POLITECNICO DI MILANO
DIPARTIMENTO DI ENERGIA

Turbomachinery Design Optimization using Adjoint Method and Accurate Equations of State

Matteo Pini

Relatore: Prof. Vincenzo Dossena
Coordinatore: Prof. Carlo Bottani

Tesi di Dottorato di Ricerca in Scienze e Tecnologie Energetiche e Nucleari – XXVI Ciclo

Dicembre 2013

Il candidato: _____

Matteo Pini
Matricola 768471

Il relatore: _____

Prof. Vincenzo Dossena

Il coordinatore
del dottorato di ricerca: _____

Prof. Carlo Bottani

Politecnico di Milano
Dipartimento di Energia
via Lambruschini 4 - 20156 Milano

Dottorato di Ricerca in Scienze e Tecnologie Energetiche e Nucleari - XXVI ciclo

Milano, Dicembre 2013

Summary

This research work presents a shape optimization approach for turbomachinery applications based on the adjoint method and accurate equations of state for the thermo-physical description of the fluids. The nature of the research is numerical, hence most of the work expense has been dedicated to the development of the tools embedded in the optimization framework. The algorithm proposed is based on a discrete inviscid adjoint method able to treat real-gas flows, state-of-the-art parametrization techniques (NURBS) and a preconditioned steepest descent optimizer to reach the optimal point. The mathematical approach followed is extensively described in chapter 2 and 3. Built-in equations of state and look-up tables (LuT) are introduced within the optimization algorithm to handle real-gas effects. In particular, a novel consistent LuT method, discussed in chapter 4, is devised.

The potential of the optimization approach is investigated by resorting to different aerodynamic and turbomachinery 2D design problems, detailed in chapter 5. The design methodology is initially applied to the re-design of a wind tunnel nozzle operating under both ideal and real gas flows.

Then, two different turbomachinery design examples are reported. The former focuses on the maximization of the performances of a transonic cascade and represents the earliest test case performed in this research. The latter aims at optimizing an existing converging-diverging turbine cascade operating under supersonic conditions, with the objective of achieving a more uniform flow at the blade outlet section. Real-gas equations of state are herein considered, properly introduced at both flow and adjoint level through the LuT approach.

An original extension of the algorithm for treating off-design conditions is also envisaged in the present research work. The method devised combines a standard multi-point optimization technique with an uncertainty quantification algorithm to assess the design points and the weights of the multi-point problem. The capability of the novel approach in providing robust designs is finally investigated by maximizing the performances of the mentioned supersonic blade configuration, working under a relatively wide range of operating conditions.

In all test cases remarkable outcomes are achieved in terms of improvement of performances of the initial configuration and computational efficiency.

Most of the applications of adjoint method are still restricted to shape optimization, however new perspectives have been recently risen on the use of adjoint. In this regard, the last chapter is devoted to illustrate the potential of adjoint-based methods for uncertainty quantification and robust optimization. The study is carried out by taking a quasi-1D transonic problem as reference test case.

Acknowledgments

This thesis is dedicated to whom encouraged me during the experience of the PhD course. Three years of exciting activities and satisfactions, awarded by the tenure track position appointed at Delft University of Technology within the Department of Flight Performance and Propulsion. A great acknowledgment to my love Daniela, my mother and father Maura and Sergio, and my little sister Giuditta for the unceasing support in every moment of the work.

A kindly acknowledgment is also due to Prof. V. Dossena, Prof. G. Persico, Prof. S. Rebay, and Ing. D. Pasquale for all the fruitful discussions during the development of this thesis. I moved the first steps into the hard but fascinating world of numerical fluid-dynamics at Brescia University three years ago under the teachings of Prof. Rebay, and presently I am very passionate in trying new horizons for computational simulation and automatic shape design.

Contents

Chapter 1

Introduction

Fluid-Dynamic Optimal Shape Design (FOSD) is the art of improving the performances of realistic devices immersed in fluid media. A fascinating blend of theory and computation, heuristics and rigor.

1.1 The Problem of Optimal Shape Design

Optimal Shape Design (OSD) is nowadays a necessity in many industrial fields. In aircraft design the reduction of few percent of drag means a lot of fuel saving, car industries are strongly involved in the limitation of structure weights and in the improvement of vehicle aerodynamics, whereas energy power sector gains considerable benefits from efficiency maximization of power systems and system components (e.g. turbomachinery). However, OSD is still a difficult and computer-demanding task. Many challenges remain from the mathematical point of view to obtain design algorithms capable of solving multi-disciplinary problems, complex physical models (e.g. turbulence models) or unsteady situations with reasonable computational resources. Among OSD subjects, Fluid-Dynamic Optimal Shape Design (FOSD) assumes a relevant role in the field of Computational Fluid-Dynamics (CFD). There are many situations in which achieving the optimal solution represents a very demanding operation without the use of automatic design tools, also for skilled designers. Wing design, blade design or design of microfluidic devices are three examples of fluid-dynamic components where FOSD is intensively applied to enhance their performances. In aeronautics, high lift configurations are also challenging because the flow needs to be accurately solved and turbulence approaches like DES or LES are still too demanding to be included in the design loop. Also, shape optimization for unsteady flows is only at an early stage.

In mathematical terms, a fluid-dynamic optimal shape design (FOSD) requires the optimization of one or more fitness function J which in turn depend on a set of design parameters $\boldsymbol{\alpha}$ defining the shape to be optimized. A single-objective deterministic optimization problem can be formulated as

$$\min \quad J(\boldsymbol{\alpha}) \quad \text{subject to} \quad \begin{cases} c(\boldsymbol{\alpha}) & 0 \\ d(\boldsymbol{\alpha}) & \geq 0. \end{cases} \tag{1.1}$$

where $c(\boldsymbol{\alpha}), d(\boldsymbol{\alpha})$ are, respectively, the equality and inequality constraints for the fitness function $J(\boldsymbol{\alpha})$. In unconstrained optimization, the fitness function is minimized (maximized) with no restriction at all on the values of design variables. Constrained optimization, conversely, approaches the optimal configuration satisfying a series of criteria (e.g. on geometry, on physical properties) that limit the space of possible solutions.

Generally, optimization problems can be either local or global. Global optimization algorithms are became of paramount interest in FOSD as they are able to treat noisy objective functions without requiring informations on continuity, existence of derivatives and uni-modality of the functions themselves. They directly allow the specification of equality and inequality constraints,

and normally require a huge amount of function evaluations to reach the optimum. The major drawback of such algorithms is in fact the slow convergence which couples with the intrinsic difficulty in finding the real global optimum of flat functions. An exhaustive analysis of global optimization for CFD applications and the related algorithms can be found in several books and works, for instance [1, 2, 3, 4, 5].

On the other hand, local optimization algorithms perturb a known initial configuration according to indications usually given by the gradient ∇J or by a combination of gradient itself and higher-order sensitivities (commonly the hessian $\nabla^2 J$). A variety of local optimization algorithms has been invented for functionals of a finite number of variables. A thorough description of them can be found in [6]. Herein only the method of steepest descent, extensively applied in this thesis, is briefly recalled. By locally approximating the fitness function J through the Taylor series expansion the following relation holds

$$J(\boldsymbol{\alpha} + \delta\boldsymbol{\alpha}) = J(\boldsymbol{\alpha}) + (\delta\boldsymbol{\alpha})^T \nabla J + \frac{1}{2}(\delta\boldsymbol{\alpha})^T \nabla^2 J(\delta\boldsymbol{\alpha}) \tag{1.2}$$

If the expansion is truncated at the first-order term, the method of steepest descent is recovered. The resulting fixed-step size formula to update the geometry at each design step can be written as

$$\begin{aligned} \delta\boldsymbol{\alpha} &= -\lambda \nabla J \\ (\boldsymbol{\alpha})^{k+1} &= (\boldsymbol{\alpha})^k + \delta\boldsymbol{\alpha}. \end{aligned} \tag{1.3}$$

The parameter λ is a small number that forces the algorithm to make small jumps. Stable conditions are realized for a proper choice of λ which guarantees that $J(\boldsymbol{\alpha} + \delta\boldsymbol{\alpha}) \leq J(\boldsymbol{\alpha})$. A graphical visualization of the steepest descent method is highlighted in Figures 1.1 and 1.2, together with the influence of the step-size λ. In Figure 1.1 a correct selection of the step-size enables a smooth descent towards the optimum, while increasing its value produces a zig-zag descent, see Figure 1.2. A too large λ leads, conversely, to a divergence of the method.

The estimation of the best step-size value can be attempted by many techniques normally called line search methods [7]. Adaptive solution usually work well at expense of a high increase of the overall computational burden of the optimization process. However, some famous pioneers of shape design for fluids, like A. Jameson and O. Pironneau, suggest that the optimal step-size is not necessarily a good idea in FOSD. It seems better to think in terms of dynamical systems attached to the minimization problems, as proposed in [8]. Thinking in these terms means adjusting by manual or automatic trial and error operations the choice of the step-size to the specific shape optimization problem. As an example, a first selection of λ can be based

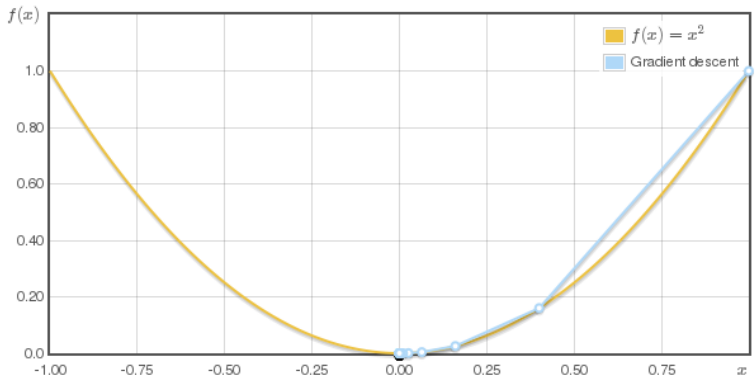


Figure 1.1. *Descent trend toward the minimum using a proper step-size.*

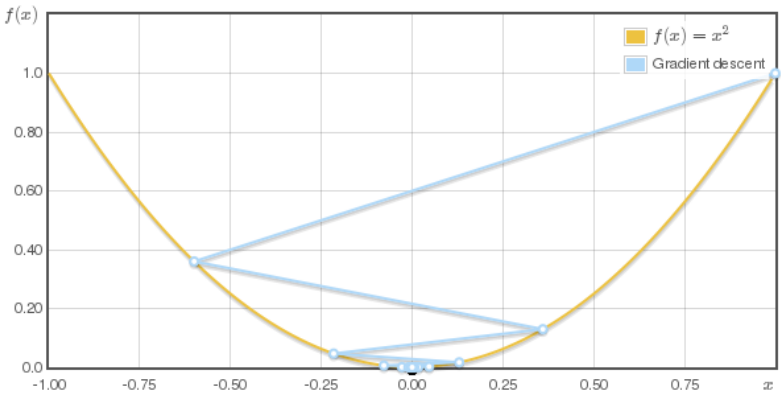


Figure 1.2. *Descent trend toward the minimum using a poor accurate step-size.*

on purely geometrical criteria concerning some characteristic length of the shape of interest. Such an approach is heuristic, often based on experience, but should be preferred in case of fluid-dynamic shape optimization, where even slightly changes of the geometry may lead to a failure of the convergence of the underlying numerical solvers.

Despite the concepts expressed thus far generally hold for all categories of FOSD problems, in the following the attention will be restricted to FOSD for turbomachinery applications.

1.1.1 Optimal Shape Design in Turbomachinery

Optimization techniques play a key role in the present-day design process of turbomachinery; thanks to the progressive increase of computational capability, high-fidelity solution methods based on Computational Fluid Dynamics

(CFD) are now routinely applied within these optimization algorithms, resulting in automatic design tools.

The turbomachinery design process presents optimization challenges at many levels, resulting in a step-by-step procedure of increasing fidelity level. The final turbomachinery layout is then usually a complex combination of the outcomes provided by each optimization step, starting from the preliminary choice of the number of stages and of the stage arrangement [9], to the throughflow (or axisymmetric) design [10, 11, 12], up to the detailed blade shape definition [13, 3].

In the last decades inverse design methods [14, 15] and CFD-based shape optimization procedures were specifically developed for turbomachinery applications. CFD-based shape optimization methods are, in many cases, supported by the application of stochastic methods, such as evolutionary algorithms (EAs) [16]. These techniques usually search for the optimal shape by resorting to genetic algorithms (GA) coupled to surrogate models to reduce the overall computational cost of the optimization procedure [17, 18]. Such an approach allows to explore a wide range of feasible solutions, identifying the best individual (i.e. global minimum in some circumstances), and handling multi-objective optimization problems [19, 20, 21, 22].

However, if complex aerodynamic configurations are of concern, the cost of the overall optimization process may become prohibitive and this practically limits the available design space and the number of possible solutions. Furthermore, in many cases the performance improvement of realistic turbomachinery cascades is achieved by optimizing existing blade geometries in presence of mechanical and geometrical constraints; as a consequence, the optimal shape cannot be significantly different from the initial configuration and minor geometry modifications are generally acceptable. In this context, deterministic gradient-based design techniques provide relevant improvements in terms of computational cost and available extension of the design space. These methods automatically determine the optimal configuration by a relatively low number of steps, outperforming GAs for these type of problems, provided that an efficient way is introduced to calculate the gradient. In general the aerodynamic design of turbomachinery requires a large number of design variables, while relatively simple objective functions are to be minimized; for such problems the adjoint approach represents the most suitable and effective technique to compute the gradient [23]. Adjoint-based deterministic methodologies, hence, can represent one of the most suitable choice for the optimization of existing turbomachinery cascades. In this perspective, the potential of adjoint approach for turbomachinery applications has been investigated in some early works, but only recently has become more popular. The early works were based on the continuous adjoint formulation,

see for example [24], [25], while the recent availability of automatic differentiation tools has offered the possibility of developing discrete adjoints, as carried out in [26], [27], [28]. However, publications regarding turbomachinery shape design using adjoint are still a few compared to the vast collection of works related to stochastic (evolutionary) optimization.

1.1.2 Optimal Shape Design in ORC Turbomachinery

The vast majority of researches focused on FOSD in turbomachinery mainly consider gas turbine technology. In fact, as already mentioned, the problem of optimal design is of extremely relevance in aircraft industry, whereby not only purely fluid-dynamic effects but also heat transfer mechanisms are to be taken into account in the optimization of turbine blades. The efficiency of turbine cascades is also a key theme for large-scale applications using supercritical steam or condensing flows (e.g. nuclear turbines) as working fluids.

However, in the energy field, alternatively to the current paradigm of centralized power conversion, a renewed interest has recently arisen in decentralized power generation, whereby energy needing is partly fulfilled by a network of relatively small units. At the same time, abundant renewable sources are often characterized by a comparatively small exploitable power capacity of the single site [29, 30]. As a consequence, small-to-medium scale power plants (from few kW up to few MW) exploiting renewable sources are candidates to play a relevant role in this future distributed energy scenario. Among the available technologies for high-efficiency conversion of thermal power into electricity and/or heat in this range of capacity, Organic Rankine Cycles (ORC) stand out in terms of reliability and cost-effectiveness, and have been demonstrated to be advantageous compared to steam Rankine cycles [31, 32].

This is mainly a consequence of the increase of specific cost of system components as the scale of the plant reduces. From this perspective, the use of organic fluids, characterized by high molecular weight, can make available cost-effective solutions for the turbo-expander, together a whole simplification of the plant arrangement.

The specific enthalpy drop, along the turbine expansion line, is inversely proportional to the molecular weight of the fluid. This determines two main advantages in case of organic fluids: primarily the relatively small specific work can be disposed in a low number of stages; secondarily, for a target power output, a relatively large mass flow rate is required, resulting in an enlarged size of small-capacity ORC turbines with respect to steam units [33].

On the other hand, the low number of stages leads to high expansion

ratios per stage; this, combined with the low speed of sound, leads to the widespread application of transonic and supersonic turbines in ORC systems. As a result, strong shock waves and chocked flow conditions are commonly found in these machines, complicating the design and operation of ORC systems. Moreover part of the expansion process usually occurs in close proximity of the saturated vapor curve, or even close to the critical point. In such thermodynamic conditions complex equations of state are necessary to accurately describe the behavior of organic working fluids.

This design scenario is further complicated by the lack, in the open literature, of experimental data regarding flows of organic fluids [34, 35], especially in the thermodynamic region of interest. Reliable design tools, such as efficiency prediction methods based on empirical loss correlations, and well-established size criteria are not still available for ORC turbines. Their dimensioning is mostly based on guidelines originally conceived for gas or steam machines, even in the selection of aerodynamic blade profiles, hence there are very often considerable margins of improvement of their performances.

1.2 Motivation of the Research Work

Considering the aforementioned difficulties, improved designs of ORC turbines are of significant interest; these improvements require the development of specific design procedures, essentially based on CFD and accurate EoS models, and the evaluation of non-conventional turbine architectures. A number of work address the second issue, for instance see [9], whereby the authors proposed a multi-stage centrifugal turbine architecture for ORC applications designed by means of a combination of 1-D/throughflow approaches. Conversely, very few research papers are dedicated to the implementation of automatic shape optimization algorithms specifically targeted to ORC turbine cascades. Some pioneering works on this topic can be found in [36, 37]. All authors undertake the optimization problems using evolutionary algorithms for the re-design of existing transonic/supersonic blades, whereas no research works have been published on adjoint methods for real gas flows. In the field of ORC, adjoint methods can assume an increasing importance due to the considerable requirements of computational resources to perform accurate real-gas CFD simulations, which can make EAs (GAs) less attractive compared to other turbomachinery applications.

The lack of published results and the assessment of the theoretical huge potential of adjoint methods for real-gas flows motivated the present thesis, which is inserted in a more comprehensive research program focused on design techniques for ORC turbines, currently ongoing within the Fluid-Dynamics

of Machines research group in a framework of collaboration with University of Brescia. Main aim of the present work is therefore the development of a 2D shape optimization approach based on adjoint method for real-gas inviscid flows.

1.2.1 Structure of the Thesis

The present thesis is divided into five main chapters as follows:

- *Chapter 2* addresses the issue of efficiently computing sensitivities in Computational Fluid-Dynamics. In particular an overview on the *present-day* methodologies is initially given, followed by an extensive description of the solution developed in this research.
- *Chapter 3* illustrates the features of the numerical tools composing the optimization loop. The main characteristics of the flow and adjoint solvers are firstly described, then the overall design loop for single and multi-point optimization is widely discussed.
- *Chapter 4* presents the look-up table approach (LuT) developed within this work that constitutes the basis for the high-fidelity simulations and design processes under real gas flow conditions.
- *Chapter 5* reports the most relevant outcomes of shape optimization applied to wind tunnel nozzles and turbomachinery blades, working under both ideal and real gas conditions. An original strategy for effectively managing the fluctuations of the operating conditions within the design process is also remarked.
- *Chapter 6* presents a comprehensive analysis of the potential of adjoint method for uncertainty quantification (UQ) and robust design optimization, (RDO), which are relatively novel research fields in CFD. Therefore this chapter represents an extension of the thesis to show new applications of adjoint. A simple model problem is used as a benchmark to investigate the performances of several UQ algorithms. At the end of the section a novel hybrid robust optimization approach for fluid-dynamic problems is proposed.

1.2.2 Major Novelties of the Research

The distinctive features of the present research work can be summarized as follows:

- *Adjoint Method*

  Most of the novelties of this work are related on the adjoint solver. The adjoint inherits the peculiarities of the native flow solver, such

as the node-pair formulation and the high resolution scheme based on the Vinokur-Montagné approach suited for real gas flows. As a result, the adjoint solver is capable to deal with real-gas effects as well. To the author's knowledge, no adjoint method for arbitrary EoS has been published yet, then this represents a distinguishing aspect of the work.

- *Thermodynamic Modeling of Real-Gases*

  A consistent look-up table approach has been developed and coupled with the flow and adjoint solvers for a generalized thermodynamic treatment. The satisfaction of the thermodynamic consistency featured by the method represents an original extension of well-established approaches proposed by other researchers. Moreover, the use of a LuT approach makes the choice of the thermodynamic library completely free. State-of-the-art and novel thermodynamic models can then be easily accounted for in the optimization loop.

- *Shape Parameterization*

  An advanced parameterization technique based on NURBS curves is introduced to iteratively modifies the geometry according to the gradient direction. The adoption of NURBS curves for transonic/supersonic flow problems is still at an early stage in adjoint-based shape optimization.

- *Shape Optimization*

  Original applicative examples have been selected to investigate the performances of the design algorithm, namely the re-design of an existing supersonic turbine cascade for ORC applications and the re-design of a supersonic wind tunnel nozzle to be installed in a facility designed for dense gas-dynamics studies.

- *Innovative Applications of Adjoint*

  The potential of the adjoint method as the basis for innovative uncertainty quantification and robust design algorithms is investigated in the present work. The use of sensitivity derivatives in the context of robust optimization has undergone a very limited application. Also, a comprehensive comparison among robust design strategies is still an open issue in the field of advanced aerodynamic design. Part of the thesis is therefore dedicated to face this problem, with original outcomes.

Chapter 2

Sensitivity Analysis for Shape Optimization

The calculation of sensitivity derivatives still represents an entangling challenge in the field of Computational Fluid-Dynamics. This chapter, focusing on modern strategies for computing sensitivities, aims to be a distinguishing contribution for those who approach this hard issue.

2.1 Sensitivity Derivatives

In Computational Fluid-Dynamics (CFD) sensitivity derivatives are to be intended as the derivatives of aerodynamic quantities, typically depending on flow, with respect to some parametrization of the geometry or some physical variables (e.g. the boundary conditions of the flow problem). Sensitivity derivatives have widespread benefits in aerodynamic problems. Firstly, they are suitable for the analysis of complex problems in which the reliable identification of the sensitivity of quantities of interest with respect to some physical or geometrical parameter is of paramount importance, for instance in control flow problems [38]. Moreover, they represent the basis of many shape design procedures. In fact, they can be easily utilized within gradient-based algorithms or as a support to enhance the accuracy of response surface techniques when gradient-free methods (e.g. genetic algorithms) are preferred. Lastly, they are employed in uncertainty quantification methods to predict statistical moments of the quantity of interest [39], [40]. Despite the key role played in many applications, sensitivity calculation has been only recently widely adopted in engineering practice. The most critical aspect for sensitivity evaluation, for a long time, has been essentially the lack of efficient methods for determining the gradient of fitness functions. A considerable computational effort is in fact usually necessary to obtain accurate values, especially if traditional solutions, as finite difference or discrete direct methods (reported in the following), are applied.

Alternatively, the control theory or the adjoint method, pioneered by Anthony Jameson in 1995, proved to be very effective to overcome the drawbacks of the previously mentioned methods. In this case, the gradient is obtained by resorting to a suitably defined adjoint problem which can be formulated and solved with a computational cost that is independent from the number of design parameters. For this reason, this method is very attractive for CFD design problems, which typically involve many design variables. Nevertheless, in spite of the relevant number of advantages, some factors have limited the diffusion of adjoint methods in the fluid-dynamic community. The difficulty of implementing continuous adjoint formulations and the derivation of flow solvers by hand have been the two predominant factors that have lowered the spread of such approach. Only recently, the availability of automatic differentiation tools has truly contribute to a wide interest into the adjoint method and sensitivity evaluation.

The main techniques which can be employed to compute the gradient of a cost functional are described in the following sections, with particular emphasis on the discrete adjoint method.

2.1.1 Finite Difference Techniques

The finite difference methods (FDM) can be regarded as the standard approach for the computation of the sensitivities of a fitness function. FDM are a non-intrusive method, as they require no modifications of the original flow solver. They only need the evaluation of the fitness function for each perturbation of the geometrical variables $\boldsymbol{\alpha}_{geo}$ or physical variables $\boldsymbol{\alpha}_{phy}$.

Let J be the cost function to be minimized in an optimization problem. In fluid-dynamics J is generally depending on a vector of physical and geometrical design variables $\boldsymbol{\alpha}_{phy}$ and $\boldsymbol{\alpha}_{geo}$, respectively, state variables $\mathbf{u}$, and grid points $\mathbf{X}$. State variables, in turn, depend on both physical and geometrical design variables, while grid variables are only function of the geometrical design variables. The functional J can therefore be written as

$$J = J[\boldsymbol{\alpha}_{phy}, \mathbf{u}(\boldsymbol{\alpha}_{phy}, \boldsymbol{\alpha}_{geo}), \mathbf{X}(\boldsymbol{\alpha}_{geo})]. \tag{2.1}$$

The state variables $\mathbf{u}$ are also constrained to satisfy the fluid-dynamic governing equations. For steady-state problems, the governing equations can be symbolically expressed as

$$\mathbf{R}[\boldsymbol{\alpha}_{phy}, \mathbf{u}(\boldsymbol{\alpha}_{phy}, \boldsymbol{\alpha}_{geo}), \mathbf{X}(\boldsymbol{\alpha}_{geo})] = 0, \tag{2.2}$$

and can be regarded as an equality constraint for the minimization problem. A single gradient component is then computed by perturbing one of the independent variables of the problem. If $\delta\boldsymbol{\alpha}_{geo}$ represents a shape perturbation of the body to be optimized, the gradient of the cost functional, using a centered second-order accurate formula, can be written as

$$\begin{aligned}\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\alpha}_{geo}} = \frac{1}{2}\{&J[\boldsymbol{\alpha}_{phy}, \mathbf{u}(\boldsymbol{\alpha}_{phy}, \boldsymbol{\alpha}_{geo} + \delta\boldsymbol{\alpha}_{geo}), \mathbf{X}(\boldsymbol{\alpha}_{geo} + \delta\boldsymbol{\alpha}_{geo})] - \\ &J[\boldsymbol{\alpha}_{phy}, \mathbf{u}(\boldsymbol{\alpha}_{phy}, \boldsymbol{\alpha}_{geo} - \delta\boldsymbol{\alpha}_{geo}), \mathbf{X}(\boldsymbol{\alpha}_{geo} - \delta\boldsymbol{\alpha}_{geo})]\}\end{aligned} \tag{2.3}$$

The previous equation implies two new flow solutions on the modified mesh, expressed as

$$\mathbf{R}[\boldsymbol{\alpha}_{phy}, \mathbf{u}(\boldsymbol{\alpha}_{phy}, \boldsymbol{\alpha}_{geo} \pm \delta\boldsymbol{\alpha}_{geo}), \mathbf{X}(\boldsymbol{\alpha}_{geo} \pm \delta\boldsymbol{\alpha}_{geo})] = 0. \tag{2.4}$$

Despite its simplicity this approach requires large computational resources (total cost is $2 \times N_{\boldsymbol{\alpha}_{geo}}$ for the second order formula) and is problematic in the choice of the step-length needed by the finite difference procedure. If it is too small the rounding errors become significant, conversely a too large step size involves a poor accuracy of the Taylor expansion.

2.1.2 The Adjoint Method for Fluid-Mechanics

In this section the mathematical fundamentals of the adjoint method are briefly recalled. Both continuous and discrete approaches to the adjoint method are available [8]. In this work we only consider the second approach, in which the flow equations are firstly discretized and then derived by Automatic Differentiation (AD), also called Algorithmic Differentiation. A general mathematical formulation of the discrete adjoint approach, including the grid deformation effects, is now extensively presented. The formulation here described enables to compute the sensitivity derivatives of the fitness function which depend on the flow equations of motion solved on either structured and unstructured meshes. All the vectors specified in the following are conventionally assumed as column vectors.

Notice that equation (2.2) holds for an arbitrary choice of the design variables $\boldsymbol{\alpha}_{phy}$ and $\boldsymbol{\alpha}_{geo}$. This implies that the total derivatives of (2.2) with respect to $\boldsymbol{\alpha}_{phy}$ and $\boldsymbol{\alpha}_{geo}$, reported in (2.5), are identically null — as long as $\mathbf{u}$ is the solution of Eq. (2.2) for any $\boldsymbol{\alpha}_{phy}$ and $\boldsymbol{\alpha}_{geo}$.

$$\begin{aligned} \frac{\mathrm{d}\mathbf{R}}{\mathrm{d}\boldsymbol{\alpha}_{geo}} &= \frac{\partial \mathbf{R}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \boldsymbol{\alpha}_{geo}} + \frac{\partial \mathbf{R}}{\partial \mathbf{X}} \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\boldsymbol{\alpha}_{geo}} = 0 \\ \frac{\mathrm{d}\mathbf{R}}{\mathrm{d}\boldsymbol{\alpha}_{phy}} &= \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}_{phy}} + \frac{\partial \mathbf{R}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \boldsymbol{\alpha}_{phy}} = 0. \end{aligned} \tag{2.5}$$

This is a crucial mathematical observation that will be used for the derivation of the adjoint problem.

The fitness function and the constraint, defined according to the above expressions, are written considering a general problem, in which both the physical quantities and the shape of the geometry of interest can be modified in the procedure. However, in standard shape optimization problems, the physical variables appearing in expressions (2.1) and (2.2) are to be considered as fixed known parameters (i.e. are not considered as variables in the design process). The gradient of the fitness function has to be computed with respect to the geometrical design variables $\boldsymbol{\alpha}_{geo}$ only, and is therefore expressed as

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\alpha}_{geo}} = \frac{\partial J}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \boldsymbol{\alpha}_{geo}} + \frac{\partial J}{\partial \mathbf{X}} \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\boldsymbol{\alpha}_{geo}}. \tag{2.6}$$

The relation (2.1) shows that the gradient is a function of the first-order state sensitivity derivatives $\partial \mathbf{u}/\partial \boldsymbol{\alpha}_{geo}$. The direct calculation of these terms is usually a very demanding operation in terms of computational cost. As already discussed in 2.1.1, the simplest but computationally prohibitive method to compute $\partial \mathbf{u}/\partial \boldsymbol{\alpha}_{geo}$ is the finite difference technique (FDM). As an

alternative $\partial \mathbf{u}/\partial \boldsymbol{\alpha}_{geo}$ can be obtained by solving the first equation of (2.5), rewritten as

$$\frac{\partial \mathbf{u}}{\partial \boldsymbol{\alpha}_{geo}} = -\left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right)^{-1}\left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\boldsymbol{\alpha}_{geo}}\right). \tag{2.7}$$

However the calculation of (2.7) requires a considerable CPU memory utilization as a series of N decoupled linear systems (with N the columns of the constant matrix $\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\boldsymbol{\alpha}_{geo}}$) has to be solved. This calculation methodology is often called the *discrete direct method* and it was pursued by many authors in the past. A more efficient strategy is achieved by plugging the previous relation into equation (2.6). The resulting gradient can be therefore expressed in the form

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\alpha}_{geo}} = -\frac{\partial J}{\partial \mathbf{u}}\left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right)^{-1}\left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\boldsymbol{\alpha}_{geo}}\right)+\frac{\partial J}{\partial \mathbf{X}}\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\boldsymbol{\alpha}_{geo}}. \tag{2.8}$$

In the previous relation the vector $-\frac{\partial J}{\partial \mathbf{u}}(\frac{\partial \mathbf{R}}{\partial \mathbf{u}})^{-1}$ is named adjoint or co-state variables vector. The adjoint variables $\mathbf{v}$ are consequently determined through the resolution of the adjoint linear system, rewritten as

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right)^{T}\mathbf{v} = -\left(\frac{\partial J}{\partial \mathbf{u}}\right)^{T}. \tag{2.9}$$

Equation (2.9) represents the discrete form of the adjoint equation, see e.g. [41]. The gradient vector of the original objective function is finally expressed, through the above relations, as

$$\left(\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\alpha}_{geo}}\right)^{T} = \left(\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\boldsymbol{\alpha}_{geo}}\right)^{T}\left[\left(\frac{\partial J}{\partial \mathbf{X}}\right)^{T}+\left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^{T}\mathbf{v}\right]. \tag{2.10}$$

To ease the development of adjoint based design algorithms for CFD applications, great interest has been recently arisen in the exploitation of Algorithmic Differentiation (AD) techniques. AD can in fact be used to directly obtain the matrix-vector products required for the computation of the gradient of an arbitrary fitness function. State-of-the-art CFD solvers adopting unstructured meshes and advanced numerical schemes may be, at least in principle, exactly differentiated. In this way the complex and error-prone hand coding operation traditionally needed for the development of adjoint-based gradient optimization methods can be completely avoided.

Within the present work the AD tool Tapenade, developed at INRIA [42], is applied to the primal equations. To improve the computational efficiency of the differentiation process, the discrete adjoint solver and the gradient evaluator are both constructed by selectively applying the AD tool to the inviscid flow solver zFlow, which will be described in the following chapter.

2.2 Principles of Algorithmic Differentiation

Algorithmic (Automatic) Differentiation (AD) is a method to numerically evaluate derivatives of a function specified by a computer program. AD exploits the fact that any numerical procedure that implements a function can be (generally) decomposed into a sequence of elementary assignments, which can be trivially differentiated by a pre-defined mathematical rules. These elemental derivatives are then assembled using the chain rule from derivative calculus to build gradients, directional derivatives, and Jacobian matrix of the top function. AD has indeed a great potential to overcome the problems limiting the use of other techniques (finite differences, hand differentiation). The main features of AD can be briefly summarized as:

1. It does not require the knowledge of a function but only its implementation.
2. The computed derivatives are exact (in the sense of machine precision).
3. It is reasonably fast, namely it needs a single run to differentiate the source code.
4. By the reverse mode, it has the ability to compute the gradient of a functional at a cost that is independent from the number of design variables.

