocities it is theoretically possible to reach a quasi-zero U-value. In fact DBB constructions, as sketched in figure 6.2, are made of a series of layers: basically outer rain screen, external air cavity, insulation layer, internal air cavity, inner dry wall. In such a configuration the air flow path has a two-dimensional feature, whose influence on the heat transfer has not been invertigated yet, in the authors' knowledge. The aim of the present work is thus to enquire into the two dimensional heat transfer in the air cavities, at a first stage on a simplified model, concerning the sole cavity.

6.3 Problem statement

Numerical simulation of heat and mass transfer through a ventilated air layer in the wall elements of a DBB envelope assumes a small inlet opening at the base of the outer wall, constant air flow velocity through the air permeable dynamic insulation inner wall, and variable temperature difference between outer and inner walls. Comsol Multiphysics©, a general purpose finite element (FE) software package was used to perform parametric simulations of performance as a function of cavity depth, temperature difference and air flow velocity.

The basic geometry is of a thin 2.4 m high rectangular cavity. Three different thicknesses of 22.5, 25.0 and 27.5 mm were considered. Approximate analysis for typical wall construction was used to estimate the maximum temperature difference between the outer and inner wall surfaces. The wall taken into account is composed of an external block wall, two air gaps, the dynamic element and an internal dry wall, as sketched in figure 6.2. Assuming typical thermal resistance values as those in table 6.1 the maximum difference in temperature was found to be around 3 K when the indoor and outdoor temperature difference exceeded 40 K.



(a)

(b)

Figure 6.1 steady one-dimensional analytical solution: (a) temperature profile; (b) q_d/q_s vs velocity

Table 6.1 typical thermal resistances in a dynamic wall construction. Values in m^2K/W

external surface	0.04
block wall	0.17
air gap	0.18
dynamic element	2.00
dry wall	0.18
internal surface	0.13

For this reason the simulations have been conduced for a range of temperature differences not beyond this threshold. The 1, 2 and 3 K cases have been considered and compared against the 0 K case referred to in the text in which the effects of natural convection are neglected.

The air properties are taken at 0 ° C for the pressure of 101.325 kPa (table 6.2). This temperature is the outside reference in UK. Moreover, as the temperature of the air increases, the buoyancy effect reduces. Thus the choice of this reference gives a slight overesteem of natural convection effects. Solar gain on the outer wall surface was disregarded (although it is accepted that this could be a significant factor in sunny climates). Air flow velocities of 0.001, 0.002 and 0.003 m/s were considered, with the latter typically corresponding to the point where all of the conduction heat is taken up by the incoming air and brought back into the building - i.e., the point at which the wall achieves a U-value of 0.

For thermal resistance calculations in building components the British Standard [66] fixes values of thermal resistances of unventilated air layers in between 0.11 - 0.18 m^2K/W. This interval might be halved in case of slightly ventilated cavities. These values apply for layers bounded by surfaces with emissivity higher than 0.8. When this condition is not achieved, an annex draws the guideline to calculate the resistance as a combination of a conductive/convective coefficient and a radiative coefficient.

At the time being the dynamic elements are usually made by a polyester fibre filter

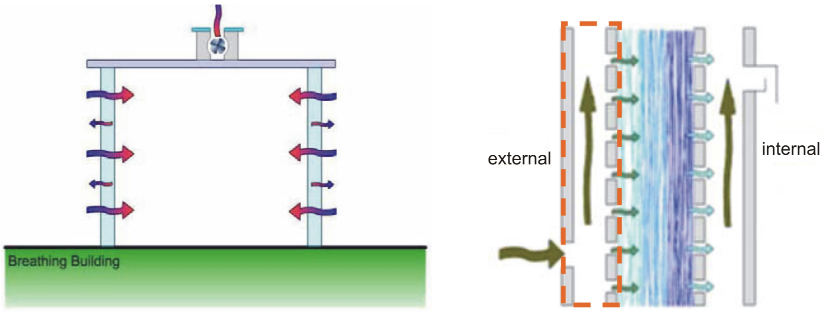


Figure 6.2 Dynamically insulated wall pattern

Table 6.2 Air termophysical properties

pressure reference: 1 atm					
ref T	ρ	μ	Cp	κ	$g\beta/\nu^2$
$[^\circ C]$	$[kg/m^3]$	$[Pa \cdot s]$	$[kJ/(kg \cdot K)]$	$[W/(m \cdot K)]$	$[1/(K \cdot m^3)]$
0	1.282	17.32E-06	1.004	24.27E-03	19.53E+07

medium and an outer casting of expanded polystyrene. The assessment of radiative properties for a material is not a trivial task. Much more in this case were there is a high light penetration.

Moreover, in this piece of work we focus on the convective heat transfer due to the imposed forced convection, enquiring into the two dimensional behaviour of the phenomenon, and the possible contributions from natural convection effects. Thus,radiant heat exchange has not been taken into account in this investigation.