AD can be performed in two different modes known as *forward/tangent* or *backward/reverse*. To better understand how AD practically works, a function of two independent variables is taken as reference. A special formalism is also introduced, namely the arrow over the parameter indicates an input (independent) variable, whereas the arrow below the parameter highlights the output (dependent) quantity. Let be `numflux` a generic subroutine written in Fortran/C++ computing the numerical flux for a discretization scheme (for instance the Godunov or Lax-Wendroff numerical flux). Using the standard Fortran notation, `numflux` is generally expressed as

$$\texttt{subroutine numflux}\,(\boldsymbol{\alpha}, \mathbf{u}, \mathbf{f}),$$

where $\boldsymbol{\alpha}, \mathbf{u}$ are, respectively, the input vectors of parameters and states, while being $\mathbf{f}$ the output numerical flux vector. Let us suppose to apply the direct/tangent differentiation to the previous numerical procedure. AD builds a new subroutine, normally identified by the suffix $\_d$, appearing as

$$\texttt{subroutine numflux\_d}\,(\boldsymbol{\alpha}, \overset{\downarrow}{\boldsymbol{\alpha}}_d, \mathbf{u}, \overset{\downarrow}{\boldsymbol{u}}_d, \mathbf{f}, \underset{\downarrow}{\boldsymbol{f}}_d),$$

where $\mathbf{f}_d$ is the differentiated output quantity that can be written as

$$\mathbf{f}_d = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\alpha}} \boldsymbol{\alpha}_d + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \mathbf{u}_d. \tag{2.11}$$

If the inputs/outputs reduce to scalars the tangent mode computes the directional derivatives of f with respect to α, u. As a matter of fact, the partial derivatives $\frac{\partial f}{\partial \alpha}, \frac{\partial f}{\partial u}$ can be found by assigning to α_d, u_d, alternatively, the values $1, 0$ and $0, 1$. Conversely, for vector quantities, the Jacobians $\frac{\partial \mathbf{f}}{\partial \boldsymbol{\alpha}}, \frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ can be iteratively evaluated performing n_c calls (with n_c the number of columns of the Jacobian matrix) of the differentiated routine. Each call, in fact, allows to calculate a single column of the matrix, as long as the input vector has only an element equal to 1, while being null otherwise.

Reverse mode, on the contrary, generates a new routine, identified by the suffix $\_b$, which normally reads as

$$\texttt{subroutine numflux\_b}\ (\boldsymbol{\alpha}, \underset{\downarrow}{\boldsymbol{\alpha}_b}, \mathbf{u}, \underset{\downarrow}{\mathbf{u}_b}, \mathbf{f}, \overset{\downarrow}{\mathbf{f}_b})$$

where the outputs of the routine are $\boldsymbol{\alpha}_b, \mathbf{u}_b$, while being $\mathbf{f}_b$ the input parameter. The new two output vectors are expressed as

$$\begin{aligned} \boldsymbol{\alpha}_b &= \left(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\alpha}} \right)^T \mathbf{f}_b \\ \mathbf{u}_b &= \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^T \mathbf{f}_b. \end{aligned} \tag{2.12}$$

The reverse mode of AD is therefore able to deliver immediately the matrix-vector products $(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\alpha}})^T \mathbf{f}_b, (\frac{\partial \mathbf{f}}{\partial \mathbf{u}})^T \mathbf{f}_b$. For scalar input quantities, reverse differentiation gives the directional derivatives at a cost of one routine call. As a consequence, for a function of n variables, it can be (at least theoretically) n times more efficient than the tangent mode. The reverse mode of AD is then of paramount importance for efficiently adjointing CFD solvers, see e.g. equation (2.9), in which the transposed matrix-vector product is clearly in evidence.

In spite of the relevant advantages, some drawbacks affect the Algorithmic Differentiation. The main drawback of AD is the request, to the user, of a deep knowledge of the source code and it needs a certain amount of work and attention for correctly differentiating the primal procedures. Furthermore, the method is not still mature to be applied to an entire source code, therefore a complex program has to be selectively differentiated to guarantee reasonable computational performances when compared to those of the original version.

Several AD tools have been developed by various academic teams. The most used are ADIFOR, ADOC, TAF and TAPENADE. The last one is employed in the present thesis. TAPENADE [42] is developed and mantained by the Tropics research team of INRIA's Nice Sophia-Antipolis and is freely downloadable from the website:

`http://www-sop.inria.fr/tropics/`

2.3 Design Chain for Shape Optimization

In aerodynamic optimization, the geometry of the body is always represented by a parameterization, namely by modeling its shape through some mathematical representation. The simplest approach consists in directly specifying the geometry of interest by the boundary mesh points, normally interpolated by splines. In the context of shape design, this solution is often labeled as CAD-free (computer aided design-free) technique. State-of-the-art approaches, conversely, allow to define the geometry through a series of weighted basis functions, whose spatial definition can be actively managed by changing the position of a low number of parameters, usually called control points. B-Splines, NURBS (Non-uniform rational B-Splines) or Chebyshev Polynomials are examples of parametric curves. In the present research work NURBS parameterization technique is accounted for, hence a full description of such type of curves is given in the following.

In a sensitivity-based design process the shape of the geometry is updated according to the gradient of the functional J, expressed, in a general form, by the relation (2.10). This relation assumes a different expression depending on the chosen shape representation. In the present work a CAD-free and a NURBS-based methodologies are considered, therefore the two different gradient formulations are extensively discussed.

As mentioned, the first idea is to select, as geometrical design variables $\boldsymbol{\alpha}_{geo}$, the grid points bounding the shape to be optimized, such as airfoils or end-walls, without the introduction of any explicit parametrization of the geometry. In the sequel, the vector collecting these mesh points, called hereinafter *active* variables, will be synthetically expressed as $\mathbf{X}_b$.

2.3.1 Design Chain using Grid Equations

In CFD-based simulation the relation between the vectors $\mathbf{X}$ and $\mathbf{X}_b$, herein representing the set of design variables ($\mathbf{X}_b = \boldsymbol{\alpha}_{geo}$), is implicitly available in the mesh generation solver, then it is usually called grid equation. The most straightforward way to exactly compute the geometric sensitivities $(\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\boldsymbol{\alpha}_{geo}})^T = (\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\mathbf{X}_b})^T$ of equation (2.10) is therefore through the differentiation

of the grid equations. However, this operation presents a number of relevant drawbacks. First of all this choice limits the possibility of adopting commercial tools, as the source code is commonly not accessible. Furthermore, fully automatic grid generators based on complex methods, such as advancing Delaunay-front techniques, may result too demanding to differentiate. Hence, this strategy is recommended only for geometries meshed through analytical or elliptic grid generators. Once the grid sensitivities are given, they can be straightforwardly substituted into (2.10) to obtain the *active* gradient expression, which can be written as

$$\left(\frac{\mathrm{d}J}{\mathrm{d}\mathbf{X}_b}\right)^T = \left(\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\mathbf{X}_b}\right)^T \left[\left(\frac{\partial J}{\partial \mathbf{X}}\right)^T + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^T \mathbf{v}\right]. \tag{2.13}$$

A powerful alternative consists in the use of a mesh deformation tool to properly adjust the computational grid according to the movement of the body. In this case, the prior gradient formulation (2.13), has to be slightly modified as suggested afterwards.

2.3.2 Design Chain using Mesh Deformation

During the design process the set of parameters $\mathbf{X}_b$ is iteratively altered according to a given search direction (i.e. the gradient) and the resulting displacement of the geometry profile affects in turn the topology of the mesh. Therefore, an additional information, accounting for the rate of grid deformation, is required to definitely establish a relation between the computational grid and the active variables. By expressing the mesh deformation equation as $\mathbf{X} = \mathbf{X}(\mathbf{X}_{b_{old}}, \Delta\mathbf{X}_b)$, whereby $\Delta\mathbf{X}_b = \mathbf{X}_b - \mathbf{X}_{b_{old}}$ indicates the deformation of the shape to be optimized between two consecutive design steps, the relation between $\mathbf{X}$ and $\mathbf{X}_b$ finally holds

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\boldsymbol{\alpha}_{geo}} = \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\mathbf{X}_b} = \frac{\partial \mathbf{X}}{\partial \Delta\mathbf{X}_b}\frac{\mathrm{d}\Delta\mathbf{X}_b}{\mathrm{d}\mathbf{X}_b} = \frac{\partial \mathbf{X}}{\partial \Delta\mathbf{X}_b}, \tag{2.14}$$

assuming $\frac{\mathrm{d}\Delta\mathbf{X}_b}{\mathrm{d}\mathbf{X}_b} = \mathbf{I}$, being $\mathbf{I}$ the identity matrix. By plugging (2.14) into (2.10) the resulting active gradient can be written as

$$\left(\frac{\mathrm{d}J}{\mathrm{d}\mathbf{X}_b}\right)^T = \left(\frac{\partial \mathbf{X}}{\partial \Delta\mathbf{X}_b}\right)^T \left[\left(\frac{\partial J}{\partial \mathbf{X}}\right)^T + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^T \mathbf{v}\right]. \tag{2.15}$$

The term in brackets of equation (2.15) and (2.13) indicates the gradient of the fitness function with respect to the entire set of grid points $\mathbf{X}$, namely when $\mathbf{X} = \boldsymbol{\alpha}_{geo}$, hereinafter called *aerodynamic* gradient. This term is a function of the flow and adjoint solutions and is normally computed as post-processing once the adjoint equations are converged.

2.3.3 Design Chain using Parametric Representations

When a parameterization of the geometry is specified, a further sensitivity derivative appears on the left side of equations (2.13) and (2.15). As a matter of fact, the vector of the active variables $\mathbf{X}_b$ becomes a function of the parameters of the curves. By indicating with $\boldsymbol{\gamma}$ the whole set of curves parameters, which now become the design variables of the problem ($\boldsymbol{\gamma} = \boldsymbol{\alpha}_{geo}$), the relation between $\mathbf{X}_b$ ($\Delta\mathbf{X}_b$) and $\boldsymbol{\gamma}$ simply holds $\mathbf{X}_b = \mathbf{X}_b(\boldsymbol{\gamma})$, or, more precisely, $\mathbf{X}_b = \Delta\mathbf{X}_b(\boldsymbol{\gamma})$. Therefore, the use of chain rule still enables to recover the final gradient formulation, which yields

$$\left(\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\gamma}}\right)^T = \left(\frac{\mathrm{d}\mathbf{X}_b}{\mathrm{d}\boldsymbol{\gamma}}\right)^T \left(\frac{\mathrm{d}J}{\mathrm{d}\mathbf{X}_b}\right)^T = \begin{cases} \left(\frac{\mathrm{d}\mathbf{X}_b}{\mathrm{d}\boldsymbol{\gamma}}\right)^T \left(\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\mathbf{X}_b}\right)^T \left[\left(\frac{\partial J}{\partial \mathbf{X}}\right)^T + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^T \mathbf{v}\right]. \\ \\ \left(\frac{\mathrm{d}\mathbf{X}_b}{\mathrm{d}\boldsymbol{\gamma}}\right)^T \left(\frac{\partial \mathbf{X}}{\partial \Delta\mathbf{X}_b}\right)^T \left[\left(\frac{\partial J}{\partial \mathbf{X}}\right)^T + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^T \mathbf{v}\right], \end{cases} \tag{2.16}$$

where the upper expression (2.16) is assembled using the grid equations approach introduced in 2.3.1, while the lower one refers to the mesh deformation approach reported in 2.3.2. Notice that both the bottom equation (2.16) and (2.15) need the specification of a closure equation to properly account for the rate of grid deformation during the evolution of the design process. Several techniques are available to properly compute the effect of grid deformation on the gradient, see e.g. [43]. The two methods developed in this work are reported in section 2.3.5. Conversely, an effective methodology to obtain the term $(\frac{\mathrm{d}\mathbf{X}_b}{\mathrm{d}\boldsymbol{\gamma}})$, identifying the sensitivities of the *active* points with respect to the parameters of the (parametric) curves, is briefly discussed in the next section.

2.3.4 Surface Representation

Non-Uniform Rational B-Spline (NURBS) curves are probably the most popular curves and surfaces in computer graphics and nowadays they are the standard for curve and surface description in computer aided design. NURBS curves provide a single precise mathematical form capable of representing common analytical shapes such as lines, planes, conics, and free-form curves and surfaces.

A generic NURBS curve $\mathbf{N}(u)$ of degree p takes the form

$$\mathbf{N}(u) = \sum_{i=0}^{n} R_{i,p}(u)\mathbf{P}_i, \tag{2.17}$$

where $\mathbf{P}_i$ is one of the $n+1$ control points where $n \geq p$. As for the B-Spline curves, the parameter u varies inside the knot sequence $U = [u_0, u_m]$. The

value u_0 and u_m are the first and the last elements of a strictly increasing sequence of $m+1$ elements, so-called *knots*.

The piecewise rational functions $R_{i,p}(u)$ are called *rational* basis functions since are expressed as

$$R_{i,p}(u) = \frac{\omega_i N_{i,p}(u)}{\sum_{i=0}^{n} \omega_i N_{i,p}(u)}, \tag{2.18}$$

where, $N_{i,p}$ are the B-Spline basis function recursively defined as

$$N_{i,p}(u)\frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u), \tag{2.19}$$

and ω_i are the weights of the control points.

NURBS basis functions and curves are a generalization of non-rational B-spline basis functions and curves. Particularly, if all the weights ω_i in equation (2.18) are the same, the basis functions $R_{i,p}(u)$ reduce to the B-Spline basis functions $N_{i,p}(u)$, see equation (2.19). In this case, the NURBS curve is equivalent to a B-Spline with the same control polygon.

NURBS curves inherit all the properties characterizing B-spline curves. Moreover, changing the weight ω_i only affects the portion of the curve controlled by the local basis function $R_{i,p}(u)$. Hence, control point movement or weight modification can both be used to achieve local shape control. Qualitatively, if ω_i increase (decreases), the curve moves closer to (farther from) P_i. Furthermore, the movement of the curve for a fixed parameter value $\bar{u}$ is along a straight line in direction of P_i.

Due to their rational basis functions, second degree NURBS curves exactly represent conics, i.e. parabolas, ellipses and hyperboles. A very exhaustive description of the NURBS curves/surfaces can be found in [44] and [45].

The vector of coordinates $\mathbf{X}_b = [\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b]$ of the set of (active) boundary nodes in the cartesian space is therefore a function of $\mathbf{N}(u)$. The underlying algebraic relation can be concisely summarized as

$$\mathbf{X}_b = \mathbf{X}_b(\mathbf{N}(u)) = \mathbf{X}_b(\mathbf{u}, \omega, \mathbf{P}). \tag{2.20}$$

By differentiating equation (2.20) against $\mathbf{u}, \omega, \mathbf{P}$ the three parametric sensitivities $\frac{\mathrm{d}\mathbf{X}_b}{\mathrm{d}\boldsymbol{u}}$, $\frac{\mathrm{d}\mathbf{X}_b}{\mathrm{d}\boldsymbol{\omega}}$, and $\frac{\mathrm{d}\mathbf{X}_b}{\mathrm{d}\mathbf{P}}$ are obtained. This work only accounts for the sensitivities of the functional with respect to the control points, thus equation (2.16) can be rewritten substituting $\boldsymbol{\gamma} = \mathbf{P}$, achieving $(\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\alpha}_{geo}})^T = (\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\gamma}})^T = (\frac{\mathrm{d}J}{\mathrm{d}\mathbf{P}})^T$.

2.3.5 Mesh Deformation

Grid deformation algorithms are still an interesting area of research in the field of computational geometry. Several methodologies are currently avail-

able to accurately deform the mesh for an assigned boundary displacement. The most used strategy is based on the spring analogy [46], although the method suffers from severe limitations in case of significant mesh deformations. More innovative approaches, relying on the use of radial basis functions, have been successfully applied by [47] to unstructured grids, proving a high robustness also in case of relevant displacements.

In the present thesis two different alternatives are considered. The first method, taken into account at an early stage of the research, neglects the deformation of the grid over the whole computational domain except for the moving boundary. In other words, at each design step, the domain is enforced to be freeze, thus the contribution of the grid deformation rate practically vanishes.

Apart from marginal applications involving tiny displacements, this method, often named frozen domain, is generally very poor to approximate the rate of grid deformation in optimization problems. However, its choice can be regarded as an effective solution to simplify the gradient equation (2.16). As shown in [48], the method is suitable for inviscid flows. In a differential form the frozen domain approach can therefore be written as

$$\frac{\partial \mathbf{X}_k}{\partial \Delta \mathbf{X}_b} = \frac{\partial (\mathbf{X}_{k_{old}} + \Delta \mathbf{X}_k)}{\partial \Delta \mathbf{X}_b} = \frac{\partial \Delta \mathbf{X}_k}{\partial \Delta \mathbf{X}_b} = \begin{cases} 1 & k = b \\ 0 & k \neq b. \end{cases} \tag{2.21}$$

Even though a relevant simplification is introduced in the gradient formulation, the use of such a strategy frequently leads to satisfactory results in terms of reliability of the overall design procedure. In fact, notice that optimization methods used for large scale optimization problems, as the preconditioned steepest descent method, do not require an accurate estimate of the gradient, see [41], as it is the case of more sophisticated algorithms such as quasi-newton BFGS, and hence can tolerate the uncertainties introduced by such an approach. As a consequence, the frozen domain methodology may be regarded as a reliable compromise for optimizations processes involving boundary grid points as design variables. This issue is anyhow investigated in the present thesis by employing the frozen domain approach in optimization processes involving inviscid flows.

The second methodology accounts for the contribution of the interior nodes in the rate of grid deformation. As already mentioned, several types of deformation laws are available to effectively predict the rate of grid deformation for a displacement of an active boundary surface. These methods are generally expressed in algebraic form, therefore, once coded, are rather easy to differentiate in order to estimate the term $(\frac{\partial \mathbf{X}}{\partial \Delta \mathbf{X}_b})^T$.

The approach chosen in the present thesis follows the seminal work of [47] to achieve a highly flexible and robust deformation tool for unstructured

grids, mainly based on the use of radial basis functions. Radial basis functions (RBF) have become a well-established tool for interpolating scattered data and they are typically adopted in fluid-structure interaction to transfer information over the discrete fluid-structure interface. Radial basis functions are interpolating functions approximating the displacement of a single mesh point through a weighted sum of basis functions in the form

$$s(\mathbf{X}) = \sum_{j=1}^{N_{X_b}} w_j \phi(\|\mathbf{X} - \mathbf{X}_{b_j}\|) + p(\mathbf{X}_b) \tag{2.22}$$

where $\mathbf{X}_{b_j} = [x_{b_j}, y_{b_j}, z_{b_j}]$ are the boundary points where the displacements are known, p an additional polynomial, N_{X_b} the total number of boundary points, and ϕ a given basis function with respect to the Euclidean distance $\|\mathbf{X} - \mathbf{X}_{b_j}\|$. s is the displacement of any mesh node X_i. The weights w_j and the polynomial p are determined by the interpolating conditions

$$s(\mathbf{X}_b) = d_{b_j}, \tag{2.23}$$

where d_{b_j} is the displacement at the X_{b_j} boundary point. Additional conditions are required to compute the coefficients of the polynomial $p(\mathbf{X}_b)$ and they can be written as

$$\sum_{j=1}^{N_{X_b}} w_j p(\mathbf{X}_{b_j}) = 0. \tag{2.24}$$

The minimal degree of the polynomial depends on the choice of the basis function. Generally, if the basis functions are conditionally positive definite of order $m \leq 2$, a linear polynomial can be used. The values of the coefficients $\mathbf{w}$ and the coefficients $\boldsymbol{\beta}$ of the polynomial are obtained by solving the linear system

$$\begin{bmatrix} \mathbf{d}_{b_j} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{b,b} & \mathbf{P}_b \\ \mathbf{P}_b^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\beta} \end{bmatrix}, \tag{2.25}$$

where $\mathbf{M}_{b,b}$ indicates a square matrix (with N_{X_b} rows and columns) containing the evaluation of the basis function $\phi_{i,j} = \phi(\|\mathbf{X}_{b_i} - \mathbf{X}_{b_j}\|)$, $\mathbf{P}_b$ a $N_{X_b} \times 4$ matrix with a single row j given by $[1 \quad x_{b_j} \quad y_{b_j} \quad z_{b_j}]$. The spatial displacement of an interior grid node is obtained by evaluating the interpolation function (2.22) at the internal grid point, namely imposing that

$$d_j = s(\mathbf{X}_j). \tag{2.26}$$

The most relevant features of the RBF-based mesh deformation are the ability of individually moving each node of the mesh and the possibility of

Name	**Function**
Quadric biharmonics	$1 + \|\|x\|\|^2$
Inverse quadric biharmonics	$\frac{1}{1+\|\|x\|\|^2}$
Thin plate spline	$\|\|x\|\|^2 log(\|\|x\|\|)$
Gaussian	$e^{-\|\|x\|\|^2}$
Volume spline	$\|\|x\|\|$
Wendland C^0	$(1 - \|\|x\|\|)^2$

Table 2.1. *Some radial basis functions with global support implemented in dMesh.*

interpolating the displacements separately for each spatial direction. As a consequence, no mesh connectivity information is required, thus making the method very suitable for both structured and unstructured meshes.

On the basis of the theoretical background explained so far, a novel mesh deformation tool, named *dMesh*, for unstructured grid has been developed within the present research work. The tool embeds a series of basis function, reported in Table 2.1 and adopts a fast direct method from the *MKL* library for the resolution of the linear system (2.25).

The quality and the robustness of the method have been assessed by deforming some existing meshes according to a prescribed movement of the solid walls of the domain. The movement of the wall can be assigned in terms of rigid rotation, rigid translation or scaling operation. Two different test cases, both referred to structured meshes, are herein reported to appreciate the capability of the method into distorting both 2D and 3D computational grids. The first test consists in a movement of a 2D viscous structured mesh due to a severe rigid rotation of 20 degree (clockwise) of the blade profile around the leading edge. The second example, on the contrary, considers a mesh deformation caused by a scaling of the 3D turbine blade geometry. The coordinates of the points are in fact scaled by a factor 0.5 in x and y direction. A Wendland's C^0 basis function is applied to both tests, but similar results have been obtained using the volume splines. Figures 2.1 and 2.2 show the initial and the deformed meshes. A reasonable quality of the final meshes is preserved, even in presence of significant distortions of the original blade wall. In both cases, the total computational burden necessary to deform the grids is very limited and anyhow much lower than that necessary to generate a new grid. However, the cost associated to mesh construction strongly depends on the type of mesh generator and the expense is generally much higher for elliptic solvers, as the one applied in this study.

It should be finally noted that in shape optimization the spatial displacement of the active surface is very limited at each design-step, hence the

inflow/outflow boundaries may be approximately considered as fixed walls throughout the design process.

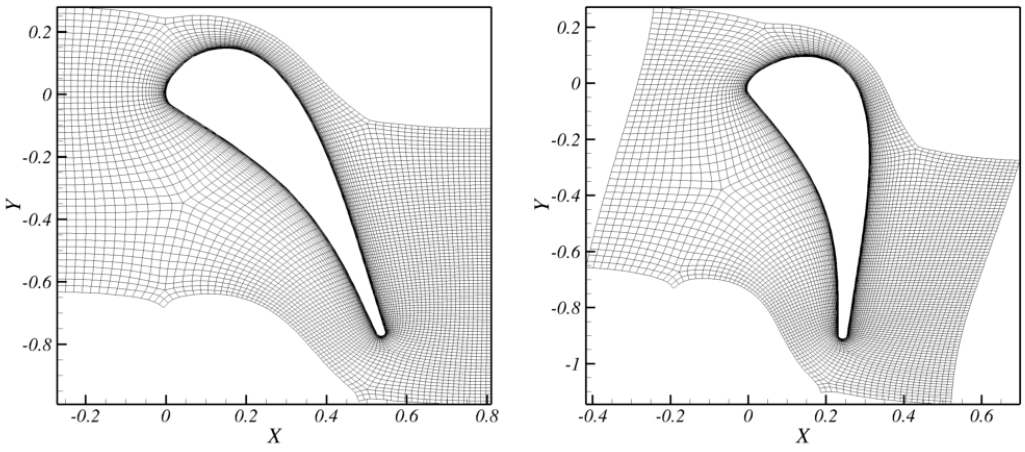


Figure 2.1. *Original (left) and deformed (right) meshes after a rigid rotation.*

The source code of the grid deformation tool has been finally automatically differentiated through AD in reverse mode to compute the term $(\frac{\partial \mathbf{X}}{\partial \Delta \mathbf{X}_b})^T$ necessary for gradient definition. As the spatial displacements are simply expressed by means of algebraic relations, see (2.22), the derivation of the RBF-based solver is a relatively manageable task.

Starting from the *aerodynamic* gradient both frozen and radial basis function approaches ensure the construction of a low dimension sensitivity vector (active gradient), that can be either directly utilized in customary CAD-free optimization loops or further processed to achieve surface gradients.

2.3.6 Implementation of the Design Chain using AD

Up to now the method to obtain a design chain for shape optimization has been presented and the strategies to compute each term appearing in the surface gradient equation have been discussed. However, to clearly understand how exploiting AD for assembling the surface gradient (2.16), the step-by-step procedure can be concisely summarized as follows:

1. The first step of the design chain provides the resolution of the adjoint equation. AD is exploited to immediately obtain the vector $\left(\frac{\partial J}{\partial \mathbf{u}}\right)^T$ and the Jacobian $\left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right)^T \mathbf{v}$. The routines implementing the functional J (the objective function of the shape optimization problem) and the vector

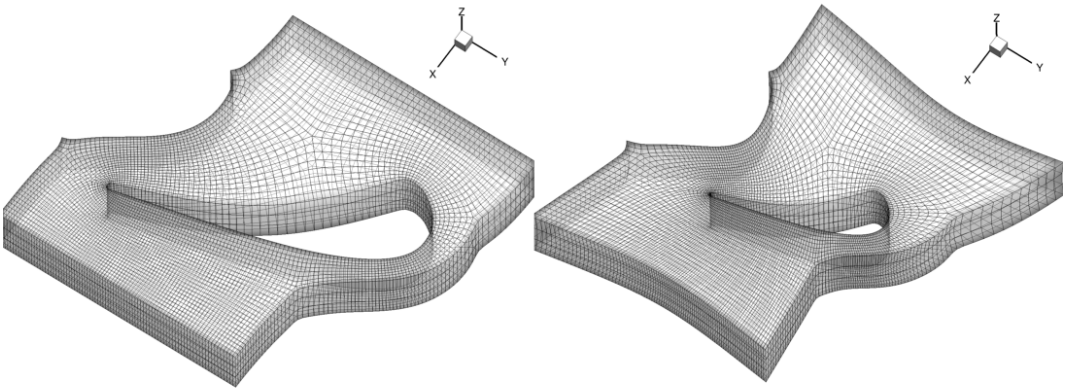


Figure 2.2. *Original (left) and deformed (right) meshes after a scaling in x/y direction.*

R (the numerical fluxes of the flow solver) are then backwardly differentiated taking u (the states) as independent variables. As a result, using the AD formalism of section 2.2, the following two expressions can be written:

$$\begin{aligned} \mathbf{u}_b &= \left(\frac{\partial J}{\partial \mathbf{u}}\right)^T \rightarrow \mathbf{f} = J, \mathbf{f}_b = \mathbf{1} \\ \mathbf{u}_b &= \left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right)^T \mathbf{v} \rightarrow \mathbf{f} = \mathbf{R}, \mathbf{f}_b = \mathbf{v}. \end{aligned} \tag{2.27}$$

Notice that second equation includes also the linearization of the boundary conditions and the nature of the thermo-physical behavior of the fluid, which also holds in the first relation. This can be clearly pointed out by restricting the analysis to conservative flow solvers. In these tools the state variables are represented by the conservative variables, i.e. $\rho, \rho u, \rho v, \rho e_t$ for 2D problems. Since, in the general case of compressible viscous flows, the flux function and the residual vector $\mathbf{R}$ depend on pressure $\mathbf{p}$ and temperature $\mathbf{T}$ and the Jacobian matrix is the derivative of the flux functions with respect to the conservative variables $\mathbf{u}$, then $\mathbf{R} = \mathbf{R}(\mathbf{u}, \mathbf{p}(\mathbf{u}), \mathbf{T}(\mathbf{u}))$. By restricting the analysis to inviscid flow, which is the flow model considered in the present thesis, the latter relation simplifies in $\mathbf{R} = \mathbf{R}(\mathbf{u}, \mathbf{p}(\mathbf{u}))$. The transposed matrix $(\frac{\partial \mathbf{R}}{\partial \mathbf{u}})^T$ appearing in the adjoint equations can then be conveniently

expressed as

$$\left(\frac{\mathrm{d}\mathbf{R}}{\mathrm{d}\mathbf{u}}\right)^T = \left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right)^T + \left(\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}\mathbf{u}}\right)^T \left(\frac{\partial \mathbf{R}}{\partial \mathbf{p}}\right)^T \tag{2.28}$$

The term $(\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}\mathbf{u}})^T$ implies the linearization of the thermodynamic model, which will be discussed more precisely in chapter 4.

2. The second step of the design chain involves the calculation of the gradient with respect the whole set of grid points, namely the *aerodynamic gradient* $[(\frac{\partial J}{\partial \mathbf{X}})^T + (\frac{\partial \mathbf{R}}{\partial \mathbf{X}})^T \mathbf{v}]$ of (2.15). This operation can be straightforwardly carried out by separately differentiating the numerical procedures implementing $J(\mathbf{X})$ and $\mathbf{R}(\mathbf{X})$, using the reverse mode of AD. By adopting the same notation used in section 2.2 and assuming as the vector of parameters $\boldsymbol{\alpha}$ the vector $\mathbf{X}$, the resulting relations can be written as

$$\begin{aligned} \boldsymbol{\alpha}_b &= \left(\frac{\partial J}{\partial \mathbf{X}}\right)^T \rightarrow \mathbf{f} = J, \boldsymbol{\alpha} = \mathbf{X}, \mathbf{f}_b = \mathbf{1} \\ \boldsymbol{\alpha}_b &= \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^T \mathbf{v} \rightarrow \mathbf{f} = \mathbf{R}, \boldsymbol{\alpha} = \mathbf{X}, \mathbf{f}_b = \mathbf{v}. \end{aligned} \tag{2.29}$$

3. The following step of the procedure involve the calculation of the *active gradient*. The aerodynamic gradient is then multiplied by the transposed grid deformation vector, achieved by reversely processing the mesh deformation algorithm through AD. By remembering the grid deformation equation and using AD notation, the following relation holds

$$\boldsymbol{\alpha}_b = \left(\frac{\mathrm{d}\mathbf{J}}{\mathrm{d}\mathbf{X}_b}\right)^T \rightarrow \mathbf{f} = \mathbf{X}, \boldsymbol{\alpha} = \Delta\mathbf{X}_b, \mathbf{f}_b = \left[\left(\frac{\partial J}{\partial \mathbf{X}}\right)^T + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^T \mathbf{v}\right]. \tag{2.30}$$

In case of the use of a differentiated version of the grid generator simply yields $\boldsymbol{\alpha} = \mathbf{X}_b$. It should be pointed out that the vector $\Delta\mathbf{X}_b$ may be quickly obtained by computing, (active) node by (active) node, the displacement between the current shape profile and that of the preceding configuration. For parameterized geometries, such a displacement is a function of the curves themselves, namely $\Delta\mathbf{X}_b = \Delta\mathbf{X}_b(\boldsymbol{P})$.

4. The last element of the design chain is the calculation of the sensitivities of the fitness function with respect to NURBS control points. The *surface* gradient is therefore evaluated in a similar fashion, namely

$$\boldsymbol{\alpha}_b = \left(\frac{\mathrm{d}\mathbf{J}}{\mathrm{d}\mathbf{P}}\right)^T \rightarrow \mathbf{f} = \mathbf{X}_b, \boldsymbol{\alpha} = \mathbf{P}, \mathbf{f}_b = \left(\frac{\mathrm{d}\mathbf{J}}{\mathrm{d}\mathbf{X}_b}\right)^T. \tag{2.31}$$

The *active/surface* gradient can be finally used in any design optimization approach, see chapter 3, to perform a sensitivity analysis or as the basis for some uncertainty quantification algorithm, see chapter 6.

Chapter 3

Optimization Methodology

This chapter illustrates the main elements of the optimization chain, providing a comprehensive description of the numerical solvers utilized in the design process. An overview of the optimization algorithm developed within this research work is also presented.

3.1 Numerical Solvers

3.1.1 Flow Solver

The method used to construct a discrete form of the Euler equations for general unstructured and hybrid grids is now described. The discretization scheme, following the ideas introduced in [49] and subsequently extended in [50] can be regarded as an hybrid finite-volume/finite-element (FEFV) scheme. The description of the hybrid discretization approach as implemented in the zFlow solver is outlined in the following. The direct application of the physical principles (mass, energy and momentum balances) to a finite amount of matter, assuming inviscid flows, leads to the Euler equations, that read:

$$\frac{d}{dt}\int_{\Omega}\mathbf{u}(\mathbf{x},t)\,d\Omega+\oint_{\partial\Omega}\mathbf{n}\mathbf{F}(\mathbf{u}(\mathbf{x},t))\,d\sigma=0 \tag{3.1}$$

An alternative formulation of the integral Euler equations is the so-called weak or variational form. This formulation is particularly convenient as a starting point to construct a discretized version of the equations (3.1) by means of both finite volume and finite element methods. Multiplying the equations (3.1) for a given test function ϕ belonging to a suitable subspace V and integrating by parts the second domain integral, the weak form is obtained:

$$\frac{d}{dt}\int_{\Omega}\phi\mathbf{u}\,d\Omega+\oint_{\partial\Omega}\phi\mathbf{n}\mathbf{F}(\mathbf{u})\,d\sigma-\int_{\Omega}\nabla\phi\mathbf{F}(\mathbf{u})\,d\Omega=0 \tag{3.2}$$

The first step needed for the construction of a discrete version of the weak formulation (3.2) is the subdivision of the domain Ω into a collection of non-overlapping elements E, i.e. the generation of a grid τ_h. The domain associated to element E is denoted by Ω_e. For every vertex i (hereinafter called node) of the grid τ_h, Ω_i denotes the union of the domains Ω_e of the elements E which contain the node i, $\partial\Omega_i$ denotes the boundary of Ω_i, and Γ_i the set $\partial\Omega_i \cap \partial\Omega$. The set of all the nodes of τ_h is denoted by K, that of the boundary nodes by K^{∂} and that of the nodes inside or on the boundary of Ω_i by K_i. The set K_i with the exception of node i is instead denoted by $K_{i\neq}$, and $K_i^{\partial} = K_i \cap K^{\partial}$.

The discrete version of (3.2) is obtained by searching the approximate solution among functions $\mathbf{u}_h \in V_h$ and test functions $\phi_h \in V_h$ (Galerkin method), where V_h denotes the finite dimensional function space of linear or bilinear functions ϕ_h, namely functions which are globally continuous and linear or bilinear inside the elements E of τ_h. Both the unknown functions $\mathbf{u}$ and the test functions ϕ appearing in (3.2) are therefore replaced by the

expansions:

$$\begin{aligned} \mathbf{u} \leftarrow \mathbf{u}_h(\mathbf{x}, t) &= \sum_{k \in K} N_k(\mathbf{x}) \mathbf{u}_k(t) \\ \phi \leftarrow \phi_h(\mathbf{x}) &= \sum_{k \in K} N_k(x) \phi_k \end{aligned} \tag{3.3}$$

thus leading, for a generic node i of τ_h, to the semi-discrete equations:

$$\sum_{k \in K_i} \left[\int_{\Omega_i} N_i N_k \, d\Omega \right] \frac{d\mathbf{u}_k}{dt} + \int_{\Gamma_i} \mathbf{n}\mathbf{F}(\mathbf{u}_h) \, d\sigma - \int_{\Omega_i} \nabla N_i \mathbf{F}(\mathbf{u}_h) \, d\sigma = 0 \tag{3.4}$$

The computations of the integrals of the discrete weak formulation (3.4) is usually performed by cycling over all the elements of τ_h and assembling all the contributions. However, following the ideas originally introduced in [49] and successively generalized in [50] the computation can be rearranged in a more convenient form, namely as a summation over "pairs" of interacting nodes, called node-pairs. Two nodes i and k interact if the associated regions Ω_i, Ω_k have a nonempty intersection, i.e. if $\Omega_{ik} = \Omega_i \cap \Omega_k \neq 0$. By omitting for brevity the derivation, which is entirely reported in [51], equation (3.4) can be expressed in the discrete form as

$$D_i \frac{d\mathbf{u}_i}{dt} + \sum_{k \in K_{i,\neq}} \boldsymbol{\eta}_{ik} \frac{\mathbf{F}_i + \mathbf{F}_k}{2} + \boldsymbol{\xi}_i^{\partial} \mathbf{F}_i = 0. \tag{3.5}$$

where the metric coefficients, which are a function of the grid points $\mathbf{X}$ only, are defined as

$$\begin{aligned} D_i(\mathbf{X}) &= \sum_{k \in K_i} \int_{\Omega_i} N_i N_k \, d\Omega \\ \boldsymbol{\eta}_{ik}(\mathbf{X}) &= \int_{\Omega_i k} (N_i \nabla N_k - N_k \nabla N_i) \, d\Omega \\ \boldsymbol{\xi}_i^{\partial}(\mathbf{X}) &= \int_{\Gamma_i} N_i \mathbf{n} \, d\sigma. \end{aligned} \tag{3.6}$$

The metric vectors introduced above are characterized by the remarkable properties:

$$\begin{aligned} \boldsymbol{\eta}_{ik} &= -\boldsymbol{\eta}_{ki} \\ \sum_{k \in K_{i,\neq}} \boldsymbol{\eta}_{ik} + \boldsymbol{\xi}_i^{\partial} &= 0. \end{aligned} \tag{3.7}$$

The above relations guarantee that the metric coefficients behave similarly to the integrated normal vectors of suitably defined control volumes in the frame of a finite volume method. In other words, a special class of finite volume methods is obtained by exploiting the metric vectors $\boldsymbol{\eta}_{ik}$ and $\boldsymbol{\xi}_i^{\partial}$ in order to define the finite volume cells, without requiring an explicit definition of the actual cell geometry (although, in practice, such information are known *a-priori*).
The simple average flux $\boldsymbol{\eta}_{ik}(\mathbf{F}_i + \mathbf{F}_k)/2$ in (3.5) approximates a centered scheme that, in absence of any stabilization terms, is not suited to the computation of hyperbolic problems. In order to provide a stabilization of the numerical scheme the centered formulation is replaced by a high-resolution numerical flux function usually employed in the finite volume method, thus leading to the so-called FE/FV scheme:

$$D_i \frac{d\mathbf{u}_i}{dt} + \sum_{k \in K_{i,\neq}} \hat{\mathbf{f}}(\mathbf{u}_i, \mathbf{u}_k, \mathbf{g}_i^-, \mathbf{g}_k^+, \boldsymbol{\eta}_{ik}) + \boldsymbol{\xi}_i^{\partial} \mathbf{F}_i = 0 \tag{3.8}$$

where $\hat{\mathbf{f}}$ denotes the numerical flux function in the direction $\boldsymbol{\eta}_{ik}$. The gradients $\mathbf{g}_i^-, \mathbf{g}_k^+$ are used to evaluate the limiters required to achieve a non oscillatory high-resolution (TVD) scheme. In this thesis the numerical flux function is due to Roe-Davis [52] with the vanAlbada flux limiter generalized to the case of an arbitrary EoS following the Vinokur-Montagné approach [53]. If we denote by $\mathbf{u}$ the global vector of the unknown solution, by $\mathbf{D}$ the lumped (or diagonalized) mass matrix, and by $\mathbf{R}(\mathbf{u})$ the global residual vector, system of equation (3.8) can be written in a compact form yielding

$$\mathbf{D} \frac{d\mathbf{u}}{dt} + \mathbf{R}(\mathbf{u}) = 0. \tag{3.9}$$

The equations are integrated in time by an implicit (Backward Euler) method; implicit schemes represent an effective solution in the context of optimization, in which the computational savings provided by implicit techniques drastically reduce the overall computational expense of the optimization process. The linear system arising at each time step is approximatively solved by means of a preconditioned GMRES method ([54]), with the incomplete LU factorization technique ILU0 (no fill-in allowed) as preconditioning solver.

The version of the solver used in the present thesis integrates a rich thermodynamic library for the calculation of properties of pure fluids and mixtures. The thermodynamic behavior of the fluid can be modeled by EoS of different complexity, comprising the polytropic ideal gas (PIG), the polytropic Van der Waals model (PVdW), cubic EoS and very accurate multiparameter equations of state. Cubic and multi-parameter equations are accessible through the *FluidProp* database or by a novel look-up table (LUT)

approach specifically developed within the present thesis. A comprehensive description of the LUT method is given in chapter 4.

3.1.2 Adjoint Solver

Basically the adjoint solver implements a linearization of the non-linear steady term $\mathbf{R}(\mathbf{u})$ of eq. (3.9), thus inheriting the numerical features of the native flow solver. The adjoint problem requires the calculation of a linear algebraic system in the form $\mathbf{A}\mathbf{v} = \mathbf{b}$, where $\mathbf{A} = (\partial\mathbf{R}/\partial\mathbf{u})^T$, $\mathbf{b} = -(\partial J/\partial\mathbf{u})^T$. In order to solve the adjoint system iterative matrix-free techniques are preferred, as they do not require the explicit calculation of the Jacobian matrix $\partial\mathbf{R}/\partial\mathbf{u}$, but they only need to evaluate the product $(\partial\mathbf{R}/\partial\mathbf{u})^T\mathbf{v}$ at each iteration. In particular, in the present research, the iterative Flexible Generalized Minimal Residual Method (FGMRES), see [54], is adopted. The adjoint system of equations (6.11) is then solved by constructing the vectors $(\partial\mathbf{R}/\partial\mathbf{u})^T\mathbf{v}$ and $(\partial J/\partial\mathbf{u})^T$ by exploiting AD in reverse mode, see [42]. In particular the vector $(\partial\mathbf{R}/\partial\mathbf{u})^T\mathbf{v}$ is assembled *on-the-fly* during the iterations of the solver. In this way the high-resolution of the numerical method is intrinsically preserved.

The efficiency of the FGMRES is further enhanced by preconditioning the linear system through the ILU0 (no fill-in allowed) or ILUT (fill-in allowed by fixing a threshold value) incomplete factorization techniques applied to the first-order Jacobian matrix. A typical convergence rate of the adjoint solver for an inviscid flow case using either ILU0 or ILUT preconditioner (with a threshold value of the order 10^{-3}) is provided in Figure 3.1.
In general, the higher is the accuracy of the preconditioner, the lower is the number of iterations required to achieve convergence, but the greater is also the computational cost to construct the preconditioning matrix. As a result, ILU0 preconditioner enables to converge more rapidly compared to ILUT technique, despite the lower number of iterations required by this latter. In both cases, the algorithm proved to robustly drive the solution of the sparse linear system to convergence, the convergence rate being approximatively an order of magnitude higher than the one of the flow calculation for an inviscid flow problem.

The adjoint solver embeds the same thermodynamic library implemented in the flow solver. However, since equations of state appear in a linearized form within the adjoint solver, as outlined in the final section of chapter 2, external calls via *FluidProp* are not suited for automatic differentiation, therefore only PIG, PVdW and LUT approaches are taken into account. The differentiation of a flow solver for real gases is not a standard operation and represents one of the major hardnesses of this work. In fact, the exact (reverse) differentiation of the residual vector based on a Vinokur-Montagné flux

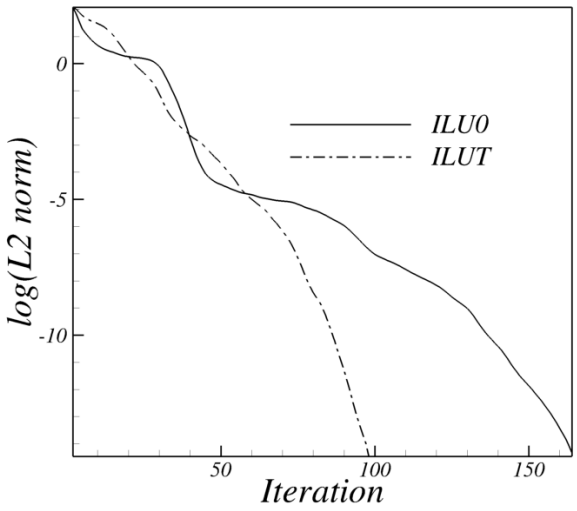


Figure 3.1. *Convergence rates of the adjoint solver using ILU0 and ILUT preconditioners.*

splitting scheme, described in the previous section, requires the establishment of effective differentiation rules for arbitrary equations of state. More details concerning this generalized differentiation treatment, in the framework of the real-gas adjoint solver developed within the present research, are addressed in the following.

Real-Gas Adjoint Equations

In section 2.3 of chapter 2 the step-by-step procedure to construct the design chain for shape optimization through AD has been deeply discussed. As outlined in the first item of section 2.3, the construction of the adjoint system has been limited to inviscid flows, hence the contribution of temperature is omitted hereinafter.

The chain expansion of the (transposed) Jacobian matrix appearing in the adjoint equations is now further extended for the case of flux functions using approximate Riemann solvers for real gases, as the one proposed in [53] and implemented in zFlow. Hence, more terms are to be adjointed during the differentiation process. The numerical flux function becomes dependent on the conservative state, pressure and derivatives of pressure with respect to density and internal energy, as reported in [55]. Therefore the global residual can be rewritten as

$$\mathbf{R} = \mathbf{R}[\mathbf{u}, \mathbf{p}(\mathbf{u}), \nabla \mathbf{p_e}(\mathbf{u}), \nabla \mathbf{p}_\rho(\mathbf{u})], \tag{3.10}$$

or, more concisely, as

$$\mathbf{R} = \mathbf{R}[\mathbf{q}(\mathbf{u})], \tag{3.11}$$

where $\mathbf{q} = (\mathbf{u}, \mathbf{p}, \nabla\mathbf{p_e}, \nabla\mathbf{p}_\rho)$. As pressure $\mathbf{p}$ and its derivatives $\nabla\mathbf{p_e}, \nabla\mathbf{p}_\rho$ are functions of the conservative state $\mathbf{u}$, the following relation for the Jacobian holds

$$\left(\frac{\mathrm{d}\mathbf{R}}{\mathrm{d}\mathbf{u}}\right)^T = \left(\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}\mathbf{u}}\right)^T \left(\frac{\partial\mathbf{R}}{\partial\mathbf{q}}\right)^T \tag{3.12}$$

The first matrix of the right hand-side of equation (3.12) involve second-order derivatives of pressure with respect to density and internal energy, that can be automatically computed by means of AD in reverse mode for any type of equation of state. This operation is herein carried out for both PIG and PVdW models, as well as for the equations of state constructed by means of the look-up table approach presented in chapter 4.

Alternative Methods for Adjoint System Resolution

The effectiveness of ILU-type preconditioning may quickly decays with problem size and stiffness, most particularly for viscous or three dimensional problems, requiring an increase of the fill-in level to impractical values. These hardnesses have been faced by many authors as [23], [56], and [57]. One of the most interesting idea is the so-called implicit dual solution method presented in [56]. The general principle behind this idea is essentially the concept of duality between the flow and adjoint system. Duality can be regarded as a mathematical property for which if an iterative method is effective for the non-linear (flow) problem, then a corresponding variant is suitable for the adjoint problem. The duality concept has been formalized by Giles in [58] and is of paramount importance in the context of linearization of non-linear problems. In practice, any time-marching algorithm successfully used for converging flow equations can be immediately adopted for solving the adjoint equations and similar convergence rates are to be expected. Assuming that the linearization is exact, complete and based on the fully converged non-linear solution, the linear computation ensures the convergence within a fixed number of iterations with the same iterative method (and CFL number) of the flow equations. Implicit time-marching algorithms outperform the efficiency of explicit methods due to the fact that, theoretically, are no subject to any limitations in the CFL number. Hence they are of great importance for the resolution of adjoint linear system when iterative linear solvers tend to stall. A time-dependent formulation of the adjoint system can be written

as

$$\mathbf{V}\left(\frac{\partial \mathbf{v}}{\partial t}\right)+\left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right)^{T}\mathbf{v}+\left(\frac{\partial J}{\partial \mathbf{u}}\right)^{T}=\mathbf{V}\left(\frac{\partial \mathbf{v}}{\partial t}\right)+\mathbf{D}=0, \tag{3.13}$$

$\mathbf{D}$ concisely indicates the original adjoint system and $\mathbf{V}$ ensembles the (mesh) cell volumes. The previous relationship can be marched to steady-state condition using the same algorithms implemented in the flow solver. In case of an explicit Euler method the following expression intuitively holds

$$\mathbf{v}^{n+1}=\mathbf{v}^{n}-\frac{\Delta t}{\mathbf{V}}\mathbf{D}^{n}. \tag{3.14}$$

By linearizing $\mathbf{D}$ between two consecutive time steps the implicit version is obtained. The linearization yields

$$\mathbf{D}^{n+1}=\mathbf{D}^{n}+\frac{\partial \mathbf{D}^{n}}{\partial \mathbf{v}}\Delta \mathbf{v}. \tag{3.15}$$

However, since

$$\frac{\partial \mathbf{D}}{\partial \mathbf{v}}=\frac{\partial[(\frac{\partial \mathbf{R}}{\partial \mathbf{u}})^{T}\mathbf{v}+(\frac{\partial J}{\partial \mathbf{u}})^{T}]}{\partial \mathbf{v}}=\left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right)^{T}, \tag{3.16}$$

after performing some algebraic manipulations, the corresponding implicit counterpart of (3.14) can be written as

$$\begin{aligned}\left[\frac{\mathbf{V}}{\Delta t}\mathbf{I}+\left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\right)^{T}\right]\Delta \mathbf{v}^{n}&=-\mathbf{D}^{n}\\ \mathbf{v}^{n+1}&=\mathbf{v}^{n}+\Delta \mathbf{v}^{n}\end{aligned} \tag{3.17}$$

The above equation represents the dual implicit formulation for the adjoint problem. In the present thesis the explicit forward formula has been used for shape optimization of a quasi-1D problem, extensively discussed in the chapter 5. The model was originally developed as a starting point to build a solid background on the fundamental aspects of adjoint method, including the capabilities of Automatic Differentiation.

3.1.3 Gradient Calculators

In chapter 2 the expressions of the the gradient of the fitness function, depending on both active grid nodes (2.15) and NURBS control points, (2.16) have been found. As already seen, both formulations require the partial derivatives of J and R with respect to the whole set of grid points. In a numerical solver the informations of the grid are explicitly accounted for through the so-called metric quantities, i.e. the normal vectors and

the areas of the grid cells. In the present node-pair numerical technique, the influence of the grid points on the cost function is therefore given by the metric vectors previously introduced, namely $J = J(\boldsymbol{\eta}(\mathbf{X}), \boldsymbol{\xi}^{\partial}(\mathbf{X}))$ and $\mathbf{R} = \mathbf{R}(\boldsymbol{\eta}(\mathbf{X}), \boldsymbol{\xi}^{\partial}(\mathbf{X}))$. The term in brackets of eq. (2.15) can be rewritten using the chain rule as

$$\left(\frac{\partial J}{\partial \mathbf{X}}\right)^T + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^T \mathbf{v} = \left[\left(\frac{\partial J}{\partial(\boldsymbol{\eta}, \boldsymbol{\xi}^{\partial})} \frac{\mathrm{d}(\boldsymbol{\eta}, \boldsymbol{\xi}^{\partial})}{\mathrm{d}\mathbf{X}}\right)^T + \left(\frac{\partial \mathbf{R}}{\partial(\boldsymbol{\eta}, \boldsymbol{\xi}^{\partial})} \frac{\mathrm{d}(\boldsymbol{\eta}, \boldsymbol{\xi}^{\partial})}{\mathrm{d}\mathbf{X}}\right)^T \mathbf{v}\right]. \tag{3.18}$$

If the fitness function is computed on the boundary, as occurs in many applications, J depends only on $\boldsymbol{\xi}^{\partial}$ and the final expression of the (active) gradient reads

$$\left(\frac{\mathrm{d}J}{\mathrm{d}\mathbf{X}_b}\right)^T = \left(\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\mathbf{X}_b}\right)^T \left[\left(\frac{\mathrm{d}\boldsymbol{\xi}^{\partial}}{\mathrm{d}\mathbf{X}}\right)^T \left(\frac{\partial J}{\partial \boldsymbol{\xi}^{\partial}}\right)^T + \left(\frac{\mathrm{d}(\boldsymbol{\eta}, \boldsymbol{\xi}^{\partial})}{\mathrm{d}\mathbf{X}}\right)^T \left(\frac{\partial \mathbf{R}}{\partial(\boldsymbol{\eta}, \boldsymbol{\xi}^{\partial})}\right)^T \mathbf{v}\right]. \tag{3.19}$$

The derivatives of the metric vectors with respect to the grid variables are computed by reversely differentiating the source code, computing the metric quantities by AD and assigning the vectors $(\partial J/\partial \boldsymbol{\xi}^{\partial})^T$ and $(\partial \mathbf{R}/\partial(\boldsymbol{\eta}, \boldsymbol{\xi}^{\partial}))^T\mathbf{v}$ as inputs to the differentiated procedures. Notice that the vector $(\partial \mathbf{R}/\partial(\boldsymbol{\eta}, \boldsymbol{\xi}^{\partial}))^T\mathbf{v}$ can be automatically obtained using the adjoint solver. The aerodynamic gradient components, i.e. the term into the brackets, are computed in a dedicated tool (named *zAero*) implementing the differentiated metrics of the flow solver. The remaining operations, namely the calculation of the active and surface gradient, are separately carried out by means of two further softwares (*zMesh* and *zSurf*) embedding, respectively, the differentiated mesh deformation algorithms and the differentiated version of the NURBS routines. In this way shape optimization based on both active and surface gradients can be performed in a flexible manner.

3.2 Shape Optimization Algorithm

3.2.1 Gradient Smoothing and Projection

Design strategies involving the direct (CAD-free methods), or the indirect (methods using standard CAD representations) use of active grid points as design variables allow a high flexibility into conforming the geometry to the desired objectives. In both cases the active gradient is required in the optimization process and this is usually a vector of large dimension, as a considerable number of boundary points is normally employed for accurately meshing the geometry. Unfortunately, the active gradient components may

assume arbitrary directions and intensities. In fact, as the active gradient is generally less smooth than the objective function, its spectrum is often characterized by high frequency noise, as highlighted in Figure 3.2 for the VKI LS-59 turbine blade.

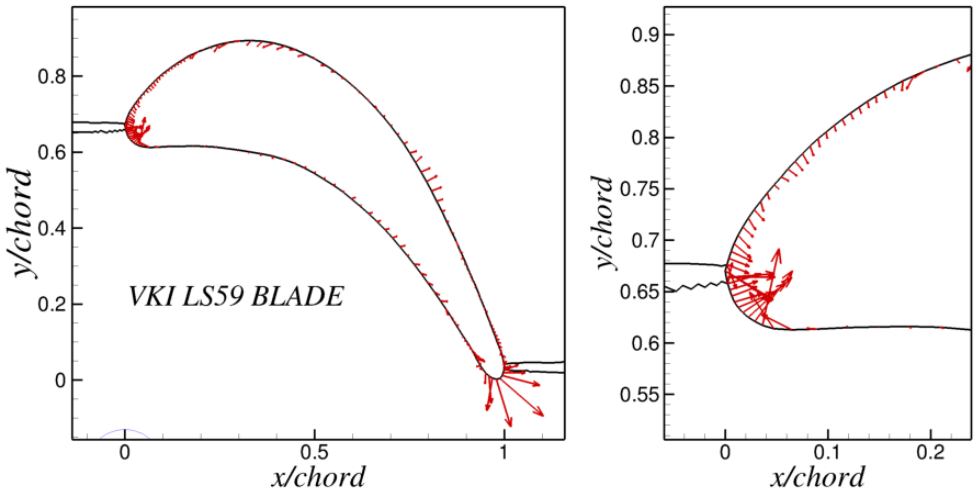


Figure 3.2. *Original active gradient components for the VKI LS-59 cascade.*

This drawback usually leads to a failure of the optimization cycle (e.g. occurrence of unfeasible designs), therefore some low pass filtering technique is to be introduced to stabilize the design process. Implicit gradient smoothing based on an elliptic procedure has been developed in the past in order to arrive at a stable and effective optimization process. In fact, as pointed out in [59], the smoother acts as a preconditioner for the optimization algorithm, therefore allowing for much larger design steps, which imply a reduction of the number of iterations needed for convergence. In the present thesis the original approach proposed by [60] is adopted. The active gradient given by equation (3.19) is therefore smoothed through the following implicit equation

$$\left(\widehat{\frac{\mathrm{d}J}{\mathrm{d}\mathbf{X}_b}}\right)^T_{i_b} - \epsilon\left[\left(\widehat{\frac{\mathrm{d}J}{\mathrm{d}\mathbf{X}_b}}\right)^T_{i_b+1} - 2\left(\widehat{\frac{\mathrm{d}J}{\mathrm{d}\mathbf{X}_b}}\right)^T_{i_b} + \left(\widehat{\frac{\mathrm{d}J}{\mathrm{d}\mathbf{X}_b}}\right)^T_{i_b-1}\right] = \left(\frac{\mathrm{d}J}{\mathrm{d}\mathbf{X}_b}\right)^T_{i_b}. \quad (3.20)$$

where $\widehat{dJ/d\mathbf{X}_b}$ are the smoothed gradients at the active boundary nodes $i_b - 1, i_b, i_b + 1$, $dJ/d\mathbf{X}_b$ the non-smoothed vector, and ϵ the smoothing factor, assumed as a prescribed constant parameter in the present work. The smoothed gradient is then projected onto a suitable direction (e.g. alongside

the direction normal to the surface or alongside the axial/tangential direction), thus preserving the required geometrical constraints of the design. The projection operator acts in fact both to reduce the allowable design space and to implicitly ensure the fulfillment of constraints that may be imposed on the geometry (as the chord length or the thickness of a blade trailing-edge). In this work the optimal configuration is searched by using the first-order steepest descent method. As a result the spatial displacements of the boundary grid points at a given iteration are computed as

$$\delta \mathbf{x}^T = -\lambda \left(\frac{\widehat{\mathrm{d}J}}{\mathrm{d}\mathbf{X}_b} \right)^T, \tag{3.21}$$

where λ is the step-size mandatory to convert the (projected) smoothed gradients into feasible spatial displacements of the active grid points. Since the use of a line search method to determine λ at each iteration is computationally extremely demanding, see e.g. [59], this factor is usually assumed as a constant parameter in FOSD. Similarly, equation (3.21) is applied in case of displacement of control points.

3.2.2 Local Shape Control

Despite the adoption of the smoothing procedure, the resulting spatial displacements given by (3.21) applied to all the active nodes may lead to large deformations of the geometry. Intersection or overlapping problems among nodes may occur during the optimization procedure. As a result, a uniform step-length λ value assigned to the active nodes $\mathbf{X}_b$ is not always the most effective solution as design strategy. As an example, the trailing-edge of a turbine blade is a fairly problematic part to move due to the presence of a relevant number of grid points in a zone characterized by a high curvature. In order to overcome the severe limitations related to the uniform value of λ, an additional strategy in which the step-length can vary from node to node is accompanied to the elliptic smoothing. The method defines the step-length through different functional forms in which λ becomes dependent on the position of the node. Purely geometrical criteria are also introduced to properly modify the initial uniform value of the step-length only in the most critical regions of the shape. In this work the simple Heaviside and the more sophisticated Gaussian-like functional forms are taken into account and written as

$$\begin{aligned} \lambda &= \begin{cases} 0 & r_i \leq r_{c_j} \\ \lambda & r_i > r_{c_j} \end{cases} \\ \lambda &= \lambda_{MAX} - (\lambda_{MIN} - \lambda_{MAX}) e^{\frac{-(r_i)^2}{r_{c_j}^2}}. \end{aligned} \tag{3.22}$$

In (3.22) r_{c_j} is the radius of the circle containing the points whose step-lengths have to be modified according to the relations (3.22) and r_i is the distance between node i and the center of the circle j. More precisely, the whole procedure for computing the spatial displacements is as follows: initially the smoothed gradient is computed and projected, then equation (3.22) is applied to obtain the step-lengths of all active nodes. The number of circles, their positions, and their radii can be specified at two different levels: first the designer can freely attempt to generate the circles by a trial and error operation and analyzing the resulting gradient vector; second a criterion based on the local mean radius of the geometry is also introduced in such a way that a circle is automatically positioned in the mesh zones characterized by a finer grid spacing, as the leading or the trailing-edge of a turbine/compressor blading. The radius of curvature at any active point is easily obtained by computing the discrete Hessian through some finite difference formulas. In this work the concavity of the discretized shape is expressed as

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 y}{\partial x^2} = \frac{y_{i+1} - 2y_i + y_{i-1}}{\delta \bar{x}^2}, \tag{3.23}$$

where $\bar{x}$ is the averaged spatial displacement in axial direction between nodes $i, i-1$ and $i, i+1$. As a result, the initial spatial displacement is proportionally altered only for the nodes contained inside the circles, while a uniform step-length is maintained for the remaining ones. Thanks to the proposed correction scheme (3.22), the satisfaction of the constraints connected to the mechanical resistance of the geometry (e.g. the trailing-edge minimum thickness of a blade or an airfoil) can also be guaranteed during the design process, without recurring to constrained optimization techniques, such as penalty function methods. Notice also that the combined use of a smoother and the proposed non-uniform step-length strategy may allow to further decrease the number of iterations required to reach the optimal design, resulting in a speed-up of the optimization. Furthermore, this shape control strategy is also an effective technique to prevent unfeasible geometry deformations in case of the use of control points as design variables.

3.2.3 Design Algorithm

The zFlow code and the adjoint solver have been combined within a fully automated optimization strategy, whose block diagram is shown in Figure 3.3. In order to easy accomplish design processes for a variety of fluid-dynamic problems the entire set of numerical tools necessary for the optimization loop has been embedded into a fully flexible Python package. The package has been structured to allow future direct interfaces with advanced Python-based optimization libraries, such as [61], and a relatively easy maintenance. The package embeds the following numerical tools:

- An unstructured grid generator based on an advancing front/Delaunay technique extensively described in [62].
- The real-gas flow solver (*zFlow*).
- The flow post-processing unit (*zfPost*). A mass-average and a mixed-out average schemes for real-gases are implemented to compute the mean quantities (e.g. velocities) at a given boundary surface.
- The corresponding real-gas adjoint solver (*JzFlow*).
- The aerodynamic gradient calculator (*zAero*).
- The active gradient calculator (*zMesh*). This code includes the smoothing and projection operators and the two differentiated versions of the mesh deformation algorithms discussed in chapter 2.
- The surface gradient calculator (*zSurf*), integrating the differentiated version of the NURBS curves.
- A multi-dimensional simulation initializer (*zStart*).

The latter tool is employed to properly initialize both the flow and adjoint calculation by interpolating the flowfields computed at the preceding iteration on the new grid, thus accelerating the convergence of both solvers. A fast interpolation is achieved by resorting to an inverse weighted-sum approach coupled to a *kd-tree* domain partitioning strategy as proposed by [63]. In addition, the convergence tolerance of the solvers may be independently fixed, thus allowing the capability of exploring so called non-converged (or *one shot*) optimization strategies, see for instance [64].
For 2D simulations the computational cost associated to the grid generation is a small fraction of the time required by the flow and adjoint solvers to converge. As a result, in the test cases presented in chapter 5, is often preferred to mesh the geometry at each optimization step instead of employing the grid perturbation algorithm to generate the new grid. A certain detrimental effect on the gradient accuracy can be attributed to this solution. However, the effect of this (possible) source of error tends to be fully compensated by the smoothing operator. Notice anyway that the use of robust algorithms such as the steepest descent method or some improved variants (e.g. Conjugate Gradient), see for instance [65], [66], allows to accept a certain level of gradient approximation without compromising the success of the optimization procedure. In FOSD, anyway, the best way to establish whether a gradient approximation is acceptable or not is the evaluation of the gradient effect on

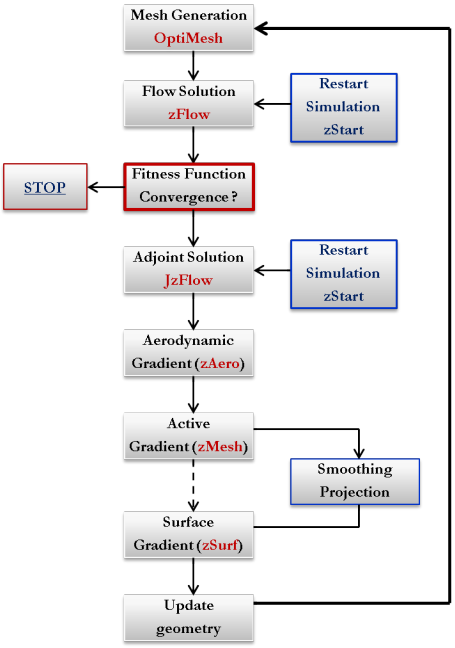


Figure 3.3. *Fully automated adjoint-based optimization algorithm for single-point optimization.*

the optimization process. Hence, the descent behavior of the process usually represents a clear indication of the suitability of gradient approximation. Both approaches for the calculation of mesh sensitivities (frozen domain and radial basis function), reported in chapter 2, can be alternatively employed in conjunction with grid generation.

As already mentioned, in the present thesis, the design loop sketched in Figure 6.2 is sequentially advanced using a (preconditioned) steepest descent method. The algorithm is particularly robust and fails only if no reduction of cost function can be obtained in gradient direction, which signifies the calculation of a very poor gradient.

3.2.4 Multi-Objective Optimization

Multi-objective optimization stands for an optimization of two or more fitness functions that are solved simultaneously. The response of a multi-objective optimization in normally not a single point, but rather a set of solutions that form a so-called Pareto front. Each point on the Pareto front satisfy the Pareto optimality conditions, which is stated as follows: a feasible vector of design variables is Pareto optimal is there exists no other feasible vector which would improve one of the objective without causing a corresponding worsening of at least another objective. Consequently, if exists a solution

that can be improved on one or more objective at the same time, it is not a Pareto optimal, or, in other words, it is dominated by the Pareto front.

Multi-objective optimizations can be handled in different ways that, exemplifying, can be grouped in two main categories. The first class employs a multi-objective optimization algorithm to simultaneously take into account a set of cost functionals. This strategy can be computationally expensive, but rather advantageous for complex and non-smooth problems. The second category, conversely, aims at transforming the multi-objective problem into a single one through the use of a weighted-sum function, i.e. a weighted sum of the objectives, with the remarkable property that $\sum_{i=1}^{N_{obj}} \omega_j = 1$, whereby ω_j is a single weight. The great advantage of such approach is the capability to deal with any optimization method suitable for single objective problems. Nevertheless, as pointed out in [67], the main drawback remains the appropriate choice of the weight coefficients. Also, if understanding the effects of changing weights on the objective is of interest, this method can become computationally demanding, since lots of simulations are required to construct the Pareto front.

Another problem concerning the use of a weighted approach is the summation of the objectives. As usually occurs, non-concurrent contributions can have values differing for orders of magnitude. This difficulty is often overcame by properly scaling the objectives or, for gradient-based methods, the corresponding gradient vectors.

In this work a multi-objective shape design algorithm based on adjoint is established and applied to a test case presented in 5. This method is based on a weighted sum approach and origins from the formulation of the gradient reported in 2, namely (2.15) and (2.16). In these expressions only the aerodynamic gradient depends on the adjoint flow-field, that in turn varies as the objective changes. The remaining operations are purely analytical and can be performed once, providing that the aerodynamic gradient is available. As a consequence, the weighted-sum fitness function gradient can be simply constructed by firstly summing the contributions of the aerodynamic gradients issuing from the set of cost functionals and then converting the resulting sensitivity vector into a usable active or surface gradient. The weighted-sum (surface) gradient vector can therefore be written as

$$\left(\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\gamma}}\right)^T = \left(\frac{\mathrm{d}\mathbf{X}_b}{\mathrm{d}\boldsymbol{\gamma}}\right)^T \left(\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\mathbf{X}_b}\right)^T \sum_{i=1}^{N_{obj}} \omega_j \left[\left(\frac{\partial J_j}{\partial \mathbf{X}}\right)^T + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^T \mathbf{v_j}\right]. \tag{3.24}$$

In turbomachinery field an interesting example of multi-objective optimization is represented by the so-called multi-point shape design. The term multi-point essentially means that the same functional is calculated at differ-

ent operating conditions (e.g. pressures, temperatures, angle of attack etc.). Thus the method allows to address with flow problems inherently characterized by relevant fluctuations of the operating conditions and, hence, may represent a viable methodology to obtain robust blade profiles, namely less sensitive, in terms of changes of performance, to the variations of the operating conditions.

3.2.5 Multi-Point Design Algorithm

To prove the potential of multi-point methodology in dealing with fluctuating operating conditions the previous single-point optimization algorithm has been adapted for enabling multiple evaluations of flow and adjoint solutions. The schematic view of the resulting design algorithm is depicted in Figure 3.4 for the same objective function computed at two different operating conditions. Both flow and adjoint solutions are separately converged and the associated aerodynamic gradient are evaluated. Therefore the gradients are linearly combined and the resulting weighted-sum aerodynamic gradient is processed according to the previous single-point optimization loop. Notice that multi-point approach enables to approximate, and then optimize, the mean value of the objective function over the range of variation of the operating conditions.