6.4 Numerical considerations and boundary conditions

Using the Boussinesq approximation and assuming constant thermal and physical properties for air, the density dependant buoyancy term varies linearly as a function of temperature. The equations of mass, momentum and energy conservation write as follows:

$$\nabla \cdot \mathbf{u} = 0 \tag{6.5}$$

$$\rho_a(\mathbf{u} \cdot \nabla \mathbf{u}) = \mu \nabla^2 \mathbf{u} - \nabla p + \mathbf{g}\beta \Delta T \tag{6.6}$$

$$\rho_a c_a \mathbf{u} \cdot \nabla \theta = \kappa_a \nabla^2 \theta \tag{6.7}$$

The set of equation is solved in a coupled way as a stationary non linear problem, with the use of the *umfpack* routine to solve the linearized system each step. The flow is assumed laminar. After a series of mesh dependence tests, a structured mesh has been chosen, consisting of 11500 quadrilateral elements with second order shape functions for velocity components and temperature, and first order functions for the pressure.

The fluid dynamic boundary conditions are of fixed outward velocity on the dynamic wall, while non-slip conditions are used elsewhere, apart at the inlet edge, where the incoming mass flow must equal the outgoing.

The actual thermal boundary conditions are unknown and depend on the transfer phenomenon in the whole wall and on the external and internal ambient conditions. In order to highlight the bi-dimensional behaviour of the heat transfer and get a wider set of conclusions, three different boundary conditions will be applied. At first, a constant wall temperature will be imposed on the vertical surfaces; secondarily, a constant wall temperature will be set on the external wall, while an integral constraint is set on the porous wall as to let the temperature profile set by itself, but with a mean value corresponding to the case of constant Temperature. In the last case, both walls will be applied with an integral constraint. Both summer and winter conditions will be discussed as the temperature gradient is reversed in the two cases, and the natural convection contribution is anti-symmetric.

6.5 Results and discussions

In tables 6.3-6.8 are summarised the results of the numerical simulations in terms of ratio between mean dynamic and static conductive heat flux.

The effect of natural convection on the global heat transfer is of least relevance in contrast to the forced convection in varying the thermic resistance of the enclosure. The cold versus hot climate conditions, as the temperature difference rise, have the effect to slightly increase or, respectively, decrease the performances.

The velocity, instead, is of most relevance in modifying the heat transfer value. Neglecting the natural convection contribution, the global heat transfer seems to not be dependent on the type of boundary conditions imposed, as one can see having a look at the first row of each table. Meanwhile, increasing the temperature difference, the third kind of boundary condition seems to affect much more than the others the q_d/q_s value.

In figure 6.3(a), 6.3(b) and 6.3(c), the local values of q_d/q_s at the external non porous wall are graphed, for a thickness of 22.5 mm and neglecting natural convection effects. the third boundary condition type shows a constant flux behaviour through the height of the cavity, while the first two cases have a growing behaviour as the distance from the inlet increase. this is due to the heat exchanger nature of the enclosure. As the air is close to the inlet, its temperature is far from a steady condition, so it start to gain (in winter) or release (in summer) heat. this two-dimensional behaviour is much more highlighted in figures 6.4(a), 6.4(b) and 6.4(c) which show the sum of conductive and convective heat flux on the porous wall. The mean value of this function equals the conductive flux on the inlet wall, so it is licit to use the same terminology in indicating as q_d/q_s this summation. In this case, the air entering the inlet at a quite high velocity increases the heat transfer at the bottom of the porous wall, while decreasing it as turning away from the inlet. In some cases, and for higher velocities this fact is amplified, the q_d/q_s reaches a counter-flux behaviour. This means that the conductive flux coming from the porous wall is lower than the convective flux through it. In its way from the inlet through the height of the layer air has gained such heat as to make this happen, proving the highly two-dimensional behaviour of the considered system.

The plots in figure 6.5 represent the temperature profile for different heights of the enclosure for the 25 mm thickness case, in absebce of buoyancy and assuming a mean defference temperature of 3° C between the walls. The profile is similar to the one-

Table 6.3 winter conditions, constant wall temperature

Temp	Velocity								
	0.001			0.002			0.003		
	Gap thickness								
	22.5	25.0	27.5	22.5	25.0	27.5	22.5	25.0	27.5
0	0.661	0.631	0.603	0.439	0.401	0.366	0.294	0.258	0.227
1	0.672	0.649	0.630	0.449	0.417	0.390	0.302	0.271	0.246
2	0.686	0.670	0.660	0.462	0.436	0.418	0.314	0.288	0.270
3	0.700	0.690	0.689	0.475	0.455	0.445	0.326	0.305	0.293

Table 6.4 winter conditions, free temperature on dynamic wall

Temp	Velocity								
	0.001			0.002			0.003		
	Gap thickness								
	22.5	25.0	27.5	22.5	25.0	27.5	22.5	25.0	27.5
0	0.657	0.627	0.599	0.433	0.395	0.361	0.289	0.254	0.223
1	0.679	0.662	0.652	0.455	0.430	0.412	0.308	0.283	0.264
2	0.704	0.701	0.709	0.481	0.468	0.464	0.331	0.315	0.307
3	0.730	0.740	0.764	0.507	0.504	0.513	0.354	0.346	0.347

dimensinal analytical solution, proving there exists a dynamic effect in the cavities, as well. The two dimensinal effect through the height of the cavity is of evidence even in these plots.