However, the main shortcoming of the multi-point technique remains the appropriate selection of weights and design points where evaluating the objective function. This choice affects the resulting mean value and, consequently, the results of optimization. A method to effectively overcome this limitation could be established by retrieving the concept of statistical expectancy for continuous aleatory variables. The stochastic mean of the objective J is defined as follows:

$$\bar{J}(\mathbf{P}, x) = \int J(\mathbf{P}) f_x(x) dx, \tag{3.25}$$

where the definite integral above is integrated over the set of possible values of the random variable(s) x, varying according to the probability density function f_x. If the weights and the set of design points are inferred through integration rules, as occurs in some uncertainty quantification methods [68], the statistical mean becomes well predicted by the resulting weighted-sum and can then be reliably used as a merit function in the multi-point optimization loop. Furthermore, in this case, the choice of weights and design points is mathematically rigorous and is no longer left to empirical laws or to designer experience. An example of (stochastic) multi-point is described in chapter 5.

By looking Figure 3.4, notice that all simulations are separately initialized by interpolating the old solution on the new grid to speed-up the overall

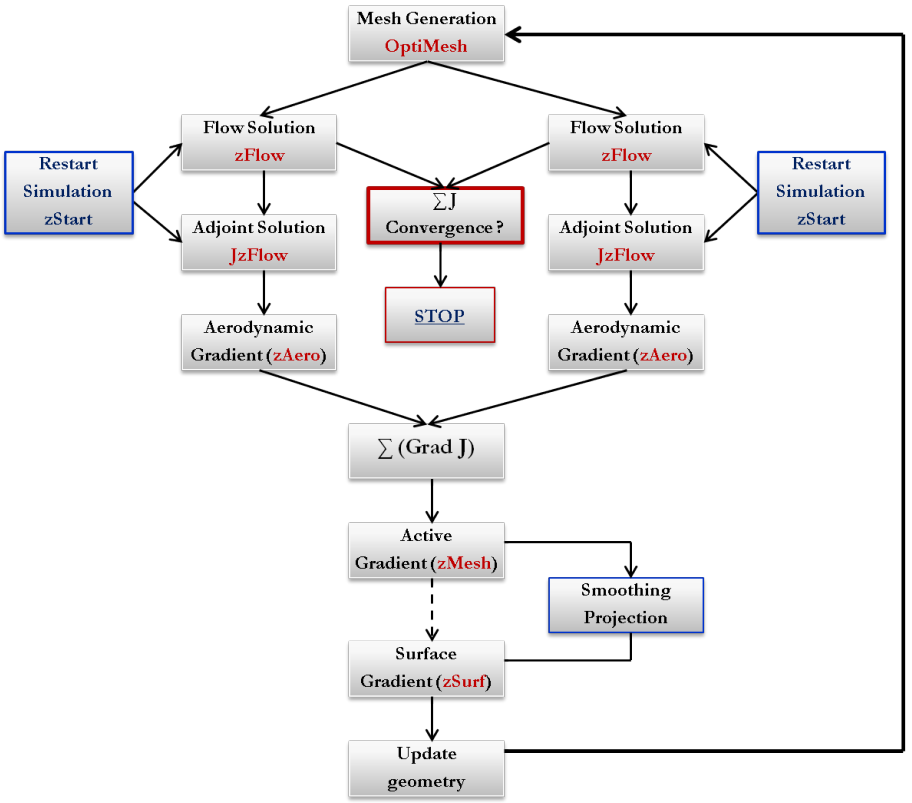


Figure 3.4. *Fully automated adjoint-based optimization algorithm for multi-point optimization.*

design process. The advancing towards the optimum is still carried out by the preconditioned steepest descent, the preconditioning operator being applied to the weighted-sum *active* gradient.

Chapter 4

Thermodynamic Modeling of Real Gas Flows

The assumption of ideal gas behavior is highly misleading in a number of realistic applications (e.g. Organic Rankine Cycles turbines, refrigerant compressors etc). Therefore several tools are currently available to accurately predict the thermodynamic properties of fluids in presence of strong real-gas effects. However, most of these tools are largely inefficient when coupled with existing CFD solvers. An effective solution to this issue is proposed in this chapter.

4.1 Introduction on Real-Gas Flow Modeling

The accurate modeling of thermodynamic and transport properties of fluids exhibiting non negligible real-gas behavior is crucial in many technical applications; in the energy-conversion technology fluids are often employed in far from ideal-gas conditions, and the availability of proper thermodynamic models is a necessary prerequisite for the accurate estimate of both component and whole system performance. The prediction of real-gas thermodynamic properties can be achieved by determining analytical expressions for the fundamental relation or, more commonly, by resorting to Equations of State (EoS) and their derivatives. Several formulations of the EoS and of transport properties correlations are available [69, 70, 71, 72, 73, 74], but their direct implementation implies a high computational cost in case of numerical studies, when a set of governing equations is iteratively solved, e.g. detailed Computational Fluid Dynamic (CFD) calculations [75], dynamic plant simulations [76].

As an alternative, Look-up Tables (LuT) can be used to represent the thermo-physical behavior of the fluid. In the thermodynamic regions of interest, a grid of nodal points (storing all thermodynamic and transport properties) is preliminary built. Within the discretized domain, the properties at any point are computed using fast interpolation methods, with a dramatic reduction in computational time [77, 78, 79, 80]. However this method does not automatically guarantee the thermodynamic consistency of the model, as instead it occurs in the case of direct EoS implementation. Thermodynamic consistency implies that, given a triple of thermodynamic properties, e.g. P, T, ρ, if $T = g(P, \rho)$ and $P = f(T, \rho)$, then $P \equiv f(g(P, \rho), \rho)$. Generally, most of the LuT approaches substitute the functions f, g with their approximate counterparts $\tilde{f}, \tilde{g}$ and this finally leads to $P \not\equiv f(g(P, \rho), \rho)$. The consistency error $\epsilon = P - \tilde{f}(\tilde{g}(P, \rho), \rho)$ can be minimized by increasing the accuracy of the LuT, improving either the number of mesh nodes or the order of the interpolation scheme. This can be usually done at the expense of a greater computational cost. On the other hand, consistency errors may induce significant non-smooth perturbations over CFD code iterations, which may even prevent the solver to converge.

This chapter presents a novel interpolation method for property calculation of real gases using LuT. At first the values of specific internal energy e are calculated for each node of a grid expressed in terms of specific entropy s and specific volume v by resorting to accurate EoS implemented in the software *FluidProp* [81]. The nodal values are then used to construct, by assigning a selected functional form, a local fundamental relation in the energy formulation ($e = e(v, s)$). Within any cell of the thermodynamic domain, the

coefficients of the functional form are calculated referring to the local grid data; the analytical treatment of the fundamental relation locally established guarantees that any thermodynamic property of any internal point is intrinsically consistent. The functional forms selected for e have indeed to exhibit a sufficient regularity (second order derivatives are required, for example, for the calculation of the speed of sound) and must fulfill the thermodynamic stability within the cell.

A similar approach has been adopted also for computing the transport properties. Two different functional forms are assigned to the dynamic viscosity μ and to the thermal conductivity k; coherently with the thermodynamic table, also the transport properties are expressed as functions of the specific volume and of the specific entropy ($\mu = \mu(v, s)$, $k = k(v, s)$). The sets of coefficients of the two interpolating functions are then computed at any cell on the basis of the transport properties stored within local grid points. Notice that no consistency issues have to be taken into account, since thermodynamic quantities (consistently calculated) do not depend on transport properties.

The chapter is structured as follows. At first the details of the LuT methodology are presented in Section 4.2. Issues concerning thermodynamic stability, accuracy and computational cost are discussed in Section 4.3, in which the LuT method is applied for the siloxane MDM and the carbon dioxide; both single and two-phase regions close to the vapor saturation line have been explored, for reduced temperature ranging between $T_r \simeq 0.6$ and $T_r \simeq 1.05$. Two examples of LuT application in combination with CFD are finally presented in Section 4.4.

4.2 Look-up Table Approach (LUT)

The standard computational scheme of any interpolation based method for the calculation of thermo-physical properties consists in a two-step procedure. As a first step, the thermodynamic mesh is generated based on the data provided by an existing database (tables of experimental data or even Equations of State and transport properties correlations). The whole set of properties is computed afterwards on the basis of the data stored in each node of the grid. The key aspects of a LuT algorithm, namely its efficiency and its accuracy, are then related to the methodologies introduced to resolve either steps. This section presents the methods developed in this work to achieve an optimal compromise between accuracy and efficiency, given that the thermodynamic consistency is intrinsically satisfied.

The main novelty of the present LuT procedure, with respect to other approaches available in the open literature, lies in the computation of the

thermodynamic properties; once the table is determined, nodal values are used to construct (by proper interpolation) a local fundamental relation, which can be analytically differentiated to compute in a straightforward way any thermo-physical quantity of interest. Each step of the procedure is separately described in the following.

4.2.1 Generation of the Thermodynamic Mesh

The construction of the thermodynamic mesh is based on the discretization of the saturation line according to a given temperature interval. The points can be uniformly or variably spaced through a spline-based reconstruction method, allowing also a refinement as the critical point is approached; an example of discretized vapor saturation line for siloxane MDM is given in Figure 4.1. The resulting saturation grid represents the support line on which the LuT is constructed by proceeding along directions locally orthogonal to the line.

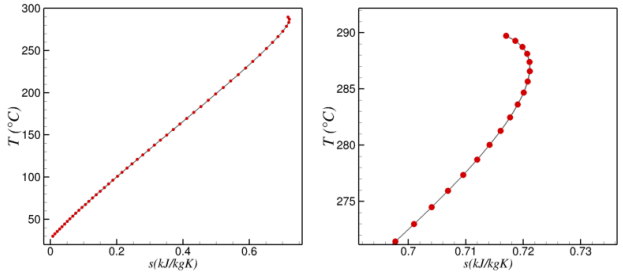


Figure 4.1. **Left:** *Discretized MDM vapor saturation line in the T-s plane.* **Right:** *Discretized MDM vapor saturation line in the critical region of the T-s plane.*

The normal spacing may be, in principle, specified using a generic pair of states (e.g. ρ, e), however, this choice strongly affects the shape and the thermodynamic regions covered by the resulting grid. Figure 4.2 depicts three different thermodynamic meshes generated starting from the same basis points and specifying the normal spacing as a function of $(log(v), s)$, $(log(T), s)$, and $(log(v), e)$. As well visible, the third grid extends in regions far from the critical point, while the former two also include the supercritical zone; this represents a crucial advantage in processes that occur, at least partially, in the dense-gas region (such as ORC turbine expansions). Furthermore, the use of $(log(T), s)$ as independent variables usually leads to relevant deformations of the mesh for different couples of thermodynamic properties. This results in higher computational costs and lower accuracies when the LuT method is applied in process or CFD calculation, in which dif-

ferent couples (e.g. ρ, e or P, ρ) are required. After several trials on a large variety of fluids, the definition of the normal spacing in the $(log(v), s)$ plane resulted as the best construction strategy. This allows to easily embed real gas regions in the thermodynamic domain, while preserving an acceptable skewness of the mesh for the couples commonly required in CFD calls. As a result, $(log(v), s)$ represents the basic choice of all grids constructed for the CFD simulations reported in the following.

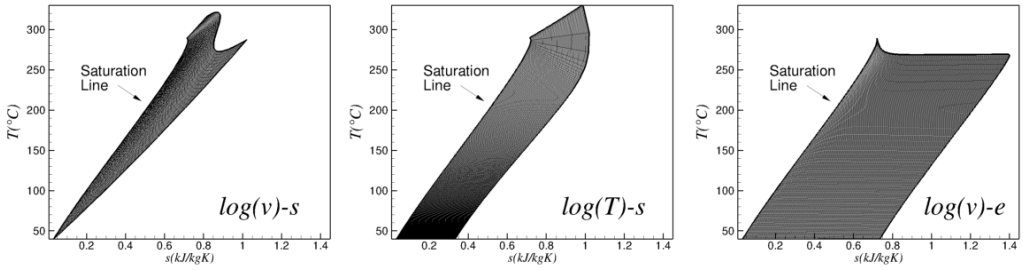


Figure 4.2. *Thermodynamic meshes generated using* $(log(v), s)$, $(log(T), s)$, *and* $(log(v), e)$ *as independent variables for normal spacing. For clarity, the tables are reported in the* $T - s$ *plane.*

The use of the saturation line as a basic support for LuT construction enables also the adoption of a multi-block approach, providing that the saturation line is split into several pieces, possibly with different discretization. The multi-block construction also allows to easily embed two-phase regions in the domain; the adjacent blocks result completely separated, with two different sets of points (one for each block) assigned at the boundary. Since the saturation line always belong to a boundary surface, all properties are continuous across any single cell of the domain. Figure 4.3 sketches, in the $T - s$ plane, two examples of thermodynamic mesh for, respectively, MDM (split into three blocks) and CO_2 (made by two blocks). As the original tables are generated in the $(log(v), s)$ plane, the corresponding grids in the T, s plane may exhibit a certain skewness, especially for fluids featuring a positive slope of the saturation line such as MDM. Notice also that mesh spacing can differ from one block to another and in both grid directions (e.g Fig. 4.3, left).

4.2.2 Construction of the Fundamental Equations

The major novelty of the present approach is the construction of a thermodynamic fundamental relation in the form $e = e(v, s)$ valid within each cell through an interpolation-based method. The pre-computed thermodynamic

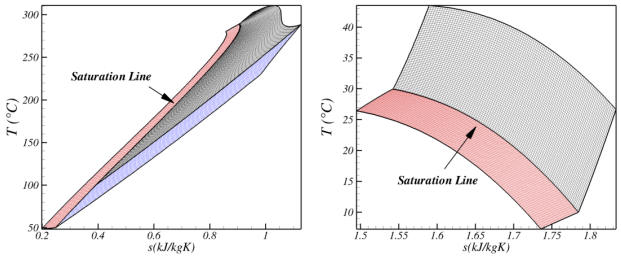


Figure 4.3. *Look-up Tables for siloxane MDM (left) and CO_2 (right) in the T-s plane.*

properties (stored at any grid node) are used as basis points. An analytic functional form is then required to mimic the real thermodynamic behavior of the fluid. In this context a simple bilinear and a more accurate bicubic bivariate interpolation methods based on edge-points are implemented. In particular the bicubic form was chosen in such a way to guarantee a degree of variability of the second derivatives within the cells; as a matter of fact the second derivatives of the fundamental relation are required for the calculation of some thermodynamic quantities, such as the specific heats or the speed of sound, as shown in Section 4.2.4. The bicubic functional form can be written as

$$\begin{aligned} e(v,s) = b_{00} + b_{01}v + b_{02}v^2 + b_{03}v^3 + \\ b_{10}s + b_{11}sv + b_{12}sv^2 + b_{13}sv^3 + \\ b_{20}s^2 + b_{21}s^2v + b_{22}s^2v^2 + b_{23}s^2v^3 + \\ b_{30}s^3 + b_{31}s^3v + b_{32}s^3v^2 + b_{33}s^3v^3 \end{aligned} \tag{4.1}$$

For each cell the weights $b_{i,j}$ of the interpolation functions are determined by using the values of entropy, specific volume, and internal energy of the surrounding 4 (bilinear) or 16 (bicubic) grid nodes.

This approach allows to manage the calculation of the whole set of weights for all grid cells in a pre-processing phase. As a result, the accuracy of the method may be further enhanced by using higher-order interpolating functions without compromising the computational efficiency of the calculation strategy; indeed, the overall expense remains only dependent on the number of cells and on the pair of state variables, as outlined in the following.

4.2.3 Transport Property Functional Forms

Similarly to the case of thermodynamic properties, the transport properties of any point of the domain are calculated according to a LuT approach. Therefore, a local analytical functional form is constructed for both the dy-

namic viscosity μ and the thermal conductivity k at any cell of the grid. In either cases the specific volume v and the specific entropy s have been taken as independents and simple bilinear functions are implemented, since no second order derivatives of transport properties are required in the context of fluid-dynamic computations. However, more complex relations can be adopted with no significant increase of the computational cost. The bilinear functional forms for μ and k can be written as:

$$\mu(v, s) = \beta_{00} + \beta_{01} v + \beta_{10} s + \beta_{11} sv \tag{4.2}$$

$$k(v, s) = \zeta_{00} + \zeta_{01} v + \zeta_{10} s + \zeta_{11} sv. \tag{4.3}$$

Each set of local values of the coefficients $\beta_{i,j}$, $\zeta_{i,j}$ of the functions is established resorting to the triple v, s, μ or v, s, k of the surrounding 4 grid nodes. As in the case of thermodynamic property calculation, the whole set of weights is determined in the pre-processing phase, thus not affecting the computational efficiency of the method.

4.2.4 Computation of Thermodynamic Properties

The whole set of thermodynamic quantities normally involved in process or CFD calculations, i.e. P, T, h, c, c_v, c_P, is provided by properly deriving the analytical expression of the fundamental relation. As a result, the thermodynamic consistency of the method is automatically satisfied. Since the speed of sound and the specific heats are functions of the second-order partial derivatives of e, the bicubic interpolation form is required in the solution of flow problems. The full set of thermodynamic properties as a function of s, v can

be written as

$$
\begin{aligned}
P &= -\left(\frac{\partial e}{\partial v}\right)_s \\
T &= \left(\frac{\partial e}{\partial s}\right)_v \\
h &= e - \left(\frac{\partial e}{\partial v}\right)_s v \\
c &= v\sqrt{\left(\frac{\partial^2 e}{\partial v^2}\right)_s} \\
c_v &= \frac{T}{\left(\frac{\partial^2 e}{\partial s^2}\right)_v} \\
c_P &= \frac{T\left(\frac{\partial^2 e}{\partial v^2}\right)_s}{\frac{T}{c_v}\left(\frac{\partial^2 e}{\partial v^2}\right)_s - \left(\frac{\partial^2 e}{\partial v \partial s}\right)^2},
\end{aligned}
\tag{4.4}
$$

in which c_P is only defined in single phase regions. Furthermore, most of the flux splitting numerical schemes suitable for arbitrary equations of state, as the one proposed in [53] and implemented in the flow solvers described in the following, require the calculation of secondary thermodynamic properties to reconstruct the fluxes among adjacent cells. These are commonly limited to derivatives of pressure with respect to density and to internal energy for inviscid simulations [51], which, properly combined, are used to recover the fluid speed of sound. The specification of temperature derivatives is further needed in case of viscous [82] and throughflow problems [83] solved by implicit time integration algorithms. If analytically rewritten as function of s, v, such derivatives can be additionally inferred as outputs of the present LuT method

as follows:

$$\begin{aligned}
\left(\frac{\partial P}{\partial e}\right)_{\rho} &= -\frac{1}{T\left(\frac{\partial^2 e}{\partial v^2}\right)_s} \\
\left(\frac{\partial P}{\partial \rho}\right)_{e} &= \frac{c^2}{c^2 + \left(\frac{\partial P}{\partial e}\right)_{\rho}\left(\frac{\partial e}{\partial v}\right)_s v^2} \\
\left(\frac{\partial T}{\partial e}\right)_{\rho} &= \frac{\left(\frac{\partial^2 e}{\partial s^2}\right)_v}{T} \\
\left(\frac{\partial T}{\partial \rho}\right)_{e} &= -\frac{v^2}{\frac{\partial^2 e}{\partial v \partial s}}
\end{aligned} \tag{4.5}$$

For a given input pair of thermodynamic properties (e.g. vs, ve, Ps, PT) the calculation strategy is divided into three main steps: the mesh cell containing the desired point is initially identified and the corresponding set of weights is separately selected; then the given pair is converted into the explicit state v, s by iteratively solving a non-linear equation (system); finally all thermodynamic properties are calculated by means of the set of equations (4.4) and (4.5).

The identification of the correct grid element where the thermodynamic point of interest lies represents the most demanding operation of any LuT approach and the time correspondingly required proportionally increases with the number of mesh nodes. A considerable reduction of the computational expense related to the search process is herein achieved by adopting fast nearest-neighbor techniques. Within the present LuT algorithm a robust *kd-tree* partitioning algorithm [84, 85] has been implemented and tuned to decompose the original thermodynamic domain in a tree structure suitable for a highly efficient search procedure.

In particular, several tree structures are assembled and stored, one for each couple of thermodynamic states required in CFD codes. Such approach leads approximately to the same computational cost for a single search procedure using different input pairs, at least on meshes characterized by similar skewness. Moreover, as *kd-tree* methods act primarily on Euclidean spaces, an algebraic transformation is initially applied to the thermodynamic properties in order to map the original values within the normalized space 0-1. This ensures an acceleration of the searching process.

Apart from the favorable case of v, s, which seldom occurs in CFD or process calculations, any other pair of states cannot be directly used for the computation of the thermodynamic properties, as the fundamental relation explicitly depends on v, s. Therefore the input state has to be preliminarily converted into the independent state v, s by solving a non-linear problem in

the form $\mathbf{f}(v,s) = \mathbf{f}_0$, $\mathbf{f}_0$ being a vector of two given properties. The most demanding case occurs when none of the input variables correspond to the independent variables of the fundamental relation. Considering for example PT as independent variables, the non-linear system to be solved to restore v, s can be written as:

$$\begin{aligned} P(v,s) &= P_0 \\ T(v,s) &= T_0. \end{aligned} \tag{4.6}$$

Equations (4.6) can be solved by any iterative algorithm available for non linear systems. In this work some Newton's updates are employed to hasten the solution. The linearization of equations (4.6) simply holds:

$$\begin{aligned} P(v+\delta v, s+\delta s) &= P(v,s) + \frac{\partial P}{\partial v}\delta v + \frac{\partial P}{\partial s}\delta s = P_0 \\ T(v+\delta v, s+\delta s) &= T(v,s) + \frac{\partial T}{\partial v}\delta v + \frac{\partial T}{\partial s}\delta s. = T_0 \end{aligned} \tag{4.7}$$

By plugging the fundamental relation in the linearized equations, the full system is more conveniently expressed as

$$\begin{bmatrix} \frac{\partial P}{\partial v} & \frac{\partial P}{\partial s} \\ \frac{\partial T}{\partial v} & \frac{\partial T}{\partial s} \end{bmatrix} \begin{bmatrix} \delta v \\ \delta s \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 e}{\partial v^2} & -\frac{\partial^2 e}{\partial v \partial s} \\ \frac{\partial^2 e}{\partial v \partial s} & \frac{\partial^2 e}{\partial s^2} \end{bmatrix} \begin{bmatrix} \delta v \\ \delta s \end{bmatrix} = \begin{bmatrix} P_0 - P(v,s) \\ T_0 - T(v,s) \end{bmatrix}, \tag{4.8}$$

where the vector $\delta v, \delta s$ represent the incremental steps of the solution v, s. A very fast convergence is usually achieved by selecting as initial guess for the iterative process the couple v, s belonging to the mesh node closest to the assigned one. Given the couple v, s corresponding to the point of interest, the thermodynamic quantities are straightforwardly evaluated by applying the relations (4.4), while the transport properties are simply given by the functional forms (4.2) and (4.3).

4.3 Analysis of the Method

In this section the accuracy and the computational efficiency of the method are carefully assessed. Carbon dioxide and siloxane MDM are considered as reference compounds for the investigations. The thermodynamic tables are constructed on the basis of the data provided by *FluidProp*.

4.3.1 Thermodynamic Stability

Each fundamental relation describing thermodynamic equilibrium states has to satisfy the so-called stability conditions [86]. To guarantee the thermodynamic stability it is necessary and sufficient that the fundamental relation is

convex with respect to the specific entropy and the specific volume, namely that, for a pure substance:

$$d^2e = \frac{1}{2}\left(\frac{\partial^2 e}{\partial s^2}\right)_v ds^2 + \left(\frac{\partial^2 e}{\partial s \partial v}\right) ds dv + \frac{1}{2}\left(\frac{\partial^2 e}{\partial v^2}\right)_s dv^2 > 0 \tag{4.9}$$

Therefore, the quadratic form (4.9) which expresses d^2e is required to be positive definite or, equivalently, that all the principal minors of the matrix of the quadratic form are positive. As the matrix of (4.9) is the Hessian matrix ($H = H(s, v)$) of e, the stability of the fundamental relation is guaranteed if and only if [86]:

$$\begin{cases} H_{1,1} = \left(\frac{\partial^2 e}{\partial s^2}\right)_v > 0 \\ det(H) = \left(\frac{\partial^2 e}{\partial s^2}\right)_v \left(\frac{\partial^2 e}{\partial v^2}\right)_s - \left(\frac{\partial^2 e}{\partial s \partial v}\right)\left(\frac{\partial^2 e}{\partial v \partial s}\right) > 0. \end{cases} \tag{4.10}$$

The bicubic relation expressed in (4.1), therefore, must guarantee the thermodynamic stability in the region of application. This simply leads to local conditions, since a different fundamental relation is defined on a single mesh element. As a result, a simple method to ensure the local stability of a given bicubic function requires the verification of the conditions specified by (4.10) over the domain of application. In the present work this operation is performed immediately after the definition of the full set of bicubic relations. In particular, the stability conditions have been investigated in additional verification points obtained by equally dividing each cell of the support mesh in eight parts. These points include the barycenter of the cells, which represent the points where the interpolation error tends to be higher. The fulfillment of the stability conditions have been satisfactorily verified for different fluids and for mesh of different dimensions over the entire (possibly multi-block) domain. Figure 4.4 shows, in the $T - s$ diagram, how the stability conditions are fulfilled by the the whole set of fundamental relations established in the domain using the LuT. The results reported in Fig. 4.4 refer to a two-block LuT constructed for the siloxane MDM using 5000 cells for the single-phase block and 2500 cells for the two-phase block. The verification points are about 40000 for the vapor/supercritical region and about 20000 for the two-phase region. It can bee seen how conditions (4.10) are respected at any investigated point of the thermodynamic domain and not only on the support grid nodes, as already guaranteed by the stability of the Span-Wagner EoS [73] employed for the generation of the mesh.

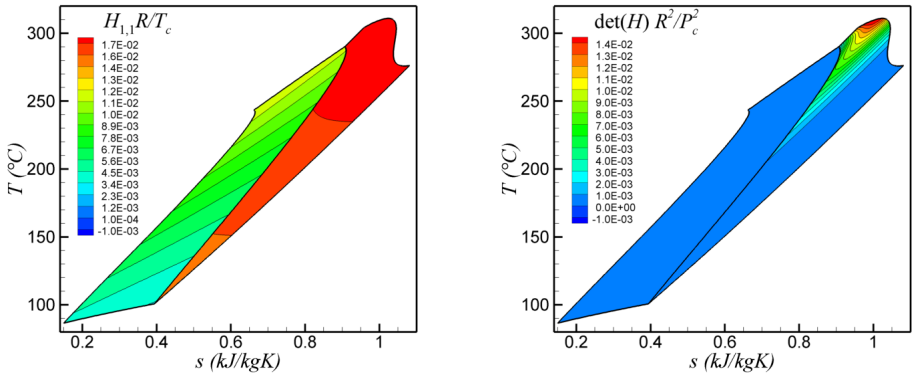


Figure 4.4. *Verification of the thermodynamic stability for the siloxane MDM.* **Left:** *Non-dimensional value of* $H_{1,1}$. **Right:** *Non-dimensional value of* $\det(H)$. $H = H(s, v)$ *is the Hessian matrix of the fundamental relation* $e = e(s, v)$ *established at a certain cell of the domain.*

4.3.2 Accuracy

The accuracy of the proposed LuT approach is herein evaluated by quantitatively estimating the deviations of the interpolated values against the quantities given by the original models implemented in the *FluidProp* database, namely the Span-Wagner EoS for both fluids [73, 87, 88]. Two ensembles of 500 randomly distributed points are initially generated and used for the analysis. The former group is representative of the regions far from the critical point, the latter one is conversely extrapolated from the critical one. The results shown in the following refer to the mean percentage relative error of P, c, c_P as a function of the grid node number for v, s as input pair. Further calculations, not reported herein, gave similar outcomes for the remaining couples.

Figures 4.5 and 4.6 display the convergence of the mean relative error as a function of the table nodes. For both fluids, the error tends to be higher in the critical region. This behavior is most probably due to strong real gas effects, which induce high deviations of the thermo-physical properties between two close states. As a consequence, the bicubic shape is less suitable to predict the trend of internal energy within a single cell element, resulting in a degradation of the accuracy of its first and second-order derivatives. As expected, Figures 4.5 and 4.6 also show a lower accuracy for c and c_P. This can be easily explained by inferring the order of the interpolating polynomials used for approximating the three properties. Such orders, with respect to the

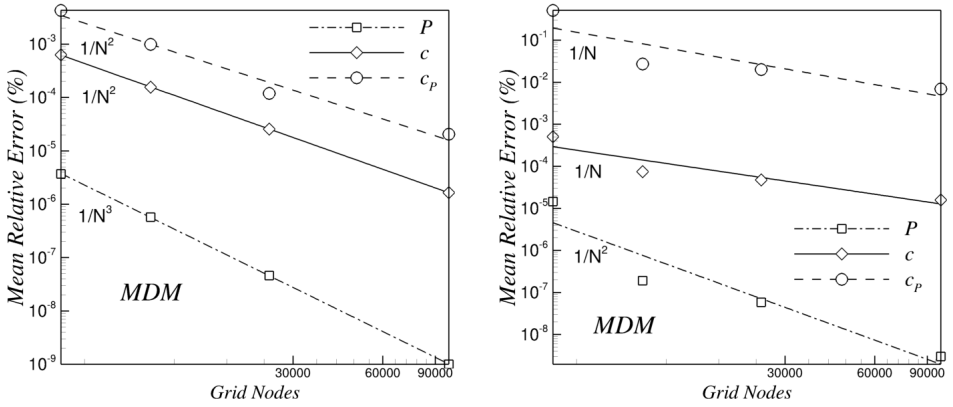


Figure 4.5. *LUT accuracy for MDM.* **Left:** *Mean relative error far from the critical point.* **Right:** *Mean relative error in the critical region.*

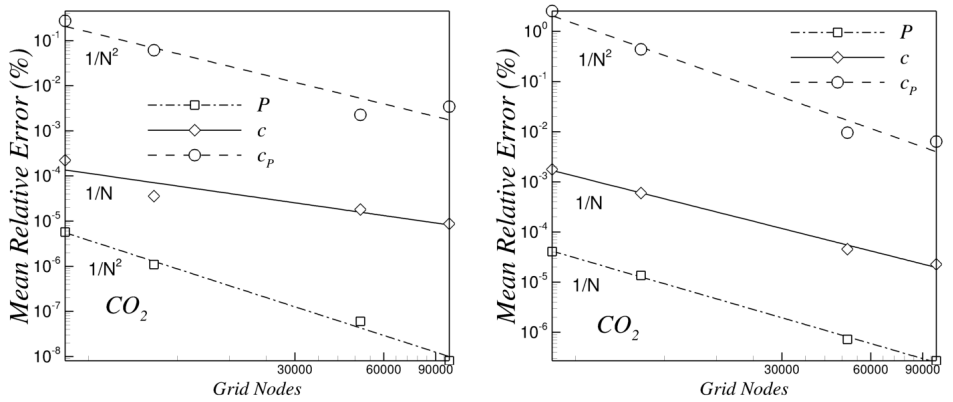


Figure 4.6. *LUT accuracy for CO_2.* **Left:** *Mean relative error far from the critical point.* **Right:** *Mean relative error in the critical region.*

independent variables, can be synthetically written as

$$\begin{aligned} P &\propto v^2 \\ c &\propto v^{\frac{3}{2}} \\ c_P &\propto s. \end{aligned} \tag{4.11}$$

As shown, pressure (and temperature) are described by quadratic functions, whereas specific heat capacities are basically linear and, therefore, result to be calculated with minor accuracy. In general the interpolation error decreases quadratically with the number of mesh elements ($1/N^2$) far from

the critical point, whereas lower convergence orders are achieved in proximity of the critical region.

4.3.3 Computational Efficiency

The computational efficiency of the LuT approach is now assessed by comparing the time necessary for a direct evaluation of the thermo-physical properties through the equation of state implemented in *FluidProp* with that required by the LuT algorithm. The mean ratio of the computational burdens of the two methods (LuT cost divided by the direct evaluation cost) for the set of the previously described 500 points, randomly distributed in the thermodynamic space (in both critical and non-critical regions) is summarized in Tables 4.1 and 4.2. For a given input pair the whole set of thermodynamic and transport quantities is evaluated on a fixed thermodynamic mesh composed by 10000 elements.

	vs	ve	Ps	PT
MDM	10^{-3}	10^{-2}	10^{-2}	10^{-1}
CO_2	10^{-3}	10^{-2}	10^{-3}	10^{-1}

Table 4.1. *Mean computational time ratio between the LuT and the direct method far from the critical region.*

	vs	ve	Ps	PT
MDM	10^{-3}	10^{-3}	10^{-2}	10^{-2}
CO_2	10^{-3}	10^{-3}	10^{-3}	10^{-2}

Table 4.2. *Mean computational time ratio between the LuT and the direct method in the critical region.*

The present interpolation-based scheme outperforms the direct evaluation method for any pair of thermodynamic states. The computational cost is reduced of at least three orders of magnitude with respect to direct calls where v, s (or one of them) explicitly appears as input state. Up to three orders of magnitude gain can be achieved by calling with v, e in critical regions, which represents the most used couple in a CFD solver. Conversely,

the use of P, T as independent variables needs the solution of the non-linear system (4.6), which entails a reduction of the computational gain offered by the LuT method. However, this type of call is not required in real-gas CFD solvers adopting conservative variables.

The influence of the grid dimension on the total computational burden is now investigated. The whole set of thermodynamic properties (previously based on a mesh of 10000 cells) is now assessed on grids of increasing size. The investigation is performed for MDM in the region far from the critical point. The computational expenses are coherently evaluated on meshes having similar skewness.

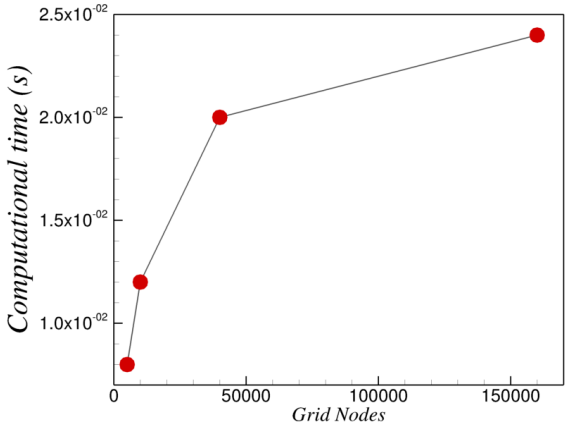


Figure 4.7. *Total computational cost of 500 search operations carried out for MDM in non-critical regions on grids of varying dimension.*

The results shown in Figure 4.7 point out a very interesting trend. The computational cost increases less than linearly with the number of mesh elements, leading to the considerable advantage of making use of very fine mesh, i.e. very accurate properties calculation, without considerably affecting the whole cost of the simulation. Since most of the computational effort is due to the cell identification rather than to the property evaluation (using the same call the evaluation time is almost the same on different grid size), this outcome can be completely conferred to the characteristics of the *kd-tree* partitioning algorithm. In fact, grids covering similar thermodynamic areas but characterized by different number of elements have comparable tree structures, namely the thermodynamic domain is divided in the same number of main partitions. Therefore, the introduction of further mesh points only leads to a modification of the tree structure at a lower level, resulting in an

increase of local partitions of the domain for a finer mesh.

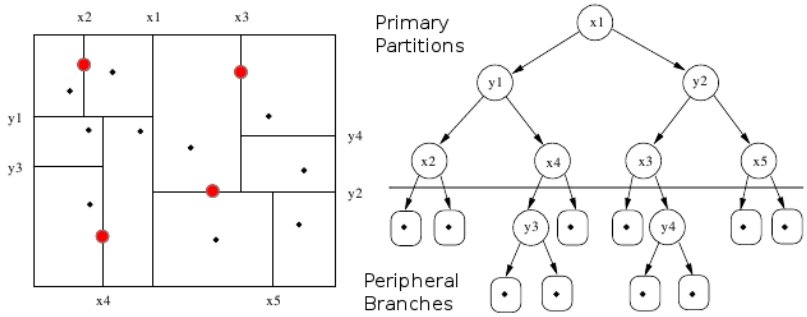


Figure 4.8. *Overview of kd-tree partitioning algorithm. The values shown are the coordinates of the dividers (red circles), namely the basis points of the table. Black circles indicate the query points.*

Hence, the main architecture of the tree is fully conserved between a coarse and a fine grid, while only its sub-structure is changed. This allows to envisage a search procedure as split into two main parts: the search starts by proceeding along the primary partitions of the tree (global search), which are approximately equal between the two grids, then, in the case of a fine grid, the cell of interest is reached through the peripheral branches (local search) effectively arranged on the main tree structure. For a coarse grid, contrarily, the search stops at the first (global) level; see Figure 4.8. The additional time consumed is then only related to the local search, which is usually very efficient. A further investigation conducted by using carbon dioxide provided the same indications. The potential of this feature of the LuT algorithm can be fully exploited in CFD simulations, as those reported in the next section.

4.4 Application to Turbomachinery Flows

The proposed LuT method has been embedded into existing CFD solvers for the simulation of turbomachinery flows of organic fluids. In particular TzFlow code [83] for throughflow calculations of multistage machines and zFlow code [51] for blade-to-blade cascade calculations are considered. The performances of both solvers for realistic turbomachinery applications are investigated in the following, considering a transonic multistage radial turbine and a supersonic axial turbine nozzle.

4.4.1 Throughflow Simulation of a Multi-Stage ORC Radial Turbine

The throughflow simulation of the transonic six-stage radial turbine proposed in [9], designed to operate with siloxane MDM, is first considered. The calculation presented in [9] was originally performed by coupling the TzFlow solver with the external thermodynamic library *FluidProp*, which models the MDM fluid with the Span-Wagner EoS [87]; this reference calculation is hereinafter indicated as SW. In the present context the TzFlow solver is extended to implement the proposed LuT approach, and then applied to the same machine using tables constructed with the Span-Wagner EoS, so to guarantee full coherence between the physical models used in the LuT and SW calculations. The comparison between the results achieved with LuT and the SW methods allows to highlight the impact of the thermodynamic treatment on the computational cost and accuracy of a CFD simulation for a whole machine.

The isentropic thermodynamic process ideally experienced by the fluid within the machine is represented in the left frame of Figure 4.9, where also the table used for the LuT calculation is provided in the $T-s$ state diagram.

From the fluid dynamic point of view the TzFlow solver implements the so-called CFD-based throughflow model. In a nutshell, the flow model is obtained by averaging the three dimensional Euler equations in the azimuthal direction; this leads to a highly simplified axisymmetric problem, but proper source terms have to be introduced to model the flow deflection imparted by the blades, the aerodynamic losses, and the blade blockage [83]. From the thermodynamic point of view the solver is formulated in conservative form, hence the couple e, v represents the input pair in many instances of the calculation; however in the throughflow model the loss source term, formulated as a friction force, is expressed as $D = T\frac{\partial s}{\partial t}$ (with t the streamwise coordinate). As a result, differently from a standard inviscid flow solver, throughflow simulations also require the calculation of temperature, entropy, and their derivatives.

The TzFlow calculations of the six-stage centrifugal turbine have been performed using Craig & Cox loss model [89], and assuming elliptic-arc mean line blades (following the design remarks proposed in [90]). The computational grid is composed by about 30 kcells (40×750 cells in spanwise and streamwise direction respectively), arranged in a structured multi-block assembly and composed by cascades and channels; the relatively high number of cells, with respect to the common experience on axisymmetric calculations, is motivated by the spatial resolution required for an appropriate modeling of the gradients of blockage in bladed zones (multiplied by the twelve cascades of the machine). For the LuT calculations four thermodynamic tables

are considered, composed by 5000, 10000, 20000 and 50000 cells, all of them covering the same area on the state diagram but characterized by different levels of refinement.

An example of result for real-gas throughflow calculations is shown in the central frame of Figure 4.9, where the distributions of total enthalpy and entropy on the meridional plane are depicted for the LuT calculation performed with the 5 kcells table. As clearly visible, the total enthalpy only drops across rotors, due to the work exchange (adiabatic flow conditions are assumed, as usually done in turbomachinery); conversely the entropy increases progressively along the machine, due to the action of losses in stator and rotor cascades; both the quantities exhibit large spanwise gradients in the last stages, due to the significant flaring of the channel. All these features are properly captured by both the SW and LuT models, from the quantitative and the qualitative point of views. A relevant quantitative comparison is provided in the right frame of Figure 4.9, where the midspan profiles of static pressure and static temperature for SW and LuT models are shown to be perfectly overlapped.

This indicates that the present LuT algorithm (with the specific grid used) allows to accurately predict the evolution of thermo-physical quantities along the machine.

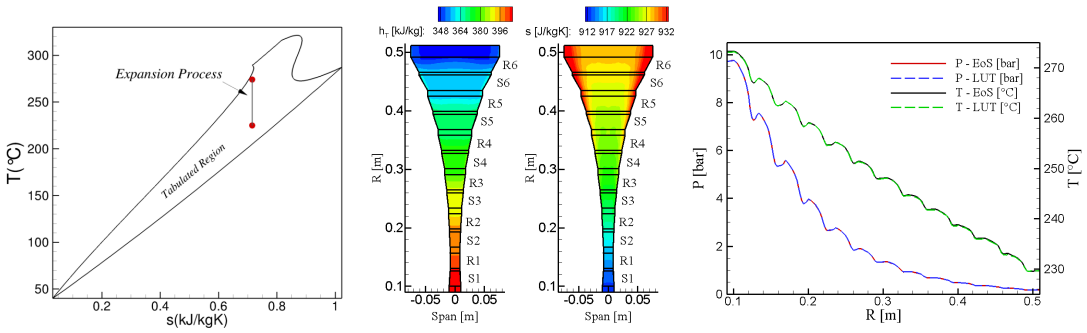


Figure 4.9. **Left:** *Tabulated region for throughflow simulation (MDM).* **Center:** *meridional evolution of total enthalpy and entropy field within the 6-stage centrifugal turbine.* **Right:** *comparison between SW and LuT results in terms of streamwise distribution of pressure and temperature at turbine mid-span.*

The crucial quantitative comparison among different models is now performed in terms of computational efficiency. For this comparison the 5 kcells table has been first considered for the LuT model and, beside the SW model, further calculations have been performed using simpler thermodynamic mod-

els, namely the polytropic ideal gas (PIG) and the polytropic Van der Waals (PVdW) gas models. These latter models are also available in TzFlow by direct implementation of the EoS within the solver. It should be noted that this procedure represents the most efficient strategy to determine the thermophysical properties of a working fluid, as just simple analytical operations are required to solve for the full set of thermodynamic relations. Both polytropic models predict the fluid properties by using a constant specific heat ratio, obtained by conveniently averaging the γ values along the ideal expansion line shown in the left frame of Figure 4.9.

Table 4.3 compares the savings in computational cost with respect to SW (implemented trough the external library) provided by the LuT, PIG and PVdW models. Since in a CFD calculation a large amount of resources are required by the integration of the flow equations (e.g., for the calculation of fluxes, the solution of linear system, etc.), to properly analyze the effectiveness of the thermodynamic approach the overall computational cost is conveniently split into two contributions: the first one is the cost needed for the fluid-dynamics calculation, which is a fixed and irremovable fraction of the total expense; the second one is the cost needed for the calculation of thermodynamic properties.

For calculations performed using PIG or PVdW models the thermodynamic cost is a few percent of the total one, so their cost represents a rough estimate of the fluid-dynamic cost - the PIG and PVdW simulations share the same computational time indeed, as seen from Table 4.3. The SW computational cost being the double of the PIG/PVdW one, the thermodynamic cost of the calls to the external library roughly equates the fluid dynamic one. When the LuT approach is used, the fast-searching algorithm and the local analytical model enable to strongly reduce the thermodynamic cost, saving about 2/3 of the corresponding one required by the SW simulation.

	PIG	$PVdW$	$LUT(5kc)$	SW
Time saving	53 %	52 %	33 %	-

Table 4.3. *Total computational cost saved by using different thermodynamic modeling approaches compared to SW model for the throughflow simulation.*

The influence of the grid size on the total computational cost of the LuT simulation is now investigated, with reference to the PIG/PVdW computational cost. The three refined grids are used for the comparison, beside the original one. From the data reported in Table 4.3 it is possible to obtain

a rough estimation of a PIG computation, which results approximately 150 seconds. The difference between the PIG and LuT burdens is around 65 seconds for a grid of 5 kcells. Increasing the number of grid nodes of one order of magnitude (50 kcells) the extra cost, purely thermodynamic, is just about 13%. As a result the grid size has a slight relevance on the total computational cost of the simulation, allowing for the use of fine (and accurate) thermodynamic meshes, if required by the problem, without compromising the effectiveness of the computational procedure.

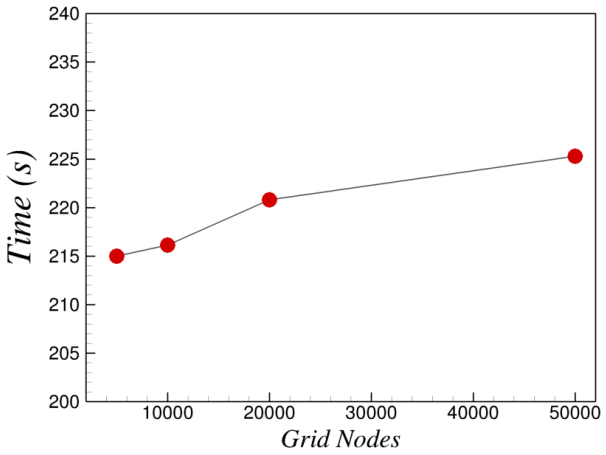


Figure 4.10. *Total computational cost of the throughflow simulation using different grid size.*

4.4.2 Blade-to-blade Simulation of a Supersonic ORC Cascade

The second test case focuses on the blade-to-blade simulation of a supersonic ORC nozzle at mid-span. The cascade is characterized by an extremely high expansion ratio, which leads to a design exit Mach number of 2. As a result, inter-blade channel is of converging-diverging shape, with a sonic throat upstream of the discharge section. The working compound is siloxane MDM also in this application; the ideal thermodynamic process through the cascade is sketched in the left frame of Figure 4.11, along with the thermodynamic table used for the LuT simulations.

The calculations have been performed with a simplified version of the zFlow code, which solves for the two-dimensional inviscid flow equations in Cartesian coordinates; the codes zFlow and TzFlow were developed on the same bases, sharing the same numerical solution procedure and most of the computational features. After a preliminary grid dependence analysis, a

computational grid of about 20000 triangular elements has been selected for the present blade-to-blade calculations; the relatively large number of cells, with respect to typical values used for inviscid blade-to-blade simulations, is motivated by the presence of strong shocks. As done for the previous test case, the results of the LuT approach are now compared with the ones achieved with SW, PIG and PVdW models, first considering a 5 kcells table for the LuT calculation.

The accuracy of the LuT approach is first evaluated against the solution gained by using SW model. The central frame of Figure 4.11 shows the distribution of Mach number, from which the sonic throat and the trailing edge shock can be observed. In the same Figure the streamlines are also reported, showing the considerable flow turning both within and, especially, downstream of the blade channel.

Once again, all the flow features are captured by the LuT approach from both the qualitative and quantitative point of views. In particular a relevant quantitative comparison is provided in the right frame of Figure 4.11, where the distributions of isentropic Mach number along the blade surface for SW and LuT models are shown to perfectly overlap, confirming the accuracy of the solution provided by the LuT algorithm.

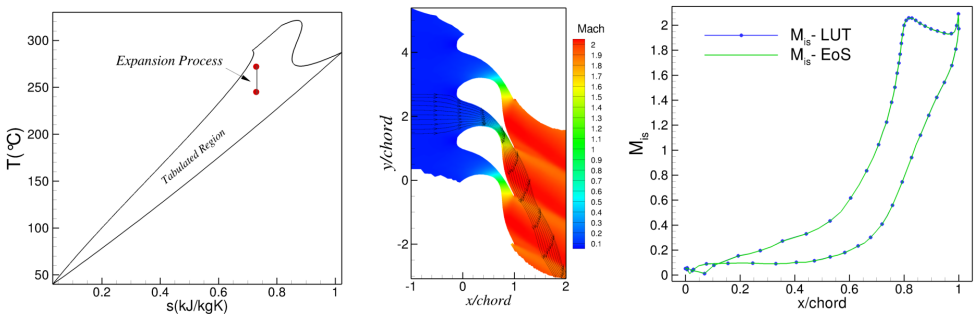


Figure 4.11. **Left:** *Tabulated region for supersonic turbine simulation (MDM).* **Center:** *Blado-to-blade Mach flow field at nominal operating conditions.* **Right:** *Predicted isentropic Mach number distribution along the blade surface.*

The performances of the LuT method are now analyzed. In full coherence with the preceding throughflow application, the same thermodynamic models are herein utilized for the analysis.

The time savings provided by LuT, PIG and PVdW approaches compared to the SW one, reported in Table 4.4, are larger than that in the throughflow case. In particular, for the SW calculation the thermodynamic cost amounts

	PIG	*PVdW*	*LuT*	*SW*
Time saving	83 %	82 %	65 %	-

Table 4.4. *Total computational cost saved by using different thermodynamic modeling approaches compared to SW model for the supersonic cascade simulation.*

to 80% of the overall computational cost, considerably higher than that occurring in the throughflow simulation; as a result, the LuT approach provides an even more significant saving, almost the double of that achieved for the throughflow test-case.

As already found for the throughflow calculation, the influence of the table size on the overall computational burden is limited, as depicted in the left frame of Figure 4.12; a further study is finally carried out to evaluate the sensitivity of the convergence rate on the table size. Very similar residual values are achieved at end of the simulation going from 5 to 200 kcells, as visible in the right frame of Figure 4.12. However, lower convergence rates of the flow solver are observed for coarser thermodynamic meshes, as highlighted in the same figure for LuT tables composed by 5 and 10 kcells.

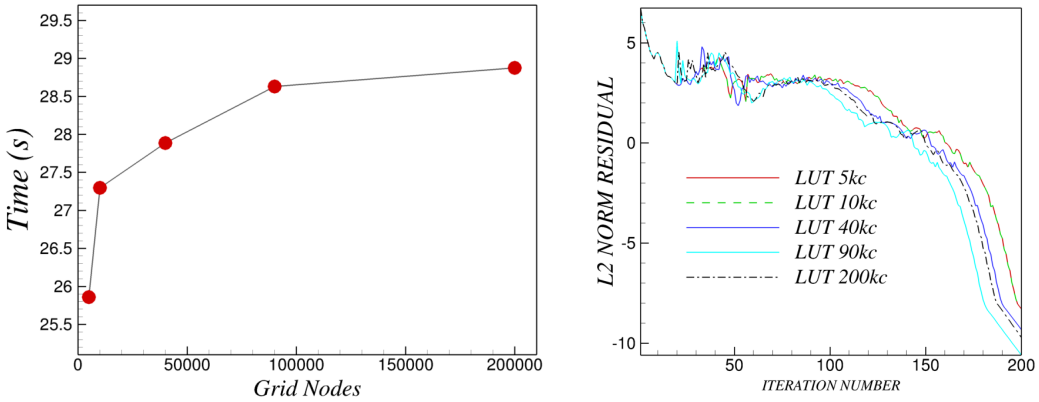


Figure 4.12. **Left:** *Total computational cost of the blade-to-blade simulation using different grid sizes.* **Right:** *Convergence rate of the flow solver as a function of grid size.*

Chapter 5

Shape Optimization Applications

This chapter presents some examples of shape optimization for inviscid flow through converging-diverging channels and turbine blade passages. The aim is to investigate the potential of the previously described adjoint-based methodology in improving the performances of existing geometries. The test cases reported in the next sections follow the natural evolution of the work, hence the discussion starts by illustrating the results of a simple inverse design problem and concludes providing the outcomes drawn by the re-design of a supersonic ORC turbine stator. The following test cases are extensively described:

- *Shape Optimization for Quasi-1D Transonic Flow through Diverging Channels*
- *Shape Optimization for Quasi-Uniform Supersonic Flow through Converging-Diverging Nozzles*
- *Shape Optimization of a Transonic Turbine Cascade*
- *Shape Optimization of a Supersonic ORC Turbine Cascade*

5.1 Inverse Design for Quasi-1D Transonic Flow through Diverging Channels

5.1.1 Problem Statement

The first example concerns internal flows inside a channel of variable area. The physical behavior of the flow is modeled by the compressible quasi-1D Euler equations. The channel shape is described by the following hyperbolic-tangent relation (5.1)

$$y(x) = a + b{\cdot}tanh(cx - d). \tag{5.1}$$

The independent parameters of (5.1), i.e. the design variables of the problem, are initially set to $a = 1.75, b = 0.699, c = 0.8, d = 4$. The purpose of this first design exercise is to evaluate the capability of the adjoint method for inverse design applications. Inverse design methods are often use to generate geometry configurations able to reproduce an exact matching between a target pressure trend and a computed one along some prescribed direction. In this case the stream-wise distribution of the static pressure is firstly given by fixing the design parameters values as previously. Then the shape of the channel is altered by acting on c, d parameters, now set to $c = 1, d = 3.8$. The aim of the optimization is therefore to establish the equivalence between the initial pressure distribution, assumed as reference, and the current one. As a result the fitness function of the problem can be written as

$$J = \frac{1}{2}(p_i - p_{tg_i})^2. \tag{5.2}$$

where p_i, p_{tg_i} are, respectively, the pressure and its target counterpart at a given cell. In order to obtain a shock formation inside the diverging part of the channel, supersonic inflow and subsonic outflow boundary conditions are imposed. Inlet and outlet Mach numbers have been fixed equal to 1.5 and 0.6. Computational grid consists of about 300 equi-spaced cells. The working fluid is air modeled as polytropic ideal gas.

5.1.2 Gradient Validation

In order to properly deform the channel geometry, the sensitivity derivatives of the fitness function with respect to each design variable *a, b, c, d* are to be computed. However, for quasi-1D flow problems with fixed mesh size, and more in general for fluid-dynamic problems in which the cost functional does not straightforwardly depend on the grid, the (surface) gradient expression

simply reduces to

$$\left(\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\gamma}}\right)^T = \left[\left(\frac{\partial J}{\partial \boldsymbol{\gamma}}\right)^T + \left(\frac{\partial \mathbf{R}}{\partial \boldsymbol{\gamma}}\right)^T \mathbf{v}\right]. \tag{5.3}$$

where the vector $\boldsymbol{\gamma}$ is herein represented by c, d. As no post-processing operation, i.e. smoothing and projection, is applied to the original gradient, it is easy to validate the accuracy of the adjoint method against some finite difference technique. Centered finite difference formula with appropriate step-size has been utilized for the purpose. The results of the comparison are shown in Table 5.1, where is clearly highlighted an excellent agreement between the two methods.

Gradient	*Adjoint*	*Centered FDM*
$\frac{\mathrm{d}J}{\mathrm{d}c}$	12.50467932	12.54722000
$\frac{\mathrm{d}J}{\mathrm{d}d}$	-4.00182424	-3.992915

Table 5.1. *Validation of shape gradient for quasi-1D flows.*

5.1.3 Shape Optimization Results

The design process is advanced using either a conjugate gradient or a quasi-newton method, both available through the open-source Dakota software [91]. The two algorithms are capable to successfully converge to the local minimum in almost the same number of iterations, see in Figure 5.1 the quasi-newton convergence.

The present test case obviously reveals a very scarce applicative interest and has been inserted in this thesis only for completeness. However, its simplicity represents a key-point when advanced studies on innovative numerical methods involving adjoint are to be carried out. As a matter of fact, the simple model problem has been taken as the reference test case for the uncertainty quantification studies performed to investigate the potential of the adjoint for stochastic predictions. The very low computational demand required by the quasi-1D model enables the use of a full (and very expensive) Monte-Carlo method for comparison purposes. The results are extensively reported in the following chapter.

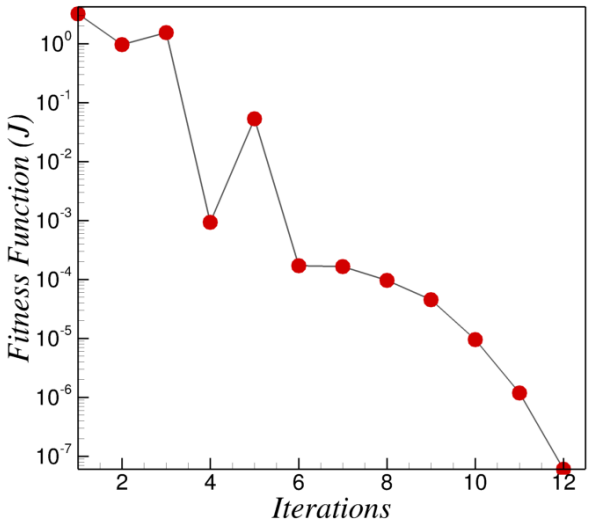


Figure 5.1. *Convergence history for the quasi 1-D pressure matching problem.*

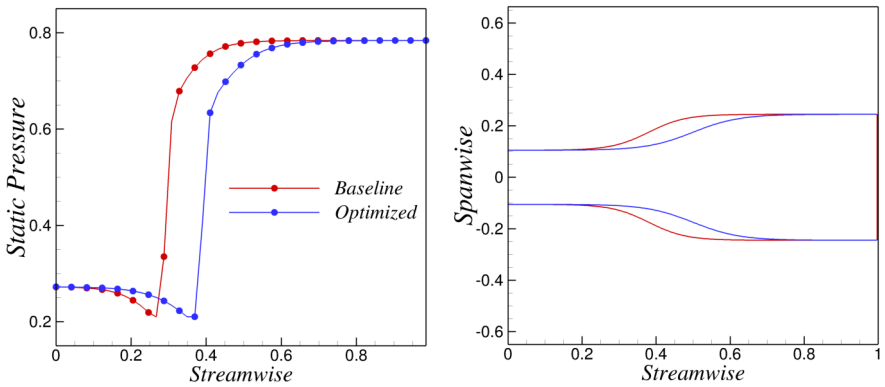


Figure 5.2. **Left:** *Initial and final stream-wise pressure distribution.* **Right:** *Baseline and optimized nozzle geometry.*

5.2 Quasi-Uniform Supersonic Flow through Converging-Diverging Nozzles

5.2.1 Introduction

Gas dynamic subsonic-supersonic nozzles are largely applied in rocket propulsion, turbines, supersonic wind tunnel etc. Supersonic wind tunnel nozzles are characterized by a highly uniform supersonic flow at outlet section and, therefore, are the benchmark for test sections where high quality uniform flow is required. In recent years an innovative test rig, called TROVA, for the experimental investigation of organic vapor streams has been designed and built at Politecnico di Milano and its commissioning is currently underway [35]. The wind tunnel accommodates a supersonic planar nozzle, representative of a supersonic turbine blade passage. The geometry of such nozzles has been preliminarily determined for different fluids and operating conditions by the Method of Characteristics (MoC), see [92]. Often, the MoC method does not permit to satisfy the trade-off between the flow uniformity and the constraint on the maximal nozzle length, which represents a limitation in most wind tunnels. As a matter of fact, the baseline nozzle geometry generated through MoC may be only the initial guess of a more complex design procedure intended to match the nozzle characteristics (e.g. flow uniformity) with the strict requirements, in terms of maximum allowable space, of the wind tunnels. In this context, shape optimization can be of noteworthy aid to improve the design, by ensuring the achievement of a high-quality uniform flow even for nozzles different from those initially suggested by MoC calculation.

5.2.2 Problem Statement

The planar converging-diverging channel herein considered was designed assuming the siloxane *MDM* as working compound. The classical 2D Method of Characteristics (MOC) was originally adopted for the design of the diverging part of the nozzle, while a polynomial functional form was used to construct the converging portion, [92]. For given reference operating conditions and working compound, the MOC-based design generates a nozzle shape capable of providing uniform flow at the outlet section. Therefore, even slightly changes in the fluid-dynamic behavior of the flow or minor geometry modifications usually lead to a rapid degradation of nozzle performances. In this case, the baseline MoC profile was simply shortened to obtain a nozzle length compatible with the space available in the TROVA test section. A spanwise re-definition of the geometry is then mandatory to achieve a high-quality uniform flow.

The design process is carried out by using two different shape optimization

approaches. In the former exercise the nozzle geometry is parameterized by directly using the boundary mesh points, while in the second case the profile is described by NURBS parametric curves. Accordingly, the descent direction towards the local minimum of the function to be optimized is given by, respectively, the *active* gradient and the *surface* gradient. In all presented test cases, the (active) gradient is calculated by resorting to the frozen domain approach.

As pointing out pros & cons of both methods is initially of main concern, the thermo-physical description of the fluid is carried out through the polytropic Van der Waals (PVdW) equation of state (EoS). The use of the PVdW model instead of the Span-Wagner EoS produces quite significant deviations in the prediction of the stream-wise Mach number distribution and enables to considerably limit the computational burdens without affecting the results of the comparison. A more accurate nozzle re-design is finally performed by introducing in the optimization loop the thermodynamic model originally adopted in MoC design. The latter design exercise serves, among others, as a first verification of the feasibility of the real gas adjoint method presented in chapter 3.

In all three examples the computational grid is generated at each design step by the automated Delaunay-based unstructured solver and is composed by about 5000 triangular elements. The baseline mesh layout is shown in Figure 5.3.

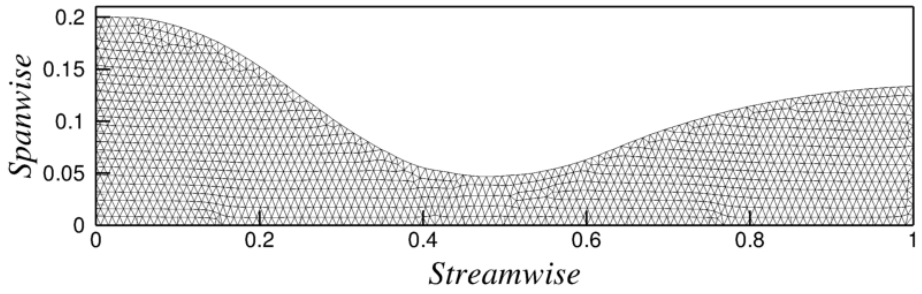


Figure 5.3. *Computational mesh of the converging-diverging channel.*

Total conditions (pressure and temperature) are imposed at inflow boundary, whereas a proper static backpressure value, ensuring a fully supersonic flow, is initially assigned at outflow domain. For the PVdW model, the caloric equation of state is specified through a constant γ, whose value is

preliminarily evaluated by taking the arithmetic mean between the inlet and the outlet section. Thermodynamic conditions at inlet/outlet section are listed in Table 5.2. As the inlet compressibility factor is close to 0.6, relevant real-gas effects occur at least in the first part of the expansion process.

Fluid	P_T	T_T	p_s	γ
MDM	10 bar	276 ℃	1.0 bar	1.0278

Table 5.2. *Total upstream conditions, static outlet pressure and specific heat ratio for shape optimization of quasi-uniform supersonic flows.*

The three test cases are now separately discussed, starting from the nozzle re-design through active points.

5.2.3 Shape Optimization of Converging-Diverging Nozzle using Active Points

The quantitative evaluation of flow uniformity is not a uniquely defined procedure. A quick indication of the level of uniformity can be achieved by measuring, at the outlet section, the dispersion of flow angles of each streamtube around a prescribed mean value. Alternative formulations can be specified in terms of Mach numbers variability or as a combination of both flow angle and Mach number distribution. In the present study the first definition is addressed. The rationale behind this decision is essentially due to the fact that the measurement techniques expected to be used for velocity measurements, as LDV and aerodynamic pressure probes, are highly sensible to the flow direction. Therefore the objective function can be written as

$$J(\mathbf{X}_b) = \left[\sum_{i=1}^{n_{ob}} (\alpha_i - \alpha_{tg})^2\right]^{\frac{1}{2}}. \tag{5.4}$$

In Equation 5.5 α_i is the flow angle at each cell of the outlet boundary, α_{tg} is the target mixed outflow angle (herein equal to 0), and $\mathbf{X}_b$ indicates the vector of the grid points coordinates of the nozzle profile to be optimized. The gradient vector of the fitness function is computed by means of the procedure explained in chapter 3 and subsequently projected along the span direction. In this way the length of the channel is conserved throughout the optimization, without any explicit constraint in the procedure. Both the flow and the adjoint solver are converged to a residual error of the order of 10^{-6}. The optimization parameters, identified after a preliminary study, and the

fitness function values are listed in Table 5.3.

ϵ	λ	*design iterations*	$J_{initial}$	J_{final}
20.0	1.0e-05	20	0.75	0.12

Table 5.3. *Results and main parameters of gradient-based optimization for quasi-uniform supersonic flows using active gradient.*

The objective function trend is reported in Figure 5.4. The convergence of the optimization process is achieved in about 20 design steps, corresponding to a total cost of about 10 minutes on a standard PC (dual-core Opteron node at 2.8 GHz).

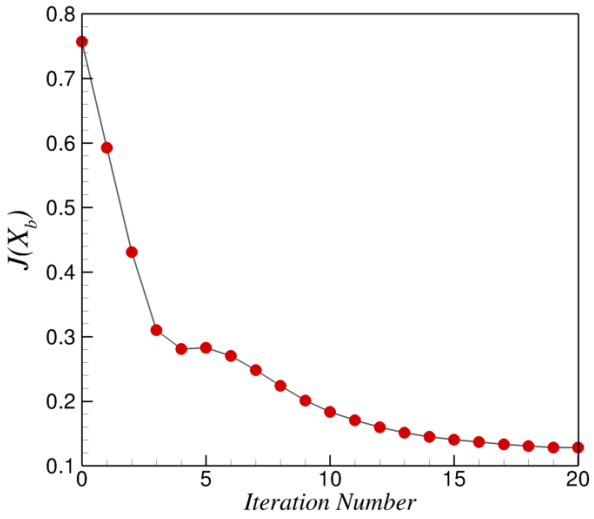


Figure 5.4. *Convergence history of nozzle shape optimization based on active points.*

Figure 5.5 shows a comparison between the flowfield in the initial and optimal configurations in terms of flow angle distribution. The most relevant differences between the initial and final nozzle contours appear visible in the region located just upstream of the outlet section, where the optimized profile results practically aligned with the prescribed mean direction. By virtue of this improvement, the flow passage area, in the final part, increases less

rapidly than in the initial configuration, thus reducing the spanwise peaks of flow angle observed at baseline outlet section, see 5.6. For both geometries, the highest values of the flow angle still yield downstream of the throat in the small portion close to the solid boundary.

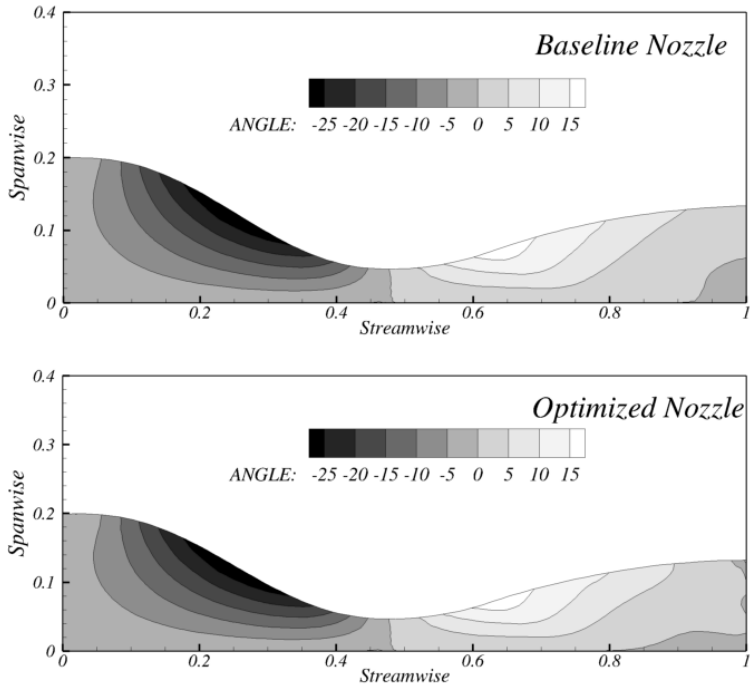


Figure 5.5. *Initial and optimized flow fields of the converging-diverging nozzle.*

The trends of the gradient modules are sketched in Figure 5.7 for the baseline and optimized configuration. In this figure the effect of elliptic smoothing is clearly pointed out; the smoothing operator is able to reduce the oscillations appearing in the original gradient. Notice that the modules of the smoothed gradient vectors in the optimized configuration (blue line) are of orders of magnitude smaller than in the baseline one (green line), as the optimization process approached the local minimum. However, for nozzle shape optimization, the evolution of the optimization problem is stopped when the relative difference of the fitness function value at two consecutive iterations is less than a fixed tolerance.