6.6 Comments

This piece of work is a first attempt in understanding heat transfer in ventilated cavities forming part of DBB systems. The aim of the study was to investigate the characteristic of the phenomenon in an environment presenting a pronounced two-dimensional pattern, while in literature most of the authors focus on the definition of monodimen-

Table 6.5 winter conditions, free temperature on both walls

Temp	Velocity								
	0.001			0.002			0.003		
	Gap thickness								
	22.5	25.0	27.5	22.5	25.0	27.5	22.5	25.0	27.5
0	0.655	0.626	0.599	0.431	0.394	0.362	0.287	0.254	0.225
1	0.700	0.689	0.683	0.465	0.440	0.422	0.312	0.286	0.266
2	0.736	0.738	0.748	0.493	0.479	0.471	0.334	0.314	0.301
3	0.769	0.783	0.807	0.519	0.513	0.515	0.354	0.340	0.334

Table 6.6 summer conditions, constant wall temperature

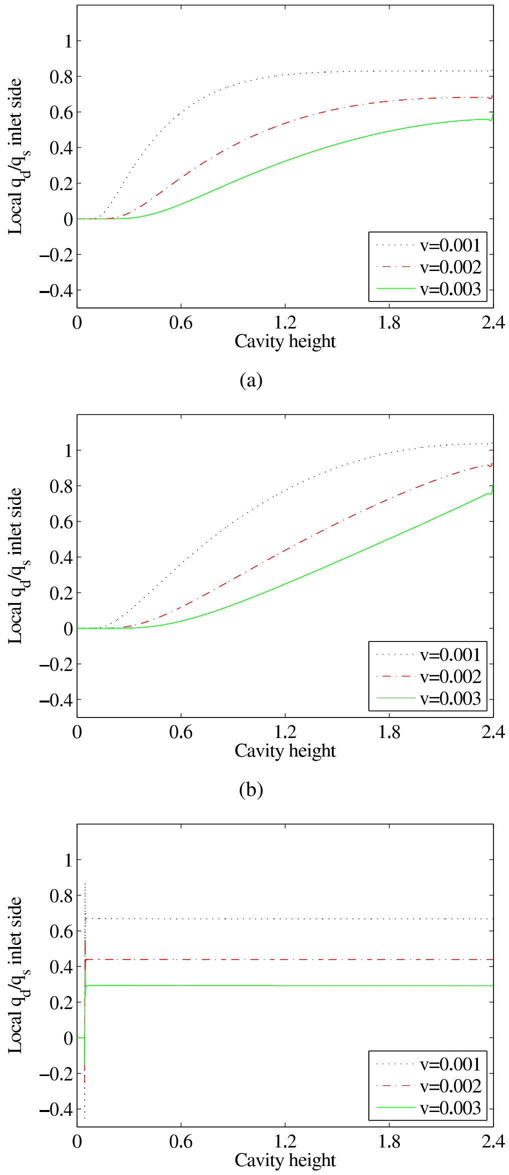
Temp	Velocity								
	0.001			0.002			0.003		
	Gap thickness								
	22.5	25.0	27.5	22.5	25.0	27.5	22.5	25.0	27.5
0	0.661	0.631	0.603	0.439	0.401	0.366	0.294	0.258	0.227
1	0.658	0.627	0.598	0.436	0.398	0.362	0.292	0.256	0.224
2	0.657	0.625	0.594	0.435	0.396	0.359	0.291	0.254	0.222
3	0.655	0.623	0.591	0.434	0.393	0.356	0.290	0.252	0.219

Table 6.7 summer conditions, free temperature on dynamic wall

Temp	Velocity								
	0.001			0.002			0.003		
	Gap thickness								
	22.5	25.0	27.5	22.5	25.0	27.5	22.5	25.0	27.5
0	0.657	0.627	0.599	0.433	0.395	0.361	0.289	0.254	0.223
1	0.655	0.624	0.594	0.431	0.392	0.356	0.287	0.251	0.220
2	0.654	0.622	0.590	0.429	0.389	0.353	0.285	0.249	0.217
3	0.652	0.619	0.585	0.427	0.386	0.348	0.284	0.246	0.214

Table 6.8 summer conditions, free temperature on both walls

Temp	Velocity								
	0.001			0.002			0.003		
	Gap thickness								
	22.5	25.0	27.5	22.5	25.0	27.5	22.5	25.0	27.5
0	0.655	0.626	0.599	0.431	0.394	0.362	0.287	0.254	0.225
1	0.635	0.601	0.568	0.413	0.373	0.336	0.275	0.239	0.208
2	0.626	0.590	0.556	0.404	0.362	0.324	0.267	0.231	0.199
3	0.620	0.584	0.551	0.397	0.354	0.316	0.262	0.225	0.193

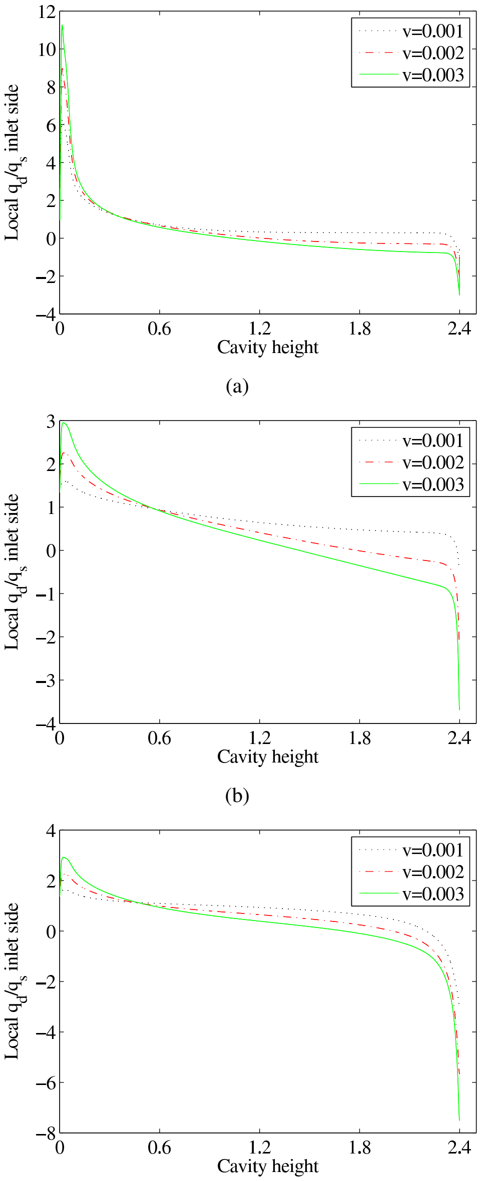


(a)

(b)

(c)

Figure 6.3 External wall q$_d$/q$_s$: (a) first boundary condition; (b) second boundary condition; (c) third boundary condition.

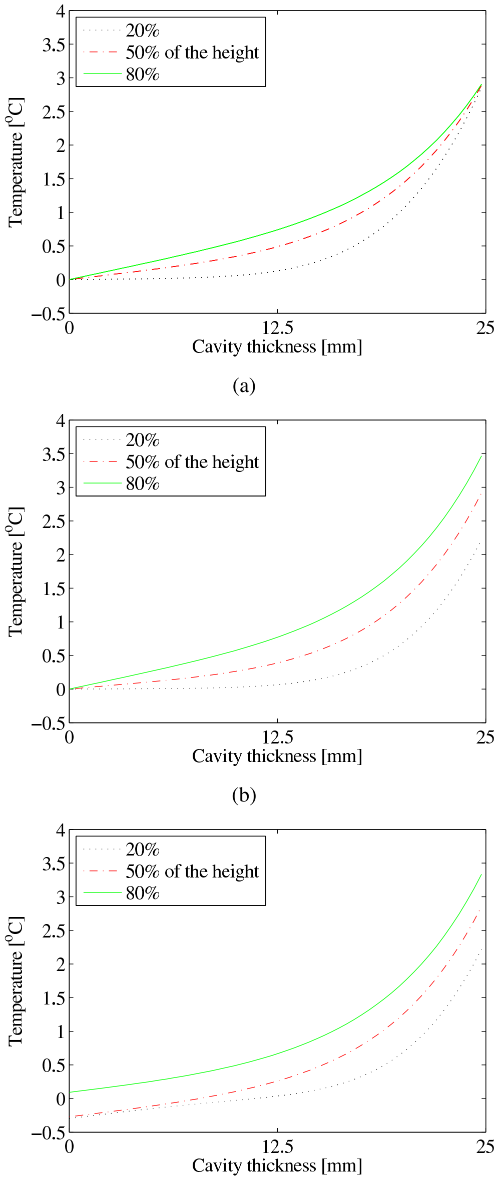


(a)

(b)

(c)

Figure 6.4 Internal porous wall q_d/q_s: (a) first boundary condition; (b) second boundary condition; (c) third boundary condition.



(a)

(b)

(c)

Figure 6.5 Temperature profiles at different height of the cavity for a 0.003 m/s velocity, neglecting buoyancy : (a) first boundary condition; (b) second boundary condition; (c) third boundary condition.

sional relations. The pioneering nature of the study has led to the employment of modelling simplifications such as neglecting radiative effects. The results show that in such narrow cavities the buoyancy effects are of minimal importance, while the forced convection influence the conductive heat transfer throughout the layer, augmenting its global thermal resistance; the layer itself behaving as a breathing element. Yet the heat transfer in the air gap reveals highly two dimensional, with possible influences on the vertical temperature and heat flux distributions in the whole DBB. This two dimensional effect might suggest a geometrical redraw of the components in order, for example, to spare materials where the heat flux is lower. Or alternatively, it might be exploited in the design of auxiliary plant components, such as the positioning of a post heating battery for a HVAC system close to the inlet of the air gaps, where the heat flux shows quite high.

Conclusions

The main objective of this thesis has been the development, application and testing of tailored evolutionary techniques to energy transfer efficiency. These have been applied to both heat transfer problems and low consumption buildings.

Problems faced in real situations are characterized by several and conflicting objectives, which lead to multi-objective optimization problems, where not a single optimum solution exists. Moreover, the relations occurring between objectives and decisional parameters are unlikely to be simple analytical expression. They are rather usually unknown functions, that might lack of continuity, derivativeness, connectedness. These facts have raised in recent times an increasing interest on evolutionary algorithms for optimization. They are heuristic methods that use some mechanisms inspired by biological evolution: reproduction, mutation, recombination, natural selection and survival of the fittest.

In Chapter 1 an overview on optimization methods has been given, which stresses the peculiarities of evolutionary optimization techniques and common practice in evolutionary optimum search. Evolutionary Algorithms (EA) are usually robust but neither accurate nor fast. Yet, their most important attribute and reason for a wide use is the applicability to almost any single- or multi-objective optimization problem of whichever complexity. Their weakest aspect lies on the theoretical side. Being EA heuristic processes, most of the current reseach aims at proving actual convergence. Nevertheless, the wide literature on successful applications of evolutionary optimizations sets EA as an extremely promising field.