Interestingly, the gradients values are almost null along the converging part of the nozzle. This is due to the fact that the flow uniformity at the outlet section of the nozzle is much more sensitive to the shape of the di-

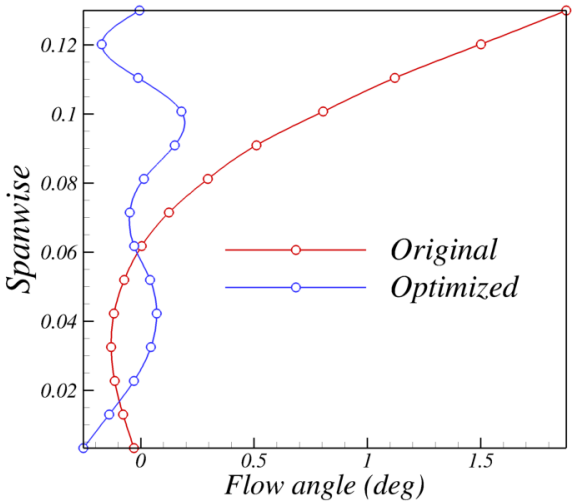


Figure 5.6. *Comparison of the spanwise flow angle distribution at nozzle outlet section between the baseline and optimized configurations.*

verging part than to that of the converging part. As expected, the gradient components become more relevant in proximity of the nozzle outlet section, indicating that the flow angle is mostly affected by the shape of the final part of the diverging portion. In this perspective the adjoint-based gradient calculation can be regarded also as a method to preliminarily analyze the influence of each active grid point on the objective function value, and this information is extremely useful in the context of a design methodology.

5.2.4 Shape Optimization of Converging-Diverging Nozzle using Surface Gradient

The same design problem of the previous section is now solved by defining the nozzle geometry through NURBS parametric curves. The profile is constructed by means of two NURBS curves, the former describing the converging portion and the latter approximating the diverging one, for a total amount of 25 control points. The control points distribution and the related nozzle profile are shown, properly scaled, in Figure 5.8.

The main difference against the previous case is that the corresponding fitness function is now minimized by directly acting on the position of the control points. Therefore the objective is re-formulated as

$$J(\mathbf{P}) = \left[\sum_{i=1}^{n_{ob}} (\alpha_i - \alpha_{tg})^2\right]^{\frac{1}{2}}. \tag{5.5}$$

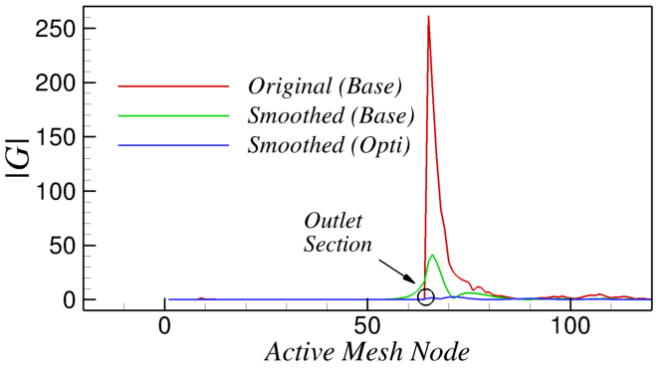


Figure 5.7. *Streamwise distributions of the gradient vector modules for the baseline (original and smoothed) and optimized (smoothed) nozzle configuration. Notice that the node enumeration does not respect the actual streamwise location along the nozzle abscissa.*

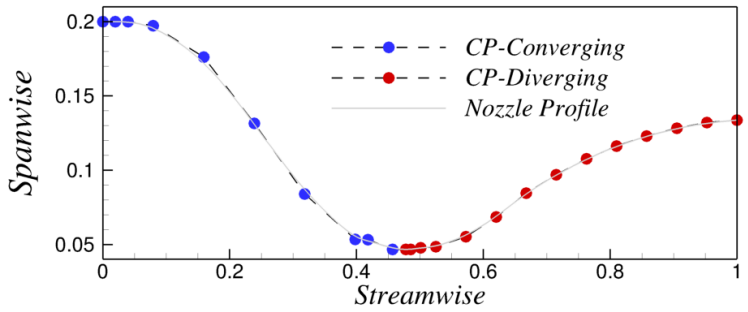


Figure 5.8. *Control point distribution and profile of the converging-diverging nozzle. Notice that x, y scales are independent.*

The design process is converged by using the same optimization parameters of the preceding application. The active gradient is firstly smoothed and projected onto the spanwise direction, then is further processed for the computation of the surface sensitivities. The smoothing and projection operations at mesh level ensure the preservation of the nozzle axial length and drastically tackle the probability of failure of the design process. At each iteration, the spatial displacement of the NURBS control points are computed according to the step-length λ listed in Table 5.4.

It should be noted that acting on a single control point implies the contemporary displacement of several mesh nodes. As a consequence, the surface gradient is typically of higher smoothness class than the active sensitivity vector. Hence the optimization process, if well posed, converges to feasible

ϵ	λ	*design iterations*	$J_{initial}$	J_{final}
20.0	1.0e-06	10	0.70	0.14

Table 5.4. *Results and main parameters of gradient-based optimization for quasi-uniform supersonic flows using surface gradient.*

geometries more easily than through CAD-free methods. A certain increase of the convergence rate is also usually to be expected, however establishing generalized convergence rules for both methods remains a complex challenge and is anyhow beyond the scope of the present study.

The convergence of the optimization process is herein achieved in about 10 design steps, see Figure 5.9, hence approximately halving the computational cost of the previous case.

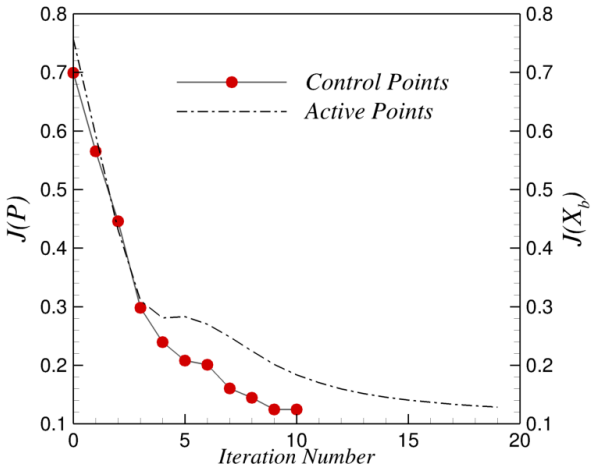


Figure 5.9. *Convergence history of nozzle shape optimization based on NURBS control points compared to descent trend based on active points.*

Similar considerations to the previous case can be traced regarding the flow angle distribution along the stream-wise nozzle direction. However, despite the fitness function values are almost the same in the optimal point, the NURBS-based design process ensures a more regular spanwise distribution of the flow angle at nozzle outlet section, see Figure 5.11. This is mainly due to the fact that the residual noise present in the active gradient is further reduced when processed to compute the surface sensitivity vector, thus

the full shape design results smoother. As a consequence the use of surface sensitivities, if available, is highly recommended in shape optimization problems. An interesting alternative strategy, not investigated in this work, could rely on the hierarchical integration between the two methods. Starting from the NURBS-based optimal configuration, the CAD-free approach could be applied to locally refine the nozzle shape. In fact, CAD-free techniques are potentially able to successfully operate on geometry details, which conversely are problematic to be captured by using a lower-dimension parameterization.

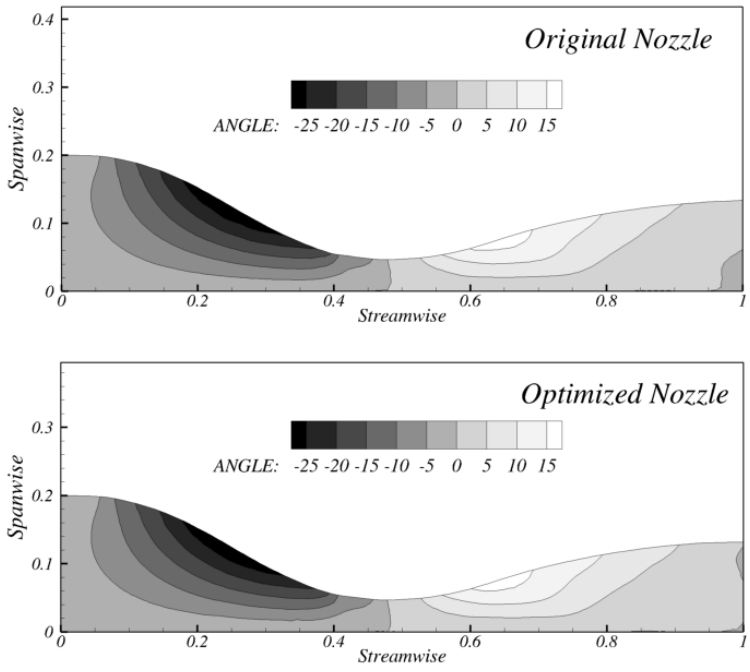


Figure 5.10. *Initial and optimized flow angle distribution using NURBS parametric curves.*

5.2.5 Shape Optimization of Converging-Diverging Nozzle using Surface Gradient and Accurate Equations of State

The previous shape optimization test case is now performed with the use of a more complex equation of state. The fluid thermo-physical behavior is herein accurately predicted through the Span-Wagner (SW) EoS, introduced at both flow and adjoint level by means of the look-up table approach presented in chapter 4. As already mentioned, the LUT algorithm is fully

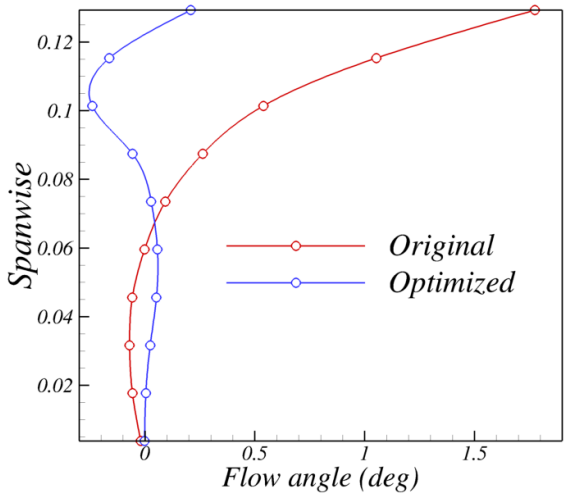


Figure 5.11. *Comparison of the spanwise flow angle distribution at nozzle outlet section between the baseline and optimized configurations.*

ϵ	λ	*design iterations*	$J_{initial}$	J_{final}
0.0	1.0e-06	10	0.885	0.081

Table 5.5. *Results and main parameters of gradient-based optimization for quasi-uniform flows using surface gradient and accurate equations of state.*

differentiated within the adjoint solver. Preliminary flow calculations were conducted to find a suitable extension of the thermodynamic mesh ensuring a high robustness of the design process. A final thermodynamic mesh of about 20000 elements is used for the calculations.

The nozzle geometry is parameterized according to the same control points distribution shown in Figure 5.8. A reasonable compromise between stability and efficiency of the design procedure has been achieved by employing the same step-size λ of the preceding case. To investigate the intrinsic smoothing properties of the NURBS parameterization the value of ϵ is herein nullified. The setup and the final results of the optimization are summarized in Table 5.5.

The initial and the optimized flowfields in terms of flow angle are reported in Figure 5.12. Again, the design process enforces a better alignment of the final part of the nozzle profile to the prescribed mean angle, leading to a considerable regularization of the flow angle distribution at outlet section, as depicted in Figure 5.14. As expected, geometry parameterization may be regarded as an intrinsic source of smoothing. Consequently, the smoothness

of the design can be preserved even without the use of the elliptic technique. This may lead also to a considerable speed-up of the overall process. The convergence rate of the design process is shown in 5.13.

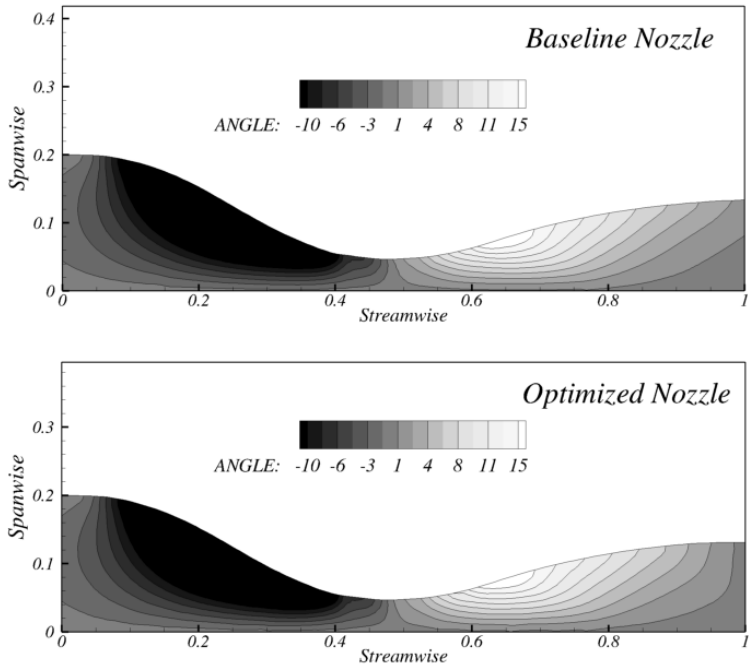


Figure 5.12. *Initial and optimized flow angle distribution using NURBS parametric curves and SW-LUT thermodynamic model.*

Summarizing, the outcomes achieved in the three test cases essentially demonstrate that *i*) the adjoint method developed is particularly robust and reasonably fast *ii*) the success of the optimization is ensured for a wide range of step-sizes and for gradients computed by both frozen domain and radial basis function approaches. *iii*) the implementation of a differentiated LUT algorithm into the adjoint solver guarantees a high flexibility, leaving unaffected the possibility of straightforwardly exploiting *state-of-the-art* thermodynamic models embedded in external tools.

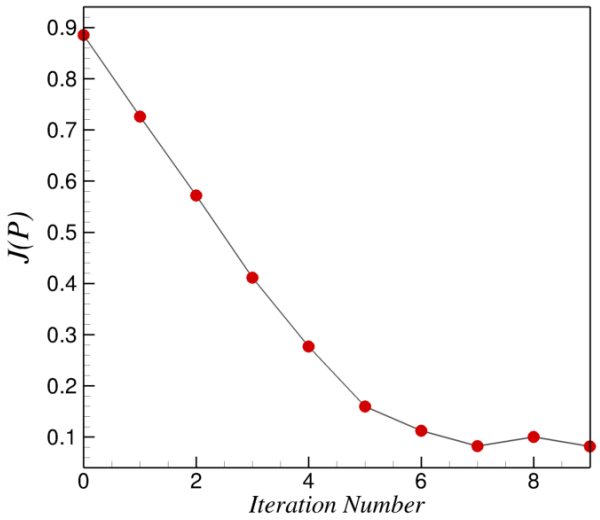


Figure 5.13. *Convergence history of nozzle shape optimization based on NURBS control points and accurate equations of state.*

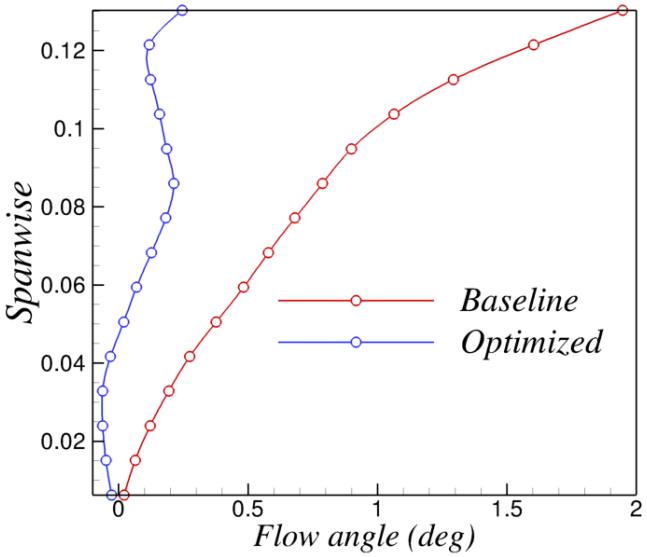


Figure 5.14. *Comparison of the spanwise flow angle distribution at nozzle outlet section between the baseline and optimized configurations.*

5.3 Performance Improvement of a Transonic Turbine Cascade

The capability of the current adjoint method in dealing with different fluid-dynamic problems is assessed in this section, reporting the maximization of the performances of a transonic turbine cascade. The design exercise herein discussed represents one of the seminal applications of the present research work, whereby no parameterization technique was still introduced in the optimization loop. Therefore the design process is proceeded by using the active grid points, i.e. the nodes of the blade profile, as design variables.

The original VKI LS-59 cascade is chosen as the reference geometry to optimize. The grid nodes forming the blade shape are moved in order to maximize the cascade performance, expressed in terms of isentropic efficiency. As a result the fitness function is given by

$$J(\mathbf{X}_b) = \frac{h_1 - h_{2_{ob}}}{h_1 - h_{2_{is}}}, \tag{5.6}$$

where the inlet/outlet static enthalpies h are computed by mass-averaging the specific properties at the outflow boundary. Total conditions are imposed upstream of the blade, while a static back-pressure corresponding to $M_{is} = 1$ is prescribed at the outlet section. With these boundary conditions a normal shock appears on the final part of the suction side, just before the trailing-edge. For such flow configuration the turbine loss mechanism is dominated by the shock losses, consequently the use of the Euler physical model can be considered as an affordable approximation. The effect of the numerical viscosity is approximately the same at each design step (thanks to the very slight change of the grid topology as the design evolves), hence, any improvement of the fitness function value results from the reduction of shocks strength. The grid is composed of about 8000 triangular elements and convergence tolerances of the order of 10^{-6} are used for both flow and adjoint (i.e. FGMRES threshold value) solvers. Several initial calculations have been carried out to verify the independence of the solution from the grid spacing.
The gradient of the fitness function is smoothed and projected onto the tangential direction in order to leave the length of the blade axial chord unchanged. Differently from the previous case, a non-uniform step length strategy is introduced to compute the displacement vectors. This latter technique has been adopted after that preliminary optimization tests, with uniform λ for all active nodes, led to unacceptable deformations of the trailing-edge; for this reason the λ parameter has been modified in proximity of the leading and trailing-edge zones by applying both the Heaviside and the Gaussian-

ϵ	λ	*design iterations*	$J_{initial}$	J_{final}
10.0	-0.02	53	94.3 %	95.7 %

Table 5.6. *Results and main parameters of gradient-based optimization for the VKI LS-59 transonic cascade.*

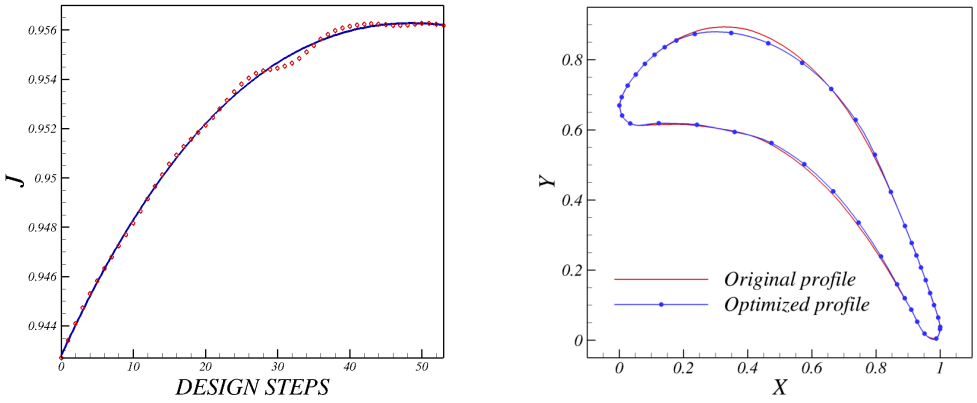
like correction scheme, as depicted by the circles positioned in Figure 5.16. The best results have been obtained by using the Heaviside approach and are therefore reported in the following. To prevent unsuitable blade shapes in the context of an unconstrained optimization, the design cycle is stopped if the ratio $\frac{|A_i - A_0|}{A_0} < 0.05$, where A_i is the blade area at a given iteration while A_0 represents the initial blade area. As a consequence the optimal blade maintains a mechanical resistance similar to the one of the baseline geometry.

The convergence is approximately reached after about 50 design steps, as plotted in Figure 5.15a, corresponding to a total computational time of about 30 minutes on the same machine previously mentioned. The final blade area reduces of -3%, thus remaining within the acceptable range.

Figure 5.15b, which compares the initial and optimal blade profiles, shows that the optimization leads to a reduction of blade curvature on the blade suction side and to a more rectilinear rear suction side. This is consistent with the distribution of displacement vectors at the beginning of the design process, reported - with proper scaling - in the left frame of Figure 5.16. The considerably lower magnitude of the displacement vectors in the optimal configuration, shown in the right frame of Figure 5.16, clearly indicates the convergence of the design procedure.

The main parameters and the final results of the optimization procedure are summarized in Table 5.6. Preliminary numerical tests have been performed to select the proper values for optimization and smoothing parameters. However, the best compromise between computational efficiency and stability, in terms of geometry deformations, has been achieved by selecting values for ϵ and λ of the same order of magnitude as those used in the nozzle case, (with negative values of λ required for maximization problems).

Figure 5.17 depicts the resulting pressure fields of the initial and optimized configuration. Even though the general flow features are almost the same throughout the optimization process, some relevant differences can be noticed. As previously discussed, the loss generation is here due to the shocks appearing on the rear part of both the pressure and the suction side, with the latter much stronger than the former. Thanks to the blade re-design the shock appearing on the rear suction side of the optimized layout exhibits a



(a) (b)

Figure 5.15. *Convergence history of the VKI LS59 optimization process (a) and sketch of both original and optimized blade shape contours (b).*

weaker intensity than that occurring in the initial configuration; the severe pressure gradients just upstream of the trailing edge on the rear pressure side are also significantly weakened, resulting in an attenuation of the shock wave in this region. These features can be fully appreciated in Figure 5.18, which shows the pressure distributions on the suction and pressure side of the blade. In particular, the optimal blade shape induces a delay in the expansion on the suction side before the throat, and a higher acceleration on the suction side just downstream of the throat. This eventually leads to a reduction of post-expansion just upstream of the trailing edge, and hence to a decrease in shock strength; as a result shock losses are weakened, and the isentropic efficiency increases of about 1.5%.

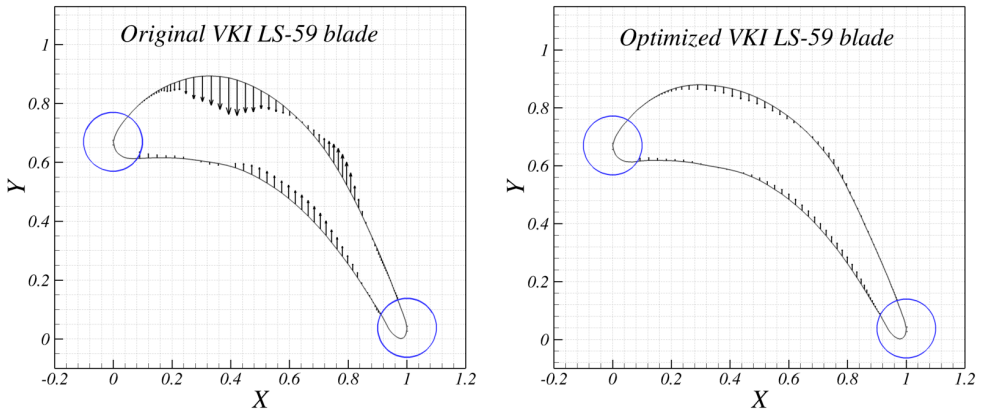


Figure 5.16. *Initial and optimized contours of the VKI-LS59 blade. The spatial displacements assigned to the boundary nodes at first and final iteration are also reported.*

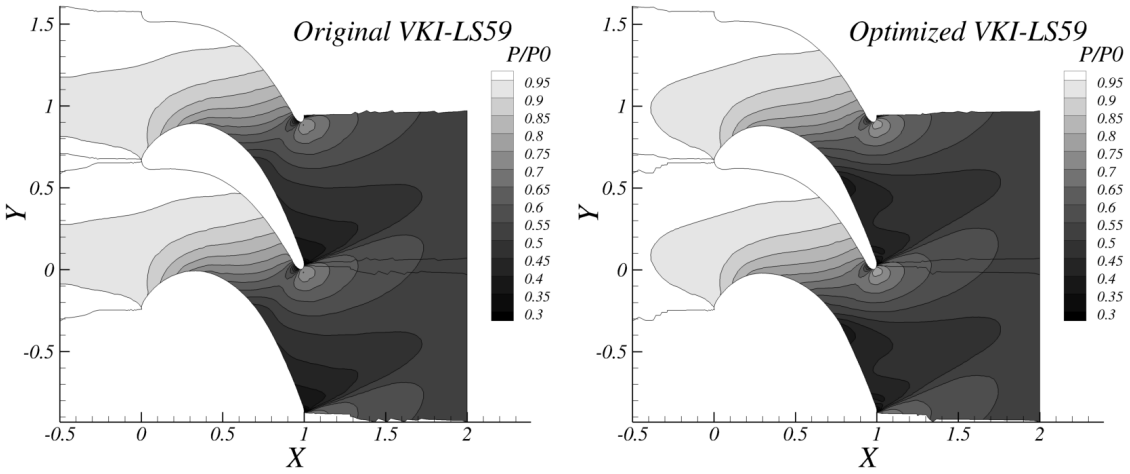


Figure 5.17. *Initial and optimized pressure fields of the VKI-LS59 cascade.*

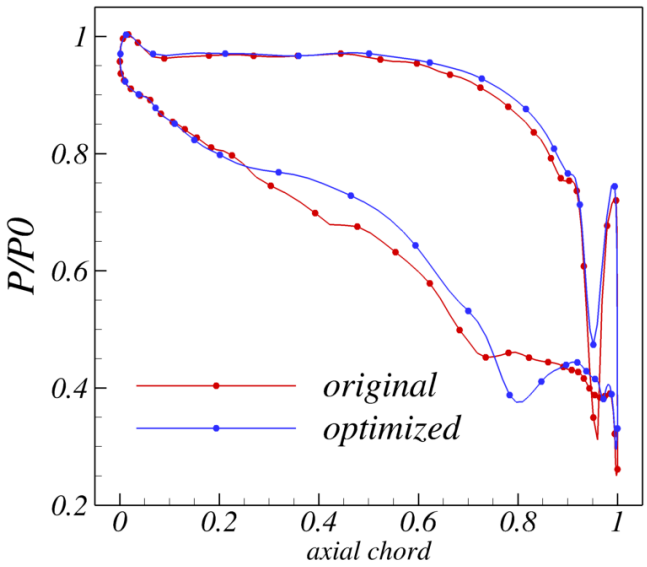


Figure 5.18. *Initial and optimized blade loading of the VKI-LS59 cascade.*

5.4 Performance Improvement of a Supersonic ORC Turbine Cascade

The design algorithm sketched in Figure 3.3 is now applied to the re-design of the profile of a converging-diverging turbine cascade operating under supersonic conditions. Radial basis function approach is used for computing active gradients. The blade geometry, shown in Figure 5.19, has been originally designed by means of the method of characteristics (MOC) for the diverging part and constructing a highly smooth leading-edge portion for ensuring acceptable performance in case of relevant variations of the incidence angle.

Two test cases are considered, a first one in which the cascade performance are maximized at reference conditions and a second one conversely finalized to obtain a blade profile less sensitive to the fluctuations of the operating conditions at off-design. As a matter of fact, supersonic cascade performance rapidly decays at off-design conditions due to the onset of strong shock waves in the post-expansion region downstream of the blade outlet section. This is, for instance, the case of supersonic turbines installed in small-size power plants, as in [31], whereby the operating scenario is subject to continuous oscillations. As a result, more robust design strategies specifically targeted to supersonic cascades appear a promising alternative compared to single-point design procedures.

5.4.1 Geometry Construction

Each side of the turbine blade is defined by a set of 11 control points, see Figure 5.19. The position of the NURBS control points is found as follows: initially, the blade camberline is defined by fixing the inlet, outlet and stagger blade angles. Once the reference line has been generated, it is conveniently divided into a number of intervals, which can be equally spaced or with a variable length. Starting from these points, arbitrary normal distances from the corresponding construction line determine the position of the set of inner control points. Therefore the position of each inner control point is iteratively changed until minimizing the distance between the reference profile and the profile generated by NURBS curves. The procedure is repeated for both suction and pressure side.

5.4.2 Simulation Setup

In this work inviscid governing equations are considered, the choice being appropriate for operating conditions in which shock waves represent the major source of losses, as commonly occurs in supersonic cascades. As the final purpose is the optimization of plane profiles, 2D blade-to-blade flow is con-

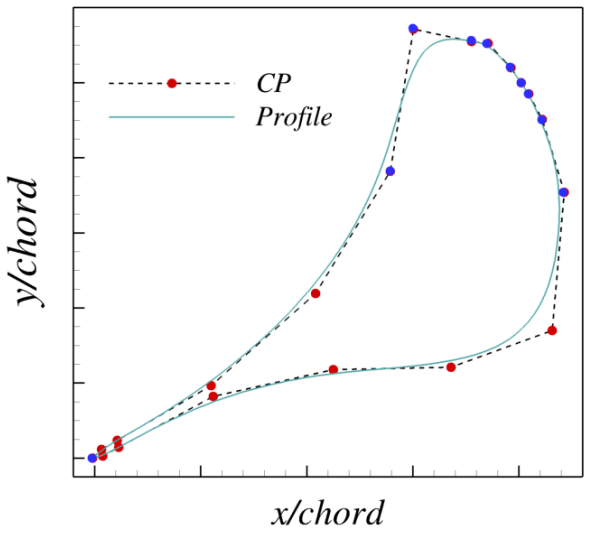


Figure 5.19. *Profile contour and control points distribution of the converging-diverging blade. Red points indicate the design variables, whereas red dots are fixed during optimization.*

sidered. Inflow conditions are computed starting from total pressure and temperature, while non-reflecting boundary conditions (Giles) are applied at the domain outflow in order to capture the realistic flow pattern. A preliminary analysis has been conducted to assess the independence of the solution from the grid spacing, leading to final grid size of about 20000 triangular elements. The working compound is the siloxane MDM, modeled by means of the polytropic Van der Waals equation of state. As relevant real gas thermodynamic effects occur, polytropic ideal gas law is unsuitable for such a problem.

5.4.3 Single-Point Optimization

The first test case aims at re-designing the supersonic blade cascade of Figure 5.19 operating under reference conditions. Thermodynamic conditions at design point are listed in Table 5.10.

Fluid	P_T	T_T	p_s	γ
MDM	8 bar	272 ℃	1.0 bar	1.0214

Table 5.7. *Upstream total conditions, static outlet pressure and specific heat ratio for shape optimization at reference condition.*

The predicted flow-field at reference conditions, shown in the left frame of Figure 5.20 in terms of Mach number, highlights a strong shock generated on the blade rear suction side.

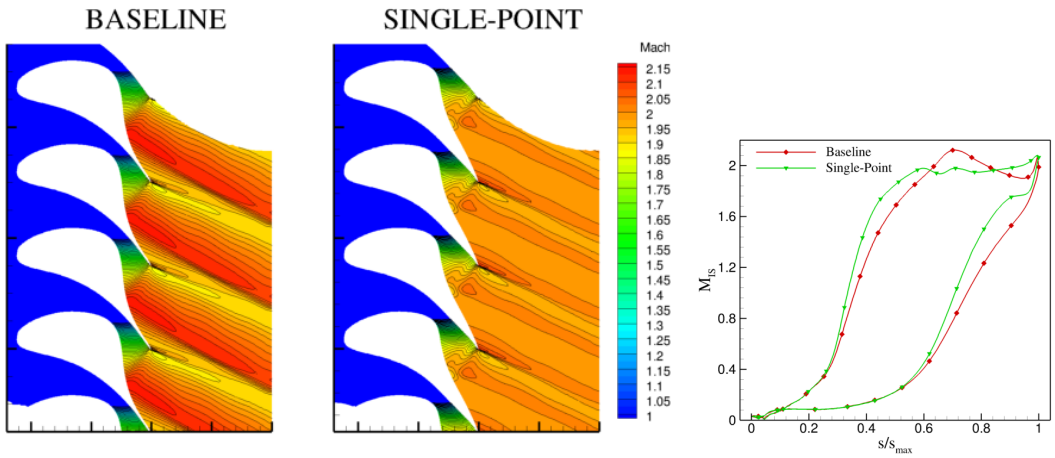


Figure 5.20. **Left:** *Predicted Mach flow-field using the baseline and optimized cascade.* **Right:** *Comparison of blade loading between the baseline and optimized cascade.*

The resulting effect is a non-uniform flow at the blade outlet section which is highly detrimental for the efficiency of the cascade and of the whole stage. Therefore, a significant objective of shape optimization is the achievement of a more uniform flow in terms of Mach number pitch-wise distribution at the outlet bound of the computational domain. Furthermore, achieving a more uniform flow necessarily leads to a weakening of the shock intensities, and therefore to a considerable reduction of the total pressure losses which is then expected as a further outcome of the design process. Moreover, in case of turbulent flows, shock waves/boundary layer interactions are significantly attenuated, reducing the risk of boundary layer separations. Also, a more uniform flow leaving the stator contributes to improve the efficiency of the subsequent rotor and, hence, of the full turbine stage. The cost functional to be minimized is therefore written as

$$J(\mathbf{P}) = \left[\frac{\sum_{i=1}^{n_{ob}} (M_i - M_{mix})^2}{n_{ob}} \right]^{\frac{1}{2}}, \tag{5.7}$$

where M is the Mach number at each boundary node, n_{ob} the number of outflow boundary nodes, and subscript mix indicates the pitch-wise mixed-

out averaged value.

As a preliminary step, the gradient of the objective, computed at the first iteration of the design process and provided in the right frame of Figure 5.21 properly smoothed and projected, has been analyzed. Notice, firstly, that a high gradient regularization of the original gradient, see left frame of Figure 5.21, is due to both elliptic smoothing coupled to a non-uniform step-length strategy applied in the region close to the trailing-edge. This initial operation can be regarded as a very useful strategy to better understand how the design process will proceed and, possibly, to decide which design variables can be removed from the optimization. As expected, the gradient distribution depicted in Figure 5.21 shows that the value of the objective J is strongly influenced by the blade region located downstream of the throat, while a minor contribution is due to the upstream part. As a matter of fact, in supersonic cascades the physical features of the shock waves stemming in the turning region are greatly affected by the shape of the semi-bladed channel. Therefore, in the present study, the control points drawn with blue circles in Figure 5.19 are kept fixed during the design process.