The fist part of the thesis has been directed to the study and optimization of heat transfer problems. The problems considered deal with geometry shapes. The objectives of the studies are functions of their physical domain, whose change in form affects the behaviour of the system. In this sense, great attention is to be paid to the method by which shapes are mathematically represented. The choice of a good parametrization is not a trivial task. Depending on the (usually unknown) optimal shape, the model has to be complete enough as to match the desired target. Yet if it is overdeveloped this may lead to slow or unstable optimization processes. In chapter 2 an overview on geometrical representations has been outlined, with particular attention to Bézier and NURBS curves and surfaces, that have been used in the following chapters to draw the computational domains, and represent the standard for form description and manipulation in industrial 3D CAD (solid modelling) systems.

In chapter 3 it has been presented an approach for the multi-objective shape optimization of two-dimensional convective periodic channels, which represent the fundamental building block of many heat exchangers and other heat transfer devices. The numerical simulation has been obtained by means of COMSOL, an unstructured Finite Element solver, for a fluid of $Pr = 0.7$, representative of air and other gases, assuming fully developed velocity and temperature fields, and steady laminar conditions. The shape of the channels have been described by either NURBS, with their control points representing design variables, or by simpler piecewise-linear profiles. Given the multi-objective nature of the optimization problem, i.e. maximization of the heat transfer and minimization of the friction factor, a *multi-objective genetic algorithm* has been used, together with the *Pareto dominance* concept. The results obtained have revealed that the type of geometrical parametrization is of paramount importance, since it affects both performance of the design and computational costs. In particular, it has been found that simpler linear-piecewise channels, although easier to optimize, do not provide the same performance obtained by channels described by NURBS. In addition, for the latter type of geometrical description, it has been shown that very different channel shape offer almost the same flow and heat transfer performance, i.e. non-uniqueness of the shape optimization problem. This non univocity in the solution space has been already noted by other authors, and it is a peculiarity that can rarely be achieved with classical optimization processes. The robustness of genetic algorithms, which mimic the evolution of living organisms in nature, evolving an initial population towards the best possible fitness, is capable of preserving multiple good solutions. The 2D results have been extended, though in an exploratory fashion, to the 3D case, showing that the optimization leads to 3D geometries characterized by the presence of secondary motions. These, in turn, highly enhance the heat transfer rate, without affecting the pressure losses. In this sense, it is also reassuring that the results obtained by the optimization agree well with the classical methods used for heat transfer augmentation.

In chapter 4 Inverse Heat Transfer Problems (IHTP) have been considered. These are ill-posed problems that admit a solution if, and only if, the geometrical domain can be appropriately modified. IHTP can be considered subsets of shape optimization problems, where a direct design of shape is sought by applying overspecified boundary conditions. In this work a genetic algorithm has been used to reproduce the two-dimensional direct design of shape considered by other authors, but where a gradient-based method had been applied. A heated substrate is embedded in a solid body and a determined constant surface temperature is sought. The optimization is single-objective and the assessment parameter to be minimized has been chosen the divergence from the target surface temperature, evaluated with a quadratic norm. Bézier curves and surfaces have been used as geometrical modelling tool. The first part of the work has concerned the assessment of three different parametrizations. A purely conductive problem has been solved to test geometrical models composed of two, three, and four curves respectively. With the addition of some adjustment to its original formulation, the four curves parametrization has proved to be best in reaching the target curves. Hence, it has been chosen to extend the optimization to the conjugate case. The solid body has been posed in relative motion respect to a cooling fluid. Forced convection assumption has been assumed in laminar, steady state conditions. Different flow regimes and thermal conductivity ratios between solid and fluid have been studied. In some cases a loss of sensibility towards the objective function has been detected, slowing down the optimization convergence. In order to tackle this problem the multi-objective feature of genetic algorithms has been exploited in order to speed the convergence rate. Multi objective problems usually deal with conflicting goals, leading to a multiplicity of solutions. In this case, a second objective has been introduced that agrees with the first, but evaluates the divergence from the goal temperature with an infinity norm. This strategy has shown successful, proving once again the capabilities of evolutionary techniques. For this study has a precursory fashion, it has been tried to reduce as much as possible the number of degrees of freedom of the geometric model. Moreover, as the computational cost associated to three dimensional CFD computations is still prohibitive in COMSOL package, a segregated approach to solve Navier-stokes equations has been implemented to carry out the otherwise intractable three-dimensional optimization. The aim of the work has been to test the possibility to reach coherent results with an optimization procedure that has shown reliable and robust in the other examples proposed in this thesis. Good solutions have been reached, but with a large CPU time amount. In order to dim the CPU time effort the use of metamodels has been tested, but producing discouraging results. A probable reason of the failure of responce surface methods lies on the fact that the design variables of the optimization are Bézier points, each of them influencing the surface shape for a wide tract, and consequently affecting the fluid-dynamic and thermal behaviour in a complex way. As a final consideration on the problems considered, the approach has proved robust and reliable, and genetic algorithms have shown to be a very general and flexible tool, able to reach convergence even with a big loss of data. However, direct design of shape is not a push-button, automatic process, since the geometrical parametrization is a very delicate activity.