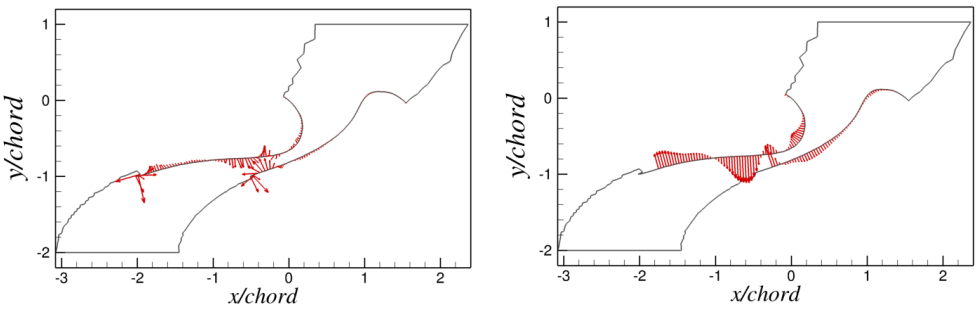


Figure 5.21. **Left:** *Original gradient components at first iteration.* **Right:** *Smoothed and projected gradient components at first iteration.*

The convergence of the optimization procedure is achieved in about 12 iterations, requiring a total computational burden of about 20 minutes on a standard PC (Opteron dual-core working station). The step-length λ and smoothing parameters ϵ were set to -0.05 and 30, respectively, after some trial and error operations. However, the design process displayed a relatively high robustness compared to significant variations of both parameters (mostly ϵ). Less smooth and slightly faster optimizations are achieved by reducing the ϵ value, whereas the final solution is found to be very similar in all cases. The final results of the simulation are reported in Table 5.8.

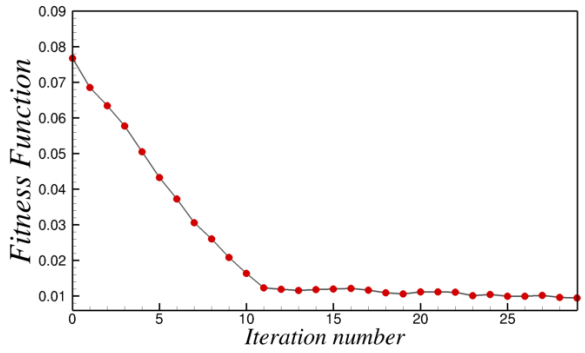


Figure 5.22. *Convergence history of the single-point optimization.*

Blade	$J(\mathbf{P})$	Y	m_{flow}	α_{flow}
Baseline	0.078	9.9 %	1	76.29 deg
Single-Point	0.011	2.8 %	0.977	76.94 deg

Table 5.8. *Results of shape optimization at reference conditions. Mass flow rates are reported in non-dimensional form. Total pressure loss coefficient Y is defined as* $\frac{P_{T0}-P_{T1}}{P_{T0}-p_{s1}}$.

The optimized blade shape, depicted in the right frame of left Figure 5.20, evidences a more straight rear suction side with respect to the initial configuration, essentially in the semi-bladed region. The differences between the two blade shapes can be better appreciated in the left Figure 5.20, where the optimized shape is labeled as Single-Point. The re-design of the rear passage geometry entails significant changes of the flow behavior inside both bladed and turning (post-expansion) regions. It should be noted that the optimization process leaves almost unaffected the size of the throat and enlarges the blade outlet section, thus ensuring a very similar mass flow rate between the two configurations. Hence the diverging portion of the optimized configuration results characterized by higher passage area ratios. As a consequence the fluid stream is subject to larger accelerations on both pressure and suction sides of the diverging channel, reaching higher outlet Mach numbers. The flow is not further accelerated at the exit of the bladed region due to the rectilinear rear suction profile. The reduction of the suction side curvature enables in fact to avoid the over-acceleration present in the original flow pattern, that are the cause of the compression fan formation just before the trailing-edge of the baseline cascade. As depicted in Figure 5.20 right, the concave rear shape of the baseline suction side provides an isentropic deceleration of the flow but the conjunction of the characteristic

waves of the fan promotes the growth of a strong shock in the mixing zone. The shock is instead completely removed in the optimized configuration and the efficiency decay remains only related to the fishtail shock detaching from the trailing-edge.

A more uniform flow is therefore finally achieved in the downstream region, leading to a relevant decrease of the total pressure loss coefficient, which passes from about 10% to 2.8% for the optimal configuration.

Notice also that, despite the unconstrained design process leads to a more stocky blade, a very slight reduction of the mass flow rate through the turbine occurs (about 2.3%). However, this variation may be adjusted by increasing the blade height of the same order or recurring to constrained design algorithms which are anyhow beyond the scope of the present work.

5.4.4 Multi-Point Optimization

As previously underlined, CFD-based optimization approaches are widely recognized as a standard to improve the performance of baseline turbomachinery configurations. However, most of the published research works still focus on single-point optimization and the effect of the variation of the turbine operating conditions is rarely taken into account in the design procedure. Two different design methodologies are able to deal with multiple values of operating conditions (i.e. total inlet conditions, static backpressure): multi-point optimization and stochastic (robust) optimization. The second approach solves the optimization problem in a fully stochastic framework, then may suitably handle the aleatory trend of the turbine boundary conditions. Conversely the first method embeds in a single objective deterministic design problem several non-concurrent contributions, one for each operating condition, through a weighted-sum approach. However, to the knowledge of the author, a comprehensive comparison between the two methods is not still available in the open literature and only few works analyze the potential of the former approach, see e.g. [67]. In the following the potential of the multi-point approach in dealing with fluctuating operating conditions is investigated. The final objective is the design of a blade profile guaranteeing improved and more robust performance than those offered by both the baseline and single-point blade shapes at off-design. From a probabilistic perspective the aim of a multi-point optimization procedure is the minimization of the mean of the fitness function $J(\mathbf{P})$ as well as its standard deviation.

In power plants, power control is generally managed by acting on the inlet total pressure through the throttling valves positioned before the turbine entrance. However, preliminary studies carried out in this work indicated that a relatively higher variability of the cascade performance is obtained by changing the static backpressure. By following the two-step approach proposed in

[93], an analysis of variance (ANOVA) is initially performed taking total inlet conditions, static backpressure, and inlet flow angle as uncertain parameters with uniform distribution. The scope of such analysis is to quantitatively estimate, in terms of Sobol's indexes [94], the impact of any single stochastic input on of the objective function $J(\boldsymbol{P})$ value, namely how the variability of a single operating condition affects the turbine performances. The range of variation of the variables are indicated in Table 5.9. A third-order tensorial stochastic collocation method is exploited for the analysis.

Variable	P_T	T_T	p_s	$\alpha_{flow_{in}}$
Interval	7.6-8.4 bar	268-276 ℃	$\sim$ 1.0-2.0 bar	$\pm$ 10 deg

Table 5.9. *Variability of the uncertain inputs for the ANOVA analysis.*

The results of the ANOVA analysis are reported in the histogram of Figure 5.23, where the bars represent the values of the Sobol's indexes. As clearly highlighted, the fluctuations of the turbine performances are primarily related to the uncertainty of static backpressure, the corresponding index being close to unity. Marginal interaction effects among the stochastic variables are also present.

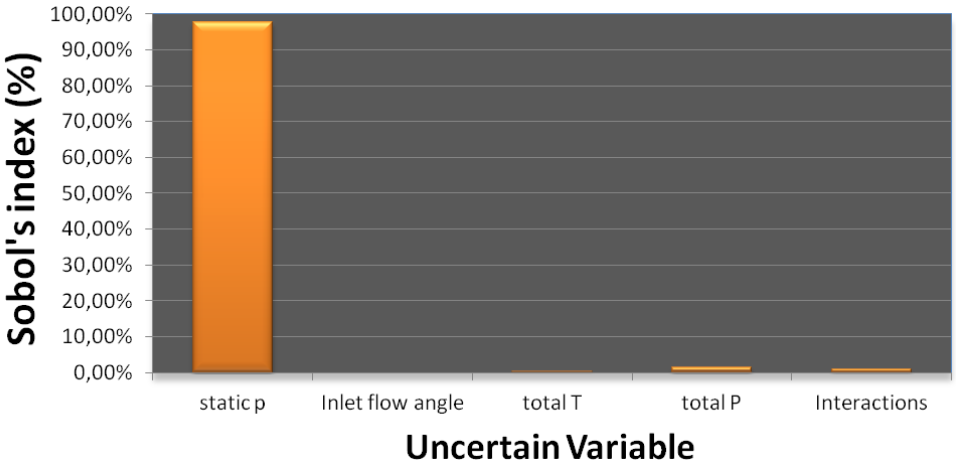


Figure 5.23. *Results of preliminary ANOVA analysis.*

Therefore, in the following off-design study, the total conditions are kept at nominal values, while the static backpressure is left to vary between about 1.0-2.0 bar according to a uniform distribution. As underlined by [95], the main shortcoming of multi-point optimization remains the selection of appropriate design points and their associated weights. To address the evaluation of such points a novel approach, based on an uncertainty quantification algorithm, is proposed. Precisely, a stochastic collocation [96] method of third order is initially applied to determine the quadrature nodes corresponding to the three off-design conditions of the multi-point optimization. Uncertainty quantification methods based on stochastic series, as stochastic collocation or polynomial chaos, estimate such values by exploiting sampling techniques (e.g. latin-hypercube) drawing samples on the basis of the specified statistics of the input parameters. As a result, for uniform probability density functions, the algorithm uniformly distributes the points over the range of possible operating conditions. The order of quadrature has been chosen as a trade-off between the computational cost of the optimization (more points imply more flow/adjoint simulations) and the minimum amount of evaluations needed for accurately predicting first and second order statistical moments, see [93]. In fact, it has been demonstrated by some authors, see for example [97], that quadrature formulas of order n accurately predict statistical moments till to order $n-1$. Finally the weights of the cost function are proportional to the probability density function (PDF) of the uncertain parameter, that are, for uniform distribution, all equal to 1.0 if normalized. In practice, the methodology herein proposed estimates the expectancy $\bar{J}$ of the fitness function through a three-point quadrature formula as follows:

$$\bar{J}(\mathbf{P}, p) = \int J(\mathbf{P}) f_p(p) dp = \sum_{j=p_1}^{p_3} w_j J(\mathbf{P}) \mid_{p_j}, \tag{5.8}$$

where f_p is the PDF of static backpressure.

Fluid	P_T	T_T	p_s	γ
MDM	8 bar	272 ℃	$\sim$ 1.0-2.0 bar	1.0214

Table 5.10. *Total upstream conditions, static outlet pressure and specific heat ratio for shape optimization at off-design.*

The functional to be minimized is therefore $\bar{J}$, written as a sum of three

non-concurrent contributions in the form

$$\bar{J}(\mathbf{P}) = J(\mathbf{P}) \mid_{p_1} + J(\mathbf{P}) \mid_{p_2} + J(\mathbf{P}) \mid_{p_3} \tag{5.9}$$

The values of p_1, p_2, and p_3 corresponding to the quadrature nodes are respectively 2.0, 1.52, and 1.13 bar. The fitness function gradient can be written in a similar fashion as

$$\nabla\bar{J}(\mathbf{P}) = \nabla J(\mathbf{P}) \mid_{p_1} + \nabla J(\mathbf{P}) \mid_{p_2} + \nabla J(\mathbf{P}) \mid_{p_3} \tag{5.10}$$

The design algorithm shown in Figure 3.4 has been used to account for multiple evaluations of the flow and adjoint solution. The optimization process is advanced using the steepest descent method with the same control parameters and reaches the convergence in about 10 iterations, see Figure 5.24. The algorithm is capable of significantly reduce the fitness function value (0.10 compared to the initial value of 0.26) in about one hour on the same machine.

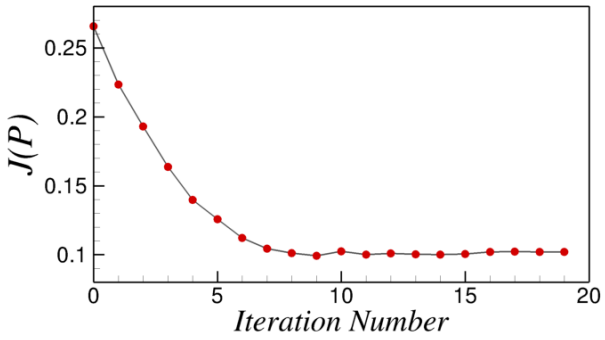


Figure 5.24. *Convergence history of the multi-point optimization.*

5.4.5 Assessment of Performance of Multi-Point Blade

The off-design performance of the baseline, single-point, and multi-point configurations are finally investigated in this section. The efficiency of the cascade over the range of variation of the static backpressure is quantitatively evaluated by means of the uncertainty quantification algorithm previously mentioned. The aim is to synthetically summarize the performances through statistical parameters, i.e. by using mean and standard deviation. For accurately predicting the statistical moments of the objective and the total loss coefficient, a four order stochastic collocation method is adopted. Both

the objective and the total loss coefficient values are initially determined onto four quadrature points (off-design conditions) uniformly spread over the range of possible operating conditions. Then the results are post-processed by the UQ solver to extract the mean and the standard deviation of the merit functions.

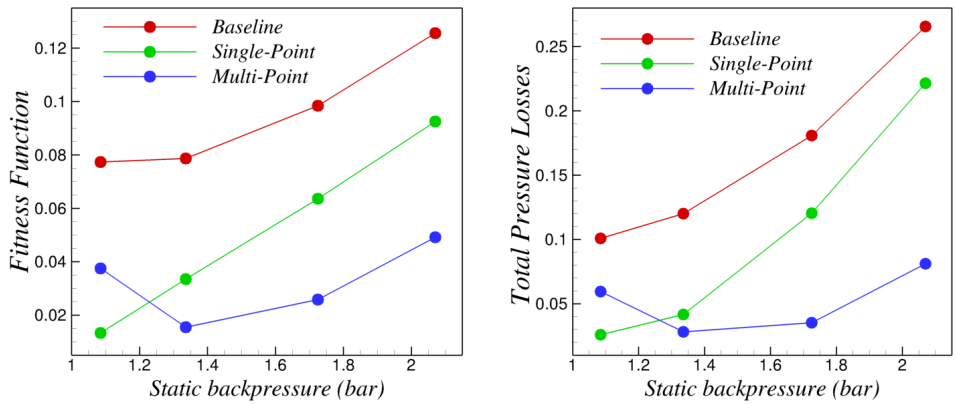


Figure 5.25. **Left:** *Fitness function trend at off-design.* **Right:** *Total pressure losses trend at off-design.*

Figure 5.25 reports the trends of the cost functional shown in equation 5.9 and Y for different values of static backpressure contained in the previous interval. Notice that the geometry designed in the previous section, labeled as Single-Point in the pictures, outperforms the baseline blade configuration for the whole range of outflow pressure. However the trends exhibit a greater variability with respect to those obtained using the baseline cascade. This is confirmed by comparing the mean and the standard deviation of J and Y obtained through the UQ algorithm. The values are reported in Table 5.11, where it well appears how the single-point technique is unsuitable to achieve robust designs, even though the mean performance of the turbine stator may result enhanced compared to the baseline case.

Multi-point methodology, conversely, satisfies both aspects of the optimization. The cascade efficiency is significantly improved in terms of mean and standard deviation with respect to both baseline and single-point assemblies. The robustness of the configuration becomes therefore greatly enhanced over a wide range of possible operating conditions. As expected, single-point optimization performs better at nominal conditions compared to multi-point approach, but performance quickly degrades at off-design. The

	Baseline	*Single-Point*	*Multi-Point*
$\bar{J}$	9.10e-02	4.65e-02	2.78e-02
σ_J	1.62e-02	2.66e-02	1.19e-02
$\bar{y}$	1.69e-01	1.14e-01	5.75e-02
σ_y	5.36e-02	6.67e-02	1.91e-02

Table 5.11. *Predicted statistical moments of the fitness function and total pressure loss coefficient for the baseline, single-point, and multi-point profiles.*

three blade shapes are sketched in Figure 5.27. Multi-point and single-point profiles have a very similar straight rear suction side, which corresponds approximatively the same outlet blade section. The major shape differences are located on the rear pressure side, where multi-point profile is practically superimposed on the baseline one.

As already discussed in single-point optimization, the straight rear suction side induces a limited over-speed in the semi-bladed region and prevents the formation of any compression fan afterwards. The advantages provided to the rear straight contour have been found to be highly beneficial in operating conditions around the nominal point for the single-point profile and they are also proved for the multi-point geometry. As highlighted in the left frame of Figure 5.26, in these conditions the two cascades exhibit comparable behaviors and no over-speed zones are observed.

At the highest part-load operation, reported in Figure 5.28, the flow over-expands in the diverging channel and suddenly adjusts to the outflow condition through a strong fishtail shock extending to the whole blade outlet section. Such shock impinges on the rear blade wall and propagates in the downstream region, producing a consistent worsening of the turbine performance for all the three cases, see Figure 5.25. The shock strength appears of greater intensity in the single-Point configuration due to the higher flow speed at outlet blade section. However, in the baseline assembly, the compression wave seems initially weaker compared to the one originating in the single-point optimization, but its intensity is rapidly strengthened in the free-stream region due to the interaction with the compression fan forming on the concave part of the suction side. The overlapping of these two physical phenomena produce a highly detrimental effect on flow uniformity and, hence, a relevant increase of the total pressure losses, as visible in Figure 5.25.

The Mach fields generated by the multi-point and single-point cascades display the same dynamics. In both cases the straight rear suction still prevents the appearance of compression fans, as shown in the right frame of Figure 5.26, while the shock wave pattern is similar. However, the former ge-

ometry takes advantage from the curvature of pressure side that significantly alters the ratio between the throat and the blade outlet section, reducing it by a factor of 10%. As a consequence the fluid flow exits from the turbine blade passage at a lower Mach number and, therefore, the shock intensity results considerably weakened compared to the one originating from the single-point profile. Notice anyway that in the multi-point case the reduction of the area ratio is essentially realized by enlarging the throat size, which induces an excess of mass flow rate of about 10% with respect to the other two configurations. Smaller deviations may be achieved by forcing different designs of the pressure side, even though lower fluid-dynamic performance are likely to be accepted. However the potential guaranteed by such a design procedure can be exploited if proper changes in the blade span are introduced.

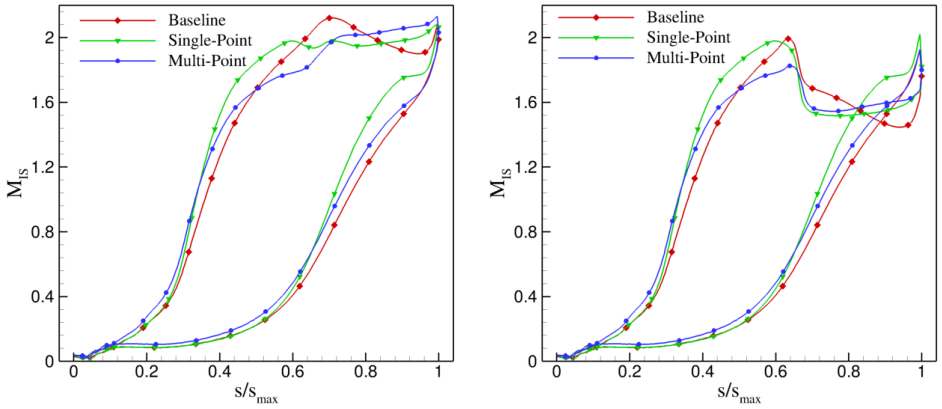


Figure 5.26. **Left:** *Predicted isentropic Mach number on the blade surface for the three geometries in reference conditions.* **Right:** *Predicted isentropic Mach number on the blade surface for the three geometries in worst operating scenario (static pressure equal to 2.0 bar).*

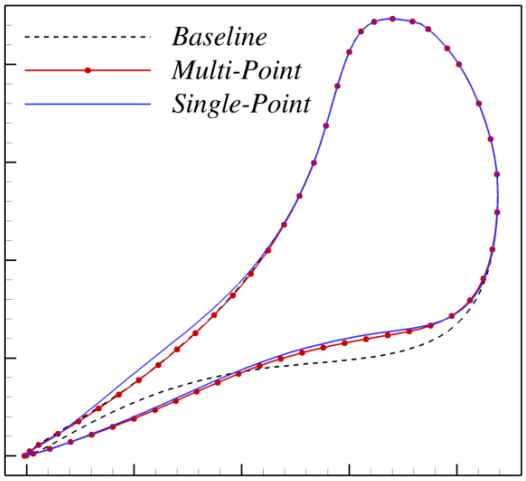


Figure 5.27. *Baseline, single-point and multi-point blade profiles.*

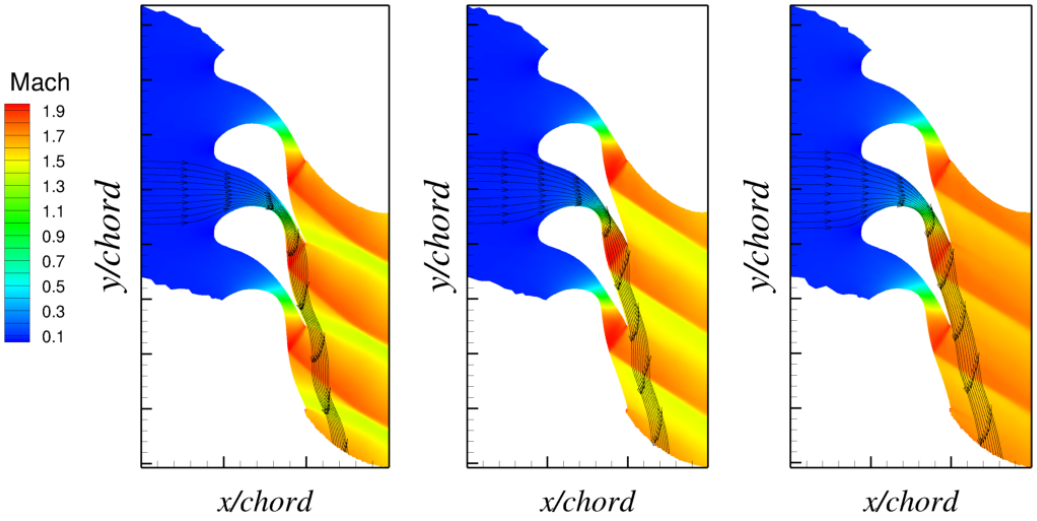


Figure 5.28. *Predicted Mach flowfields at worst operating scenario for the three geometries: baseline (left), single-point (center), and multi-point (right).*

5.5 Effect of Thermodynamic Model and Turbulence on Supersonic Blade Design

Most of the ORC turbines operate in thermodynamic regions characterized by strong real gas-effects. The importance of an accurate thermo-physical description of the fluid in ORC turbomachinery has been underlined in in a number of seminal works, see for instance [51], [98], and [82]. In these works has been outlined how the neglect of non ideal effects in the simulation process can lead to unsatisfactory predictions of the main flow features, such as Mach numbers and pressure coefficient distributions along the blade. Hence, automated design methods not considering real effects are expected to be not suited to obtain proper design in regions where strong non-ideal effects take place. Therefore, the present section investigates the potentialities offered by a fully real-gas adjoint-based design methodology compared to approaches based on simplified equations of state.

5.5.1 Influence of Thermodynamic Model

The main outcomes of shape optimization of a converging-diverging blade passage using the simplified polytropic polytropic Van der Waals have been extensively analyzed in the previous section. However, the nominal operating conditions of the supersonic turbine leads to an expansion process characterized, almost in the initial part, by relevant real-gas effects (inlet compressibility factor close to 0.7). As a result, the scant accuracy of the PVdW model in predicting the thermo-physical behavior of the fluid can have a significant impact on the resulting optimized blade shape. In fact the optimal configuration found by introducing the polytropic Van der Waals equation cannot longer represent the optimal point for the same design procedure employing a different equation of state. For turbines working close to the critical point the influence of this effect may be even amplified. In fact, the use of approximate thermodynamic relations in both flow and adjoint solver may lead to descent directions different from those observed using a more accurate EoS. As a consequence, in the worst scenario, the optimal assembly could even behaves worse than the initial baseline when its performances are verified by using more accurate EoS, although the optimization process has converged to a local minimum.

In order to point out the influence of the accuracy of the thermo-physical model of the fluid on the final blade shape, two further single-point design procedures are carried out, the former using the simplest polytropic ideal gas law and the latter adopting the more complex Span-Wagner equation of state. Span-Wagner (SW) EoS represents a *state-of-the-art* multi-parameter fundamental relation having a superior accuracy compared to that of cubic

equations of state, see [99] and [100]. In this study the SW EoS, hereinafter called SW-LUT, is available through the LUT algorithm presented in chapter 4, which is implemented in both flow and adjoint solver. Notice that the use of a LUT method in both solvers, which represents one of the major novelties of this work, fully decouples the issue of selection of the most suited thermodynamic model from the simulation processes. In fact, any type of equations of state is potentially usable, the choice being only dependent on the database employed to construct the look-up table.

The design processes use the same parameterization, computational grid, and optimization parameters of the previous test case. The LUT is composed by about 20000 elements and covers a thermodynamic area similar to that represented in the left frame of Figure 4.9. Similar convergence rates are achieved in both cases, as depicted in Figure 5.29. The total computational expense is about 30% higher compared to PVdW optimization.

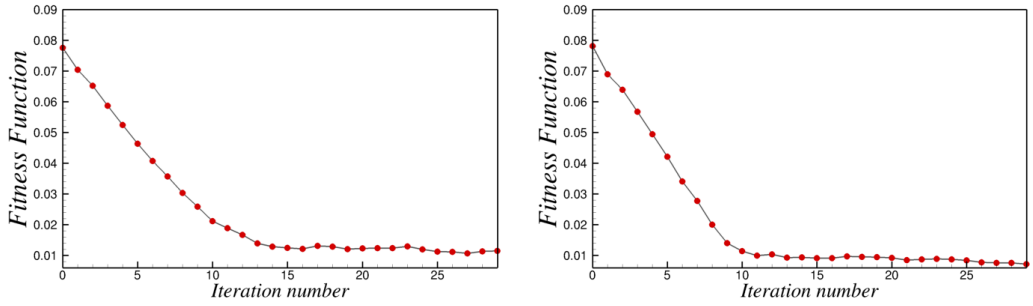


Figure 5.29. **Left:** *Convergence history of shape optimization using the SW-LUT model.* **Right:** *Convergence history of shape optimization using the PIG model.*

The profiles of the three geometries are reported in the left frame of Figure 5.30. The slight differences on both pressure and suction side affect the shape of the diverging part of the bladed channel. In fact the combination of a more straight pressure side and more curved suction profile in the bladed channel leads to an increase of the ratio between the throat area and the blade outlet section, as it is well depicted in the right frame of Figure 5.30.

The performances of the resulting optimized configurations are therefore investigated by computing the flow pattern using the accurate SW-LUT model. The resulting flow-fields in terms of Mach number are depicted in Figure 5.31.

The modification of the area ratio of the diverging channel induces a higher flow acceleration within the turbine blade passage, both on pressure

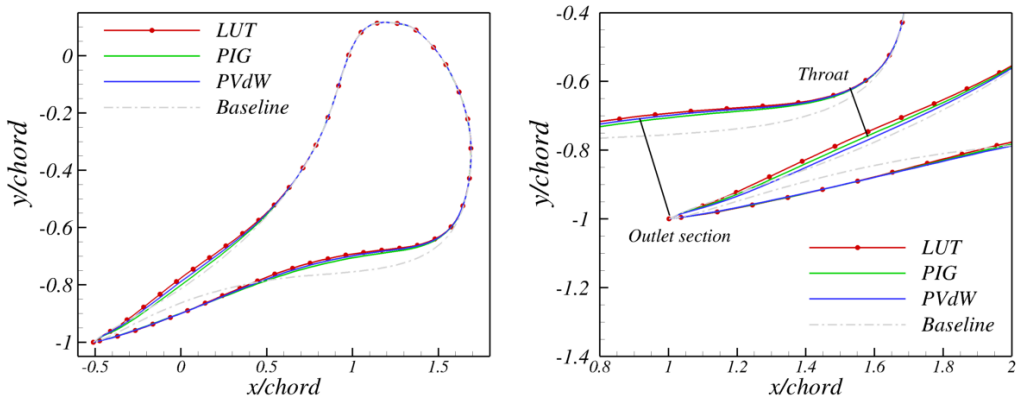


Figure 5.30. **Left:** *Layout of the optimized geometries obtained using PIG, PVdW and SW-LUT equations of state.* **Right:** *Area ratio of the baseline geometry compared to the geometries optimized by using PIG, PVdW and SW-LUT thermodynamic model.*

and suction side, as suggested by the isentropic Mach number distribution shown in Figure 5.32. Then, fluid velocity does not further increase on the rear straight suction side for the Span-Wagner case. In the other two configurations the effect of a greater acceleration on the rear suction profile lead to a higher Mach number just upstream of the trailing-edge. As a consequence, the intensity of the fishtail shock detaching from the trailing-edge is more stronger for the configurations optimized using the simplified equations of state. This in turn strongly affects the flow uniformity at the outlet domain, shown in Figure 5.33, and the resulting performances of the cascades, see Table 5.12. Summarizing, this study pointed out the intrinsic limitation of ideal gas-based adjoint methods for shape optimization in dense-gas flow regions and demonstrates the capability of the current real-gas adjoint method in re-designing existing ORC cascades with relatively low computational expense.

	Baseline	*Optimized (PIG)*	*Optimized (PVdW)*	*Optimized (SW)*
J	0.077	0.02	0.018	0.014
Y	11.78 %	4.8 %	4.5 %	3.6 %

Table 5.12. *Predicted fitness function J values and total pressure loss coefficient $Y = \frac{P_{T0}-P_{T1}}{P_{T0}-p_{s1}}$ for the baseline and optimized blades using PIG, PVdW, and SW-LUT Eos.*

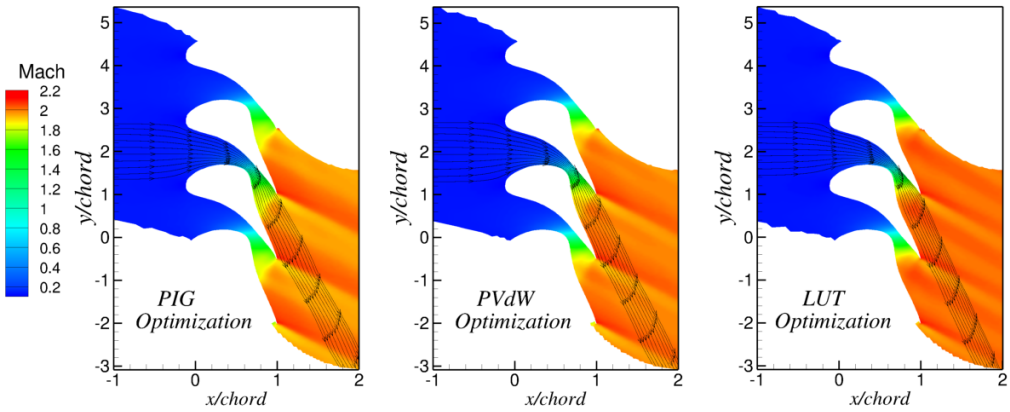


Figure 5.31. *Mach flow-field obtained with the three optimized blades (PIG, PVdW and SW-LUT).*

Notice that, as already mentioned, the final objective of these shape optimizations is the maximization of fluid-dynamic efficiency of 2D profiles. Therefore, in realistic 3D prismatic blades, different predicted values of mass flow rate in blade-to-blade plane can be easily managed by acting on spanwise direction, i.e. by either adjusting the blade height or the diameter of the cascade.

5.5.2 Design Validation for Turbulent Flows

The inviscid fluid model, in absence of significant boundary layer-shock wave interactions and flow separations, is often sufficient to predict the performances of supersonic stages with a reasonable accuracy. Therefore the optimized configuration resulting from an inviscid design process should better behave also in case of turbulent flows, if the most detrimental phenomena, i.e. the shock waves, are attenuated in the final cascade layout. However, a quantitative assessment of the performances of both baseline and optimized geometry using a turbulent model is mandatory to confirm the indications provided by the (inviscid) design methodology. Moreover, turbulent simulations give the opportunity of an indirect verification of the whole design procedure, especially if they are performed through highly validated CFD tools, widely recognized as a standard from industries and academic institutions. In the present study the ANSYS-CFX is utilized. The optimized geometry taken as reference for the calculations is the profile designed by means of the Span-Wagner EoS, as the thermodynamic model used in the design procedure is easily accessible via interpolation methods in a number of commercial tools.

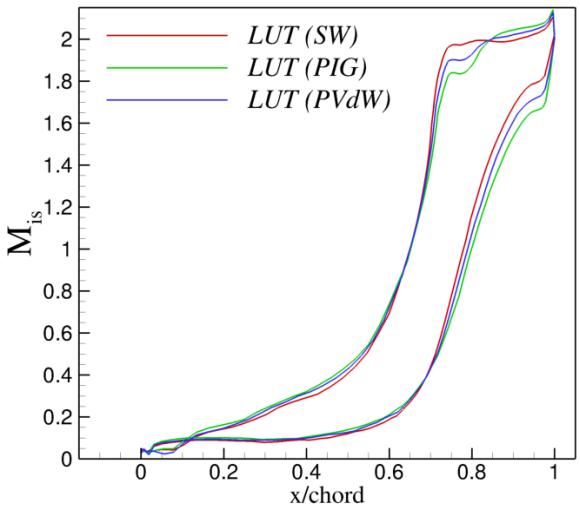


Figure 5.32. *Predicted isentropic Mach number distribution along the blade profile for the three optimized blades (PIG, PVdW and SW-LUT).*

Computational Model

The analysis of the cascade performances is carried out on the basis of quasi-3D CFD simulations. As only blade-to-blade effects are of interest in this study, a straight stream-tube around midspan is taken into account, hence the quasi-3D calculations were performed with constant thickness of the computed channel.

The calculations were performed using the ANSYS-CFX code, applied to structured grids composed by hexahedral elements, with 2 cells in spanwise direction. Turbulence effects are introduced using the $k-\omega$ SST model, ensuring wall y^+ well below the unity (maximum value about of 0.2) all along the blade; as discussed above, the MDM is treated as a real gas through a look-up table interpolation method suitably defined for the CFX solver. High resolution numerical schemes and implicit time integration are used. Total conditions, flow angles and turbulence quantities are assigned at the inlet, static pressure is given at the outlet. Non-reflecting conditions are indirectly obtained by placing the outflow domain at a distance of about three axial chords from the trailing-edge. Preliminary calculations were carried out to determine the proper position of the boundary section. The calculations were performed on a Linux cluster composed by 7 parallel two-processors.