In the second part of the thesis, problems related to energy savings in buildings have been considered.

In chapter 5 a multi-objective genetic optimization has been performed on a climatized part of a building with night ventilation. Night ventilation is a passive cooling technique that can significantly reduce the cooling loads and energy requirements, but a trade off must be found between energy cost savings and zone thermal comfort. The effect of night ventilation is to lower the fabric temperature during nighttime. In this way the daily cooling loads are reduced. The main effect of this technique is a smeared distribution of energy damand, which reduce daily power requirement peaks due to air conditioning systems. With the help of optimization strategies, a global energy consumption reduction can be achieved as well. On the other hand, night ventilation has a drawback: a reduced internal temperature in the morning hours might affect human comfort, augmenting the *percentage of persons dissatisfied* (PPD). In this work the room temperature and humidity set points together with the night ventilation system characteristics have been chosen as design variables for an optimization process. The reduction of both diurnal and nocturnal energy consumption have been the objectives of the study. In addition, the minimization of the PPD indicator has been considered. Two different sites have been investigate: Rome and Trieste. The simulations have been performed for two characteristic days, one in July to represent conditions with high thermal loads, and one in May characterized by lower ambient temperature and solar irradiation. The results show that night ventilation can be a viable strategy for reducing the overall energy requirements for building's cooling. On the other hand, it has been noted that for quite mild climates as Trieste's one is, where the temerature difference between diurnal and nocturnal hours is not substantial, the effect of night ventilation is very poor.

Finally, in chapter 6 a numerical study on dynamic insulation systems has been conducted. This work is a first attempt in understanding heat transfer in ventilated cavities forming part of *dynamic breathing building* (DBB) systems. The aim of the study was to investigate the characteristic of the phenomenon in an environment presenting a pronounced two-dimensional pattern, while in literature most of the authors focus on the definition of one-dimensional assumptions. The pioneering nature of the study has led to the employment of modelling simplifications such as neglecting radiative effects. The results show that in such narrow cavities buoyancy effects are of minimal importance, while forced convection influences the heat transfer in the cavity, augmenting its global thermal resistance, the cavity itself behaving as a breathing element. Yet the heat transfer in the air gap reveals highly two dimensional, with a vertically varying heat flux or temperature distribution, depending on the imposed boundary conditions. This two dimensional effect might influence the temperature stratification in living spaces, suggesting further assessment of comfort conditions.

As a final remark, in this thesis evolutionary optimization has been applied to different kind of problems, proving to be a very robust and helpful tool.

The application to efficient buildings design is a newborn area of interest and, coupled to a software for energy simulation in buildings, it has proved capable of providing plenty of assessment information in a natural and easy way.

Although their computational efficiency is not comparable to other optimization techniques, the use of evolutionary algorithms within shape optimization contest has shown the possibility to perform truly multi-objective searches in absence of functional constraints. In particular, the results obtained in the optimization of wavy channels, where multiple solution have beem found is a peculiarity rarely achieved with classical optimization processes. The attempt to use metamodels in order to reduce the computational effort has been unsuccesfull in the presented case. However, other authors' experiences suggest further trials should be undertaken, as the methodology has shown quite encouraging.

Ringraziamenti

Mi permetto di inserirea a chiosa di questa tesi qualche riga di ringraziamento.
In primis, ringrazio il prof. Nobile, che mi ha dato la possibilità di intraprendere quest'esperienza e mi ha guidato durante il percorso di Dottorato. Al prof. Manzan devo tutto quello che so su linux e gratitudine infinita per la collaborazione sui temi riguardanti il risparmio energetico. Una menzione va a tutto il gruppo di Fisica Tecnica con cui ho amabilmente lavorato negli ultimi tre anni. Ringrazio il prof. Imbabi per avermi concesso di trascorrere presso l'Università di Aberdeen un bellissimo periodo e di avermi introdotto alla tecnologia dei muri traspiranti. Ringrazio i miei genitori e tutta la mia famiglia, che mi ha sempre sostenuto in ogni scelta da me fatta, e a cui dedico, per quel che vale, questo lavoro. Un ringraziamento particolare a Laura che mi ha sopportato negli ultimi mesi. Un ulteriore ringraziamento particolare ad Andrea, che mi sopporta da anni, ma viene ripagato con la stessa moneta.

Grazie anche a tutti voi, e a coloro che ho dimenticato:
Aida, Alberto, Andrea C., Andrea S., Angelo, Claudia, Claudio, Egidio, Elisa, Enrico, Ezio, Fabrizio, Fausta, Federico, Franceca, Franz D.P., Gigi, Gino G., Gino R., Giulia, Giuseppe, Jim, Laura, Luca, Luigi, Marco, Marko, Martina, Martino, Mauro, Michela, Mohammed, Monica, Nicola, Nina, Paolo B., Paolo G., Ravi, Silvia, Silvia P., Stefano C., Stefano P., Valentina, Valentino, Walter.

Bibliography

[1] Singiresu Rao. *Engineering Optimization, Theory and Practice*. John Wiley & Sons, Inc., New York, 1996.

[2] D. E. Kirk. *Optimal control theory: an introduction*. Prentice-Hall, Englewood Cliffs, NY, 1970.

[3] L. Padovan, V. Pediroda, and C. Poloni. *Multidisciplinary Methods for Analysis Optimization and Control of Complex Systems*, volume 6 of *Mathematics in Industry*, chapter Multi Objective Robust Design Optimization of Airfoils in Transonic Field, pages 283–295. Springer Berlin Heidelberg, 2005.

[4] P. A. Jensen and J. F. Bard. *Operations Research Models and Methods*. Wiley and Sons, 2003.

[5] Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, 2002.

[6] C. Coello Coello. *Recent Trends in Evolutionary Multiobjective OptimizationEvolutionary Multiobjective Optimization: Theoretical Advances And Applications*, chapter Recent Trends in Evolutionary Multiobjective Optimization, pages 7–32. Springer-Verlag, London, 2005.

[7] S. Poles, Yan Fu, and E. Rigoni. The effect of initial population sampling on the convergence of multi-objective genetic algorithms. In *7th international conference on multi-objective programming and goal programming (MOPGP'06)*, Tours, France, June 12–14 2006.

[8] Gunter Rudolph. Convergence of evolutionary algorithms in general search spaces. In *International Conference on Evolutionary Computation*, pages 50–54, 1996.

[9] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.

[10] Denis Bouyssou, Thierry Marchant, Marc Pirlot, Alexis Tsoukiàs, and Philippe Vincke. *Evaluation and decision models with multiple criteria: Stepping stones for the analyst*. International Series in Operations Research and Management Science, Volume 86. Boston, 1st edition, 2006.

[11] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proc. of the International Conference on Genetic Algorithms and Their Applications*, pages 93–100, Pittsburgh, PA, 1985.

[12] A. Dias and J. Vasconcelos. Multiobjective genetic algorithm applied to solve optimization problems, 2002.

[13] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, Reading, MA, January 1989.

[14] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[15] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, 1994. IEEE Service Center.

[16] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Genetic Algorithms: Proceedings of the Fifth International Conference*, pages 416–423. Morgan Kaufmann, 1993.

[17] S. Poles. Moga-II an improved multi-objective genetic algorithm. Technical Report 003-006, ESTECO, Trieste, 2003.

[18] *modeFRONTIER version 3 Documentation*. See also URL `http://www.esteco.com`.

[19] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley, New York, 546 pp, 1986.

[20] Martin Meckesheimer. *A FRAMEWORK FOR METAMODEL-BASED DESIGN: SUBSYSTEM METAMODEL ASSESSMENT AND IMPLEMENTATION ISSUES*. PhD thesis, Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, Pennsylvania State University, 2001.

[21] Michael E. Mortenson. *Modelli geometrici in computer graphics*. McGraw-Hill, Milano, 1989.

[22] Gerald Farin. *Curves and Surfaces in Computer-Aided Geometric Design: A Practical Guide*. Academic Press, San Francisco, CA, fifth edition, 2002.

[23] R. Farouki and V. Rajan. On the numerical condition of polynomials in bernstein form. *Computer Aided Geometric Design*, 4(3), 1987.

[24] R. Farouki and V. Rajan. Algorithms for polynomials in bernstein form. *Computer Aided Geometric Design*, 5(1), 1988.

[25] R. Farouki. On the stability of transformations between power and bernstein polynomial forms. *Computer Aided Geometric Design*, 8(1), 1991.

[26] Frank Uhlig Gisela Engeln-Müllges. *Numerical algorithms with Fortran*. Springer-Verlag, Berlin, 1996.

[27] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, Berlin, second edition, 1997.

[28] R. Farouki. Closing the gap between cad model and downstream application. *SIAM news*, 32(5), 1999. URL `http://www.siam.org/siamnews`.

[29] S. V. Patankar, C. H. Liu, and E. M. Sparrow. Fully developed flow and heat transfer in ducts having streamwise-periodic variations of cross-sectional area. *ASME J. Heat Transfer*, 99:180–186, May 1977.

[30] R. K. Shah and A. L. London. *Laminar Flow Forced Convection in Ducts*. Academic Press, New York, 1978.

[31] E. Stalio and E. Nobile. Direct numerical simulation of heat transfer over riblets. *Int. J. Heat and Fluid Flow*, 24:356–371, 2003.

[32] A. Barletta and E. Zanchini. The existence of an asymptotic thermally developed region for laminar forced convection in a circular duct. *Int. J. Heat Mass Transfer*, 39(13):2735–2744, 1996.

[33] *MATLAB 7.0.4 Documentation*. See also URL `http://www.mathworks.com`.

[34] C. Nonino and G. Comini. Finite-element analysis of convection problems in spatially periodic domains. *Numerical Heat Tranfer*, Part B(34):361–378, 1998.

[35] O. C. Zienkiewicz and R. L. Taylor. *The finite element method, Fluid dynamics*. Butterworth-Heinemann, Oxford, fifth edition, 2000.

[36] *FEMLAB 3 Documentation*. See also URL `http://www.comsol.com`.

[37] H. Ugail. Parametric design and optimisation of thin-walled structures for food packaging. *Journal of Optimization and Engineering*, 4(4):291–307, 2003.

[38] D. R. Sawyers, M. Sen, and H. C. Chang. Heat transfer enhancement in three-dimensional corrugated channel flow. *Int. J. Heat Mass Transfer*, 41:3559–3573, 1998.

[39] C. Poloni and V. Pediroda. GA coupled with computationally expensive simulations: tools to improve efficiency. In D. Quagliarella, J. Periaux, C. Poloni, and G. Winter, editors, *Genetic algorithms and evolution strategy in engineering and computer science: recent advances and industrial applications*, chapter 13, pages 267–288. John Wiley & Sons, Chichester, UK, 1996.

[40] G.S. Dulikravich and T.J. Martin. *Advances in NUmerical Heat Transfer*, volume 1, chapter Inverse Shape and Boundary Condotion Problems and Optimization in Heat Conduction, pages 381–426. Taylor & Francis, London, 1997.

[41] R. Alsan Meric. Shape design sensitivity analysis and optimization for nonlinear heat and electric conduction problems. *Numerical Heat Transfer*, 34, part A(2):185–203, 1998.

[42] Chin-Hsiang Cheng and Chun-Yin Wu. An approach combining body-fitted grid generation and conjugate gradient methods for shape design in heat conduction problems. *Numerical Heat Transfer*, 37, part B:69–83, 2000.

[43] Chin-Hsien Lan, Chin-Hsiang Cheng, and Chun-Yin Wu. Shape design for heat conduction problems using curvilinear grid generation, conjugate, and redistribution methods. *Numerical Heat Transfer*, 39, part A:487–510, 2001.

[44] Chin-Hsiang Cheng and M. H. Chang. Shape design for a cylinder with uniform temperature distribution on the outer surface by inverse heat transfer method. *Int. J. Heat and Mass Transfer*, 46:101–111, 2003.

[45] G. D. Raithby A. Ashrafizadeh and G. D. Stubley. Direct design of shape. *Numerical Heat Transfer*, 41, part B:501–520, 2002.

[46] P. M. Gresho. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. part 1: Theory. *Int. J. for Numerical Methods in Fluids*, 11:597–620, 1990.

[47] E. Nobile. Simulation of time-dependent flow in cavities with the additive-correction multigrid method, part I: Mathematical formulation. *Numerical Heat Transfer, Part B*, 30:341–350, 1996.

[48] P. Blondeau, M. Spérandio, F. Allard, Night Ventilation for Building Cooling in Summer, *Solar Energy*, vol 61 n 5 (1997), pp. 327–335.

[49] B. Givoni, Effectiveness of mass and night ventilation on lowering the indoor daytime temperatures. Part I: 1993 experimental periods Energy and Buildings , 28 (1998), pp 25–32.

[50] V. Geros, M. Santamouris, A. Tsangasoulis, G. Guarracino Experimental evaluation of night ventilation phenomena *Energy and Building*, 29 (1999), pp. 141-154

[51] R. Becker, M. Paciuk Inter-related effects of cooling strategies and building features on energy performance of office buildings *Energy and Building*, 34 (2002), pp. 25–31.

[52] J.Pfafferott, S. Herkel, M. Jäschke Design of passive cooling by night ventilation: evaluation of a parametric model and building simulation with measurements *Energy and Buildings*, 35 (2003), pp. 1129–1143.

[53] J.A. Wright, H. A. Loosemore, R. Faramani, Optimization of building thermal design and control by multi-criterion genetic algorithm *Energy and Buildings*, 34 (2002), pp. 959–972.

[54] W. Huang, H.N. Lam, Using genetic algorithms to optimize controller parameters for hvac systems *Energy and Buildings*, 26 (1997) 277–282.

[55] M. Wetter, J. Wright Acomparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization *Building and Environment*, 39 (2004), pp. 989–999.

[56] Office of Integrated Analysis and Forecasting. International energy outlook 2006 (ieo2006). Technical report, Energy Information Administration, US Department of Energy, Washington DC, 2006.

[57] The Environmentla Building Partnership Ltd. Ebp tech bulletin 1: Energyflo ^{TM} cell and the dynamic breathing building (dbb) system. 2005. Rev. I.

[58] B.J. Taylor, D.A. Cawthorne, and M.S. Imbabi. Analytical investigation of the steady-state behaviour of dynamic and diffusive building envelopes. *Building and Environment*, 31(6):519, 1996.

[59] B.J. Taylor and M.S. Imbabi. The effect of air film thermal resistance on the behaviour of dynamic insulation. *Building and Environment*, 32:397, 1997.

[60] B.J. Taylor, R. Webster, and M.S. Imbabi. The building envelope as an air filter. *Building and Environment*, 34:353–361, 1999.

[61] B.J. Taylor and M.S. Imbabi. Environmental design using dynamic insulation. In *ASHRAE Transactions*, volume 106, pages 15–28. 2000.

[62] P.H. Baker. The thermal performance of a prototype dynamically insulated wall. *Building Services Engineering Research and Technology (BSER&T)*, 24:25–34, 2003.

[63] A. Dimoudi, A. Androutsopoulos, and S. Lykoudis. Experimental work on a linked dynamic and ventilated wall component. *Energy and Buildings*, 36:443–453, 2004.

[64] A.S. Kalagasidis. The efficiency of a dynamically insulated wall in the presence of air leakeages. *Thermal Science*, 8:83–94, 2004.

[65] M.S. Imbabi and J. Colas. Simulation of airflow in modular breathing wall systems. in Preparation.

[66] British Standards Institute. *Building components and building elements - Thermal resistance and thermal transmittance - Calculation method*, 2003. BS EN ISO 6946:1997.