A grid independence analysis was performed, by reducing grid spacing in the blade-to-blade plane. The outcome of this analysis indicates that a grid independent solution is achieved with at least 50 kcells in the blade-to-blade plane (resulting in 100000 hexahedral cells in the 3D domain). The relatively

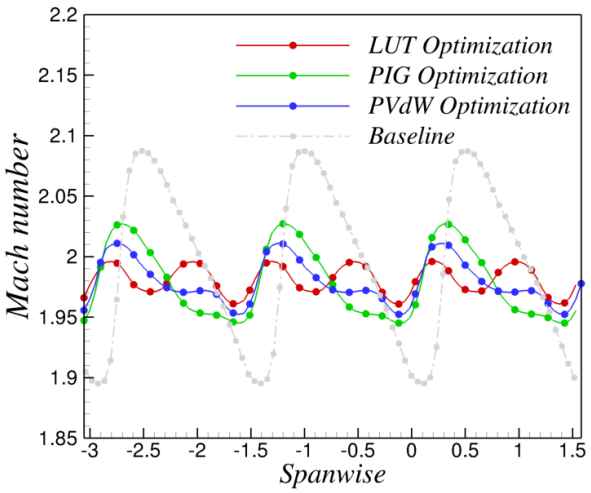


Figure 5.33. *Mach pitch-wise distribution obtained with the three optimized blades (PIG, PVdW and SW-LUT).*

high number of elements required is motivated by the presence of shocks, and by the significant grid clustering at the profile walls to guarantee the desired value of y^+. The resulting computational time for each simulation was about one hour, relatively high due to the large number of element and the 3D nature of the model, but still acceptable for the present investigation.

The numerical model used in this context was previously assessed against measurements performed on a research turbine stage installed at Politecnico di Milano [101]. In this paper, the CFD model was shown to provide estimates of stage efficiency with an error around 1%, while the difference between experimental and computed stator losses was found of the order of 0.2%. These results let the author consider the numerical model herein used as a high-fidelity analysis tool.

Results of the Simulations

The results of turbulent simulations are now deeply discussed. Most of the flow features observed in the inviscid simulations are conserved in turbulent calculations. As well visible in Figure 5.35 and Figure 5.36, the strong shock originating on the trailing-edge and impinging on the curved suction side in the baseline geometry is highly weakened in the optimized configuration. This is a direct consequence of the full re-design of the blade rear suction side, which affects the shape of both diverging channel and semi-bladed region. The diverging portion of the optimized configuration results characterized by higher passage area ratios. As in the case of inviscid flows, the fluid

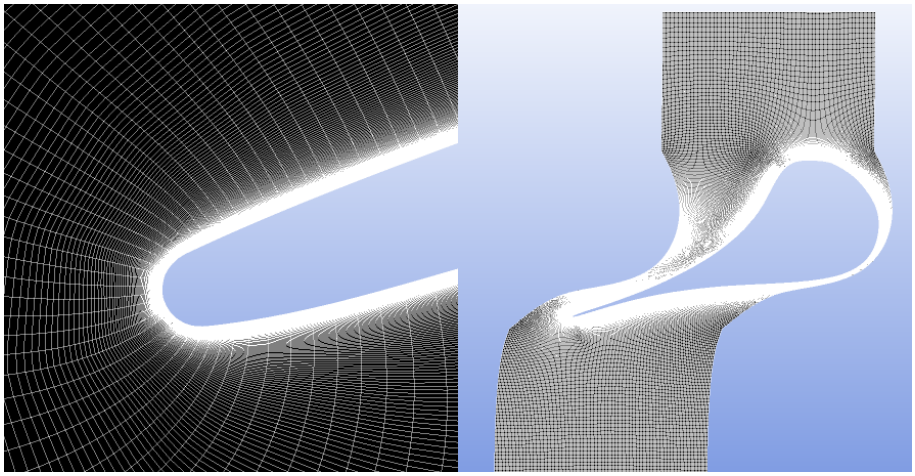


Figure 5.34. *Computational mesh of the turbulent simulations. A very high-quality mesh resolution is obtained close to the trailing-edge.*

stream is then subject to larger accelerations on both pressure and suction sides of the diverging channel, reaching higher outlet Mach numbers. The flow is not further accelerated at the exit of the bladed region due to the straight designed rear suction profile, see Figure 5.38. Mach number and flow angle spanwise distribution are therefore more uniform in the optimized configuration, see Figure 5.37, resulting in a considerable improvement of the performances.

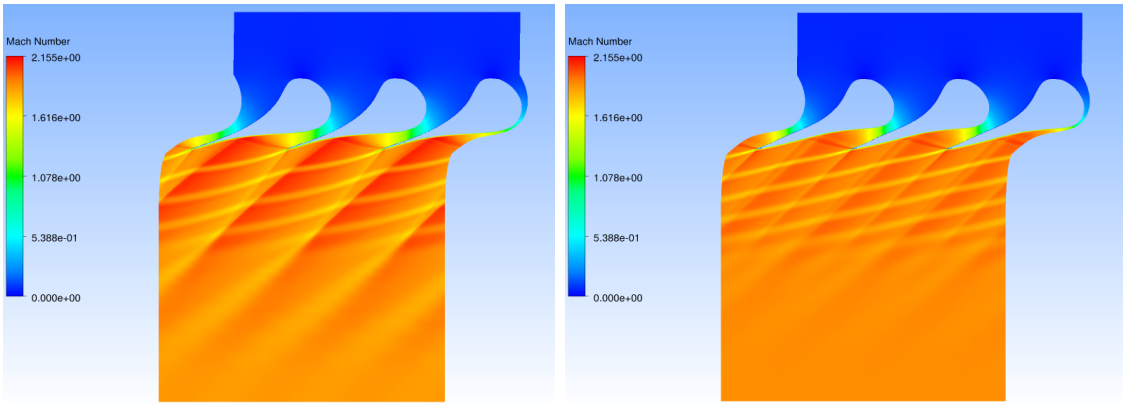


Figure 5.35. **Left**: *Predicted Mach number distribution for the baseline geometry.* **Right**: *Predicted Mach number distribution for the optimized geometry.*

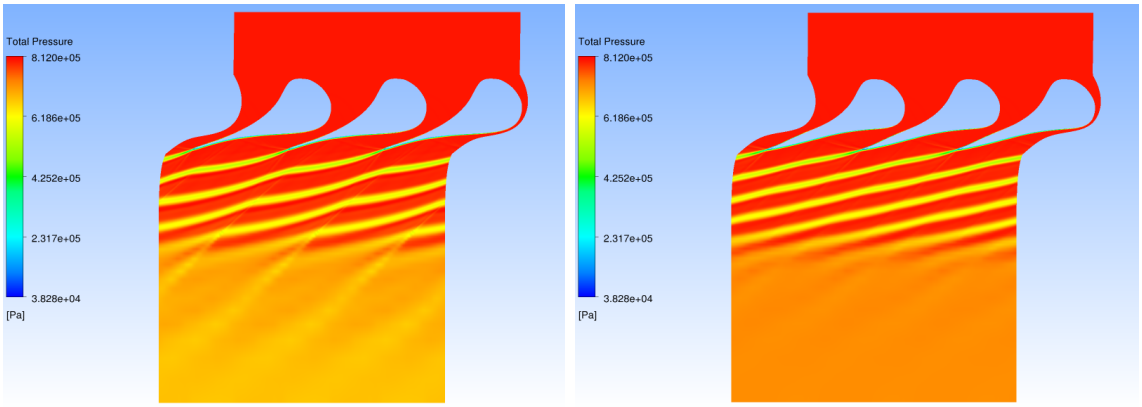


Figure 5.36. **Left**: *Predicted total pressure distribution for the baseline geometry.* **Right**: *Predicted total pressure distribution for the optimized geometry.*

The final results of the simulations are reported in Table 5.13. Notice that the reduction of the total pressure loss coefficient is approximately in accordance with the values provided by the inviscid simulations. This means that the improvement of the performances is almost fully related to the weakening of shock wave intensities. A similar loss contribution in both configurations (baseline and optimized) is expected due to shear stresses and viscous shear layer generated at the trailing-edge. Hence, the outcomes of the turbulent simulations demonstrate the validity of the design methodology developed within this research.

Blade	Y	m_{flow}	α_{flow}	$Mach_{mix}$
Baseline	20.32 %	1	76.53 deg	1.886
Single-Point	12.24 %	0.94	77.65 deg	1.911

Table 5.13. *Results of turbulent simulations for the baseline and optimized configurations. Mass flow rates are reported in non-dimensional form.*

As long as flow separations do not occur and shock wave-boundary layer interactions are of weak intensity, inviscid model has been deemed suitable for accurately predicting the behavior of transonic/supersonic flows at a low computational expense. Compared to inviscid case, the results of the turbulent calculations over-estimate the total pressure loss coefficient of about 8% for either baseline or optimized configurations. Most of this difference

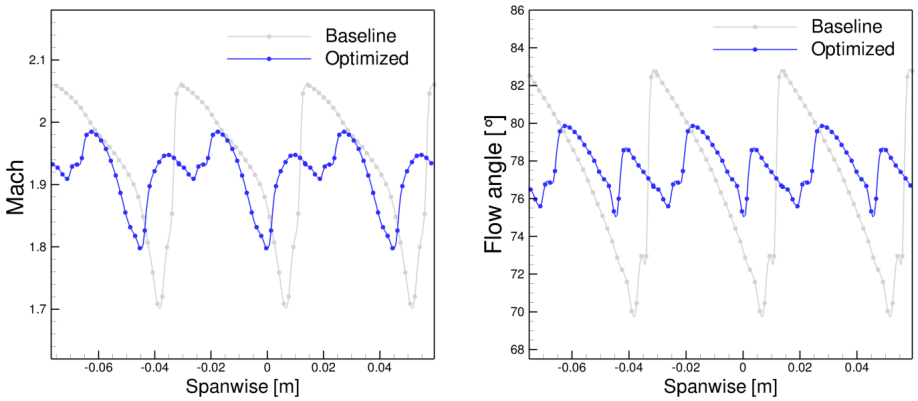


Figure 5.37. **Left**: *Predicted spanwise Mach number distribution at outflow boundary.* **Right**: *Predicted spanwise flown angle distribution at outflow boundary.*

originates from the mixing process downstream of the cascade, which now includes the effect of the shear layer (in addition to shock waves) on the mixed-out averaged quantities. A minor contribution may be associated to the dissipation induced by the different numerics of the two solvers. However, the use of very fine grids strongly reduces such an effect.

To reliably compare inviscid and turbulent simulations the attention has then to be placed on the flow pattern up to the trailing-edge, thus neglecting the mixing zone. The isentropic Mach number distribution along the blade surface, for instance, represents a good choice to understand whether significant differences between the two models (and solvers) happen. Alternatively, static pressure fields provide useful indications about the reliability of the inviscid solution. In this context, Mach number distribution along the blade surface are reported. Left and right frames of Figure 5.39 show the profiles on the baseline and optimized blade. Inviscid and turbulent Mach profiles greatly agree all along the pressure side and for the major part of the suction side. The largest discrepancies are located on the rear suction side, whereby the fishtail wave generating at the trailing-edge impinges against the blade wall, see Figures 5.40 and 5.41. The phenomenon becomes really dissipative only for turbulent flows, mostly due to viscous effects, whereas for inviscid flows such a wave is simply formed by a sequence of isentropic expansion fans. This finally reflects in an increase of total pressure losses which translates into a reduction of isentropic Mach number along the rear suction side.

Moreover, turbulent calculations pointed out negligible interaction effects between boundary layer and shock wave in both configurations.

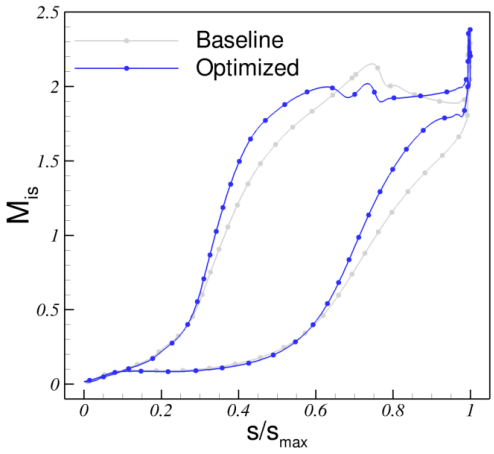


Figure 5.38. *Isentropic Mach number distribution on the blade surface.*

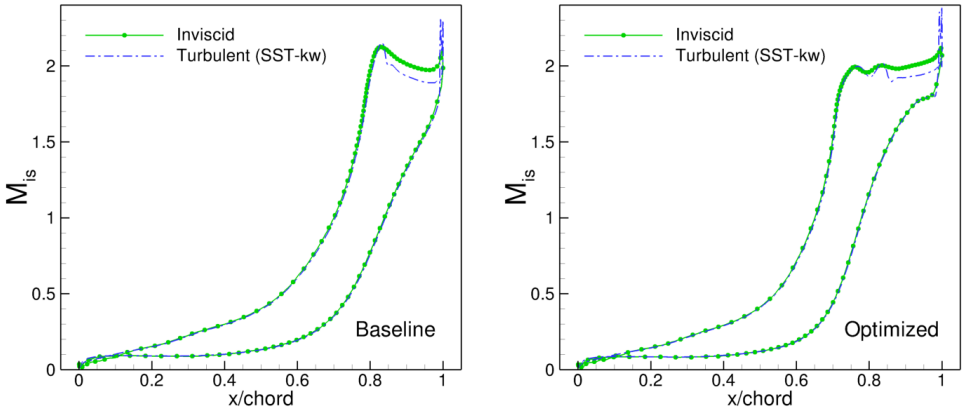


Figure 5.39. **Left:** *Predicted isentropic Mach number on the baseline blade surface for inviscid and turbulent flows.* **Right:** *Predicted isentropic Mach number on the optimized blade surface for inviscid and turbulent flows.*

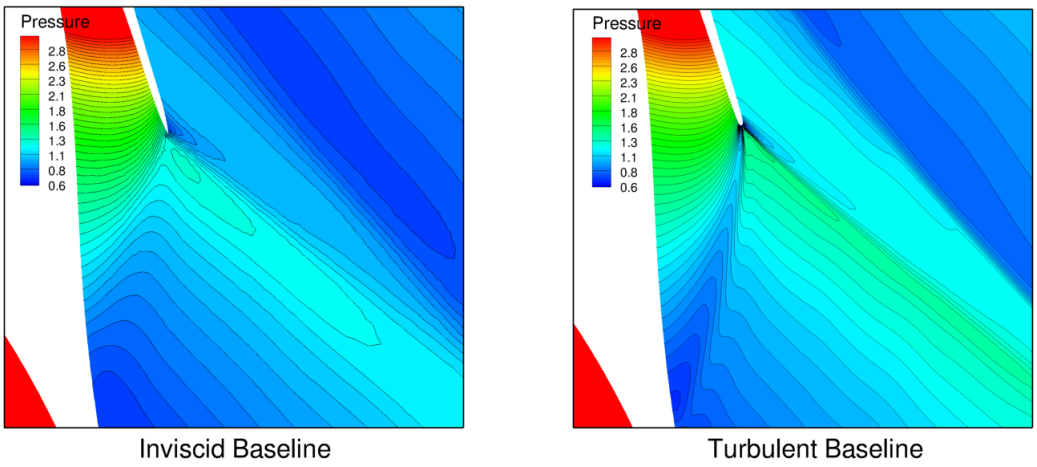


Figure 5.40. **Left:** *Predicted fishtail wave pattern for baseline configuration in case of inviscid flows.* **Right:** *Predicted fishtail shock pattern for baseline configuration in case of turbulent flows.*

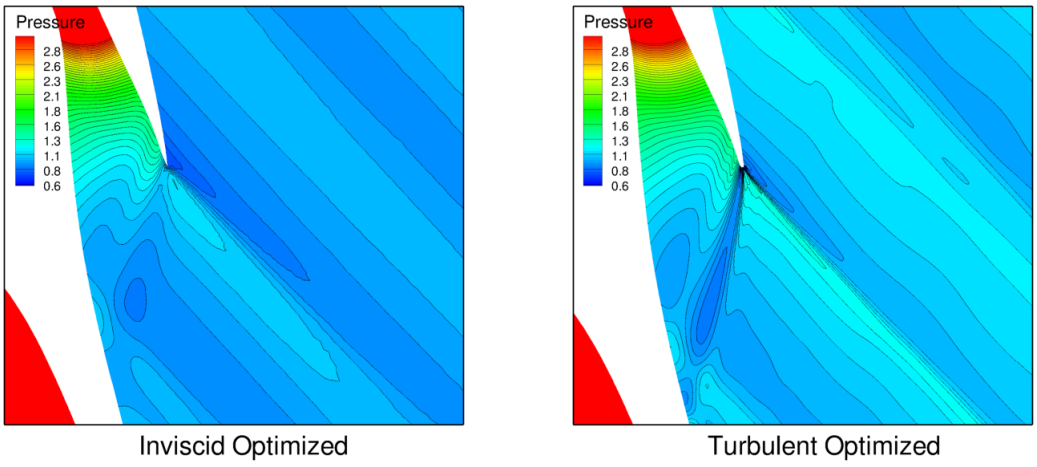


Figure 5.41. **Left:** *Predicted fishtail wave pattern for optimized configuration in case of inviscid flows.* **Right:** *Predicted fishtail shock pattern for optimized configuration in case of turbulent flows.*

Chapter 6

Adjoint-based Uncertainty Quantification

According to AIAA, the uncertainty if CFD is a potential deficiency in any phase or activity of the modeling process that is due to the lack of knowledge. On the contrary, error in CFD simulations is clearly deterministic and is defined as a recognizable deficiency in any phase or activity of the modeling process that is not due to the lack of knowledge. This chapter focuses on the uncertainty concept investigating the potential of the adjoint method for statistical predictions.

6.1 Introduction on Uncertainty Quantification in CFD

The use of automatic optimization algorithms in engineering is nowadays widely used in the design process. Thanks to the recent increase of computational capabilities, effective optimization calculations can be carried out in a time compatible with industrial needs. These are generally based on a deterministic approach. i.e. all involved quantities (operating conditions, models, geometrical data) are supposed to be perfectly known and to exactly match the real-world situation. In practice, an engineering component is affected by geometric tolerances; moreover, it may operate at off-design conditions for a large amount of its life; finally, the mathematical model itself, along with its associated closure parameters may also be imperfectly known. Given the number of uncertainties affecting engineering systems, deterministic optimization approaches may offer good performances at nominal conditions, but poor off-design behavior.

As a consequence, design strategies addressing the effect of uncertainties are particularly attractive for the optimization of realistic problems. The objective is to achieve designs that are less sensitive to input parameter variability. Such an approach is known as robust design. The objective of robust design methodologies may be of two kinds: to achieve a suitable trade-off between maximization of mean performances and their stability under system uncertainty or to minimize system failure probability under uncertainty (see, e.g., Ref. [102]). In this work we address the first kind of problems.

Our purpose is to develop an efficient robust design methodology to optimize the stochastic performances of a given functional (typically its mean and/or variance). Low-order statistical moments are evaluated by an uncertainty quantification (UQ) algorithm coupled to the optimization process. Nested approaches, given by an internal UQ method and an external optimizer [103] have a total computational cost that is roughly the sum of those of the two algorithms. When expensive fitness functions or multiple uncertain variables are taken into account, the global expense may indeed become prohibitive. This is often the case for Fluid Dynamics applications, where fitness function evaluations imply the solution of complex systems of non-linear equations (Euler or Navier-Stokes).

Considerable research efforts are hence currently underway in order to reduce the number of deterministic evaluations required for obtaining a robust design. For instance, Congedo et al. [104] integrate a Simplex Stochastic Collocation (SSC) method with a geometric simplex optimization algorithm. The SSC method combines effective random sampling in high dimensional

spaces (multiple uncertainties) and polynomial interpolation. Nevertheless, its application to complex configuration strongly relies on the possibility of massive parallelization [105]. Massive parallelization of one or both algorithms is also adopted in Refs [103, 106]. In Ref. [107] the original fitness function is replaced by less expensive models (meta-modeling). Also, two-step approaches, in which only the most influential uncertain parameters –selected through a preliminary step– are considered in the optimization process have been explored [108, 109]. In the last case, global efficiency of the algorithm depends on the ability of quickly and accurately sorting the uncertain parameters according to their impact on the optimization objectives.

The versatility and efficiency of robust optimization is tightly related to the performance of the underlying uncertainty quantification methods. Intrusive approaches, like Polynomial Chaos (PC) [110], guarantee faster convergence for problems with multiple uncertainties, but their implementation needs a substantial modification of existing deterministic codes [111]. On the other hand, non intrusive methods, such as Monte Carlo (MC) methods, require a number of function evaluations that is often too large to be affordable. Non-Intrusive Polynomial Chaos (NIPC) [112] and Probabilistic Collocation Methods (PCM) [113] considerably reduce computational cost with respect to MC methods, but their cost increases exponentially with the number of uncertain parameters, even if it can be somewhat alleviated by means of sparse-grid sampling [114] and ANOVA reduction [115] techniques.

Among the UQ methodologies, the Method of Moments (MoM), which approximates statistical moments of the fitness function by Taylor series expansions, has proven to be remarkably fast, as long as the fitness sensitivity derivatives with respect to uncertain parameters are provided by an efficient method and complete output statistics are not required. In the context of fluid-dynamics, the adjoint formulation has been conceived for this aim and successfully applied to several problems [24]. Since the cost of adjoint is nearly independent from the number of uncertain variables, it allows saving a relevant amount of time with respect to finite difference techniques. Thanks to these features the Adjoint Method of Moments (AMoM) is a good candidate for coupling with optimization algorithms in robust design strategies, because of its ability of providing low-order moments of the cost functionals, once the sensitivities are obtained by solving the adjoint problem. This methodology may be particularly attractive for multi-objective optimization, as commonly required for fluid-dynamics applications: for this kind of optimization problems, genetic algorithms are often adopted due to their flexibility and their capability to reach global optima. However, they require multiple function evaluations at each iteration, with an associated

high computational expense, especially if such function evaluation requires in turn multiple deterministic runs to estimate statistical moments of the objectives [109].

In this chapter a novel hybrid robust optimization approach is proposed. It combines the efficiency of AMoM with the capability of a multi-objective genetic algorithm (MOGA) to search for the best design over the whole space of solutions. According to the order of approximation used for statistical moments (first or second), only two or three deterministic calculations (non-linear, adjoint and eventually linear) have to be run for each individual generated by MOGA.

To demonstrate the potential of this approach a package of quasi 1-D codes, including a non-linear, a discrete adjoint, and a linear solver is developed. The numerical codes are integrated in an automatic procedure that allows to compute efficiently the sensitivity derivatives of the fitness function(s) based on the assigned input distributions of uncertain variables. The derivatives are subsequently used to estimate the expectancy and variance of the quantities of interest through a generalized Adjoint Method of Moments [116]. The performances of the AMoM, in terms of output statistics accuracy and cost, are initially assessed against other well-established UQ techniques. Then, the method of moments is coupled to an optimization algorithm and the proposed robust optimization method is applied to an inverse design exercise for quasi 1-D flows through a diverging nozzle.

The chapter is organized as follows: Section II provides an overview of some popular UQ techniques; the adjoint theory is also briefly recalled. Section III synthetically illustrates the optimization algorithms considered in this work. Section IV describes the numerical solvers and the fully automated procedure developed to compute the low-order statistical moments by AMoM. Section V assesses the method of moments against several UQ methods for a simple quasi 1-D problem with multiple uncertainties. Finally in Section VI the proposed robust optimization strategy is described and applied to an inverse design exercise for quasi 1-D flows with an embedded shock. A comparison with other possible robust design methodologies is also provided in terms of solution accuracy and computational cost.

6.2 Uncertainty Quantification Algorithms

This section describes the UQ algorithms used in this study. The Adjoint Method of Moments developed in this study is presented first. Then, we briefly recall two alternative methods considered here for comparison purposes and available in the Dakota open source software [91] (provided by Sandia Lab), which is extensively employed in this work.

6.2.1 The Method of Moments

The idea behind the Method of Moments (MoM) [107] origins from the application of the expectation operator to the local approximation (Taylor series) of a functional J around the mean value of the stochastic input variables. Here we consider a generalized formulation for the case of multiple uncertain variables with statistical distributions *a-priori* are known.

The second-order Taylor expansion of a functional $J(\alpha)$, with $\boldsymbol{\alpha}$ a random vector, reads:

$$J(\overline{\boldsymbol{\alpha}} + \delta\boldsymbol{\alpha}) = J(\overline{\boldsymbol{\alpha}}) + \sum_{i=1}^{N} \frac{\partial J}{\partial \alpha_i}\bigg|_{\overline{\boldsymbol{\alpha}}} \delta\alpha_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial^2 J}{\partial \alpha_i \partial \alpha_j}\bigg|_{\overline{\boldsymbol{\alpha}}} \delta\alpha_i \delta\alpha_j + o(|\delta\alpha_i|^2) \tag{6.1}$$

where $\overline{\boldsymbol{\alpha}} = E[\boldsymbol{\alpha}]$ is the expectancy of $\boldsymbol{\alpha}$ and $\delta\boldsymbol{\alpha} = \boldsymbol{\alpha} - \overline{\boldsymbol{\alpha}}$ is by construction a randomly distributed vector with zero mean. By applying the statistical definition of mean and variance to either the linear or quadratic Taylor approximation, a first- or, respectively, second-order approximation of the first two moments (mean and variance) is obtained. For instance the first order expectancy and variance for generally distributed and correlated uncertainties may be expressed as:

$$E[J]_1 = J(\overline{\boldsymbol{\alpha}}) \tag{6.2}$$

$$Var[J]_1 = \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial J}{\partial \alpha_i}\bigg|_{\overline{\boldsymbol{\alpha}}} \frac{\partial J}{\partial \alpha_j}\bigg|_{\overline{\boldsymbol{\alpha}}} C_{i,j} \tag{6.3}$$

where $C_{i,j} = E[\delta\alpha_i \delta\alpha_j]$ the covariance matrix. If the components of $\delta\alpha$ are not correlated the covariance matrix reduces to a diagonal one and equation 6.3 takes the simplified form:

$$Var[J]_1 = \sum_{i=1}^{N} \left(\frac{\partial J}{\partial \alpha_i}\right)^2\bigg|_{\overline{\boldsymbol{\alpha}}} \sigma_i^2 \tag{6.4}$$

with $\sigma_i^2 = Var[\delta\alpha_i]$. In this work, we mostly use second-order formulas (see references [117] and [116] for details), given by:

$$E[J]_2 = J(\overline{\boldsymbol{\alpha}}) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial^2 J}{\partial \alpha_i \partial \alpha_j}\bigg|_{\overline{\boldsymbol{\alpha}}} C_{i,j} \tag{6.5}$$

$$Var[J]_2 = \sum_{i=1}^{N}\left(\frac{\partial J}{\partial \alpha_i}\right)^2\Big|_{\overline{\boldsymbol{\alpha}}}\sigma_i^2 + \sum_{i=1}^{N}\frac{\partial^2 J}{\partial \alpha_i^2}\frac{\partial J}{\partial \alpha_i}\Big|_{\overline{\boldsymbol{\alpha}}} S_i\sigma_i^3 + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\left(\frac{\partial^2 J}{\partial \alpha_i \partial \alpha_j}\right)^2\Big|_{\overline{\boldsymbol{\alpha}}}\sigma_i^2\sigma_j^2 + \frac{1}{4}\sum_{i=1}^{N}\left(\frac{\partial J}{\partial \alpha_i}\right)^2\Big|_{\overline{\boldsymbol{\alpha}}}(K_i-1)\sigma_i^4 \tag{6.6}$$

where S_i and K_i are the skewness and the kurtosis of the input distributions. Note that eqs. (6.3), (6.5), (6.6) contain the first and second order sensitivities of J. As already seen, the most direct way to compute them is through perturbation techniques, such as finite differences. Nevertheless the accuracy of finite difference estimates is subject to the step-size definition and truncation errors. Furthermore, for expensive calculations involving a large number of uncertainties, their computational cost in terms of function evaluations is so high that they are simply not applicable. As in the case of shape optimization, a more powerful and effective alternative is represented by adjoint method [118].

Hereafter, the mathematical fundamentals of the approach is briefly recalled on the basis of what already discussed in chapter 2. A simplified theory, that does not account for mesh deformation, is presented. However, the given formulation is suitable for treating quasi 1-D problems, in which the mesh size is not perturbed by a geometry modification (if the overall domain length remains fixed). Let be J a scalar function of a vector of some control/design variables $\boldsymbol{\alpha}_{geo}, \boldsymbol{\alpha}_{phy}$ and state variables $\boldsymbol{u}$. For the sake of brevity the set of control/design variables is simply indicated by $\boldsymbol{\alpha}$ in the following. The state variables u are in turn constrained to satisfy the non-linear (Euler or Navier-Stokes) governing equations $\boldsymbol{R}$. The resulting relations can be written as

$$J = J[\boldsymbol{\alpha}, \boldsymbol{u}(\boldsymbol{\alpha})] \qquad s.t. \quad \boldsymbol{R}[\boldsymbol{\alpha}, \boldsymbol{u}(\boldsymbol{\alpha})] = 0. \tag{6.7}$$

Differentiating the functional J in equation (6.7) with respect to $\boldsymbol{\alpha}$ it allows to express its gradient as a function of the first-order sensitivity derivatives $\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{\alpha}}$ of state variables with respect to control parameters:

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \frac{\partial J}{\partial \boldsymbol{u}}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{\alpha}} \tag{6.8}$$

This dependency can be eliminated by deriving the governing equations as follows:

$$\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{\alpha}} = 0 \tag{6.9}$$

which gives

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{\alpha}} = -\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right)^{-1}\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}\right) \tag{6.10}$$

By plugging the last expression $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\alpha}}$ into eq. (6.8) and by setting:

$$\boldsymbol{v} = -\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{u}}\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right)^{-1},$$

one finally obtains the following linear system:

$$\left(\frac{\partial J}{\partial \boldsymbol{u}}\right)^{T} + \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right)^{T}\boldsymbol{v} = 0 \tag{6.11}$$

where v is the so-called adjoint variable vector. Eq. (6.11) represents the discrete adjoint formulation considered in this context. The $i-th$ component of the fitness gradient is finally evaluated by:

$$\frac{dJ}{d\alpha_i} = \left(\frac{\partial J}{\partial \alpha_i}\right)^{T} + \left(\frac{\partial \boldsymbol{R}}{\partial \alpha_i}\right)^{T}\boldsymbol{v} \tag{6.12}$$

By further differentiating the fitness function J we are able to determine each component of the Hessian matrix, which involves the evaluation of the second-order state sensitivity derivatives $\frac{\partial^2 \boldsymbol{u}}{\partial \alpha_i \partial \alpha_j}$. In order to avoid their calculation, a mathematical procedure similar to the previous one is applied, i.e. the constraint $\boldsymbol{R} = 0$ is double differentiated and the resulting expression for $\frac{\partial^2 \boldsymbol{u}}{\partial \alpha_i \partial \alpha_j}$ is then plugged in the second-order differentiated formulation of J. As a final result, the ij_{th} component of the Hessian matrix reads:

$$\frac{d^2 J}{d\alpha_i d\alpha_j} = D^2_{i,j}J + \boldsymbol{v}^T D^2_{i,j}\boldsymbol{R} \tag{6.13}$$

where $D^2_{i,j}J$ is given by:

$$D^2_{i,j}J = \frac{\partial^2 J}{\partial \alpha_i \partial \alpha_j} + \frac{\partial^2 J}{\partial \alpha_i \partial \boldsymbol{u}}\left(\frac{\partial \boldsymbol{u}}{\partial \alpha_j}\right) + \frac{\partial^2 J}{\partial \alpha_j \partial \boldsymbol{u}}\left(\frac{\partial \boldsymbol{u}}{\partial \alpha_i}\right) + \frac{\partial^2 J}{\partial^2 \boldsymbol{u}}\left(\frac{\partial \boldsymbol{u}}{\partial \alpha_i}\right)\left(\frac{\partial u}{\partial \alpha_j}\right) \tag{6.14}$$

and $D^2_{i,j}\boldsymbol{R}$ is defined in a similar way. Note that eq. (6.14) involves anyhow the calculation of state sensitivity derivatives $\frac{\partial \boldsymbol{u}}{\partial \alpha_{i,j}}$. These are in turn obtained by solving the linearized version, eq. (6.10), of the non-linear partial differential equations. The overall mathematical treatment and some implementation details can be found in [119]. Once the first and second-order derivatives are available, along with the variances of the control parameters

$\boldsymbol{\alpha}$, the Method of Moments allows to obtain the low-order moments (mean and variance) in a straightforward way, simply by inserting the derivatives of J in eqs. (6.2) and ((6.3). or in eqs. (6.5) and (6.6).

The development of a discrete adjoint and a linear code may be eased by means of Algorithmic Differentiation (AD) software. We refer the interested reader to [41] to understand how AD works. Roughly speaking, the automatic differentiation acts directly on a subroutine generating a linear or adjoint differentiated version of the provided numerical procedure, thereby relieving the computational burden associated to by-hand code differentiation. Within this study the AD tool Tapenade, developed at INRIA [42], is applied to the discretized primal equations. The adjoint and linear code are created by means of the reverse (adjoint) and forward (linear) mode of differentiation, respectively. A tangent-on-tangent (double linear differentiation) approach is instead applied to obtain the Hessian matrix [119].

6.2.2 Sampling Methods

Monte-Carlo methods produce the output statistics from a set of samples of the fitness function spread over the whole multidimensional space formed by the uncertain variables. Its basic, or crude, version consists in the following few steps:

1. Sample input random variable(s) from their known, or assumed, probability density function.
2. Calculate deterministic output(s) for each sampled input value(s).
3. Compute an hystogram by binning the results.
4. Determine statistics of the ouput distribution, such as *mean, variance, PDF.*

Despite the method is proven to be easy to implement and robust, the application of a brute-force MC algorithm is rather prohibitive for fluid-dynamics problems, since MC displays a low convergence rate (of the order of $1/\sqrt{(N)}$) and, as a result, hundreds or thousands of deterministic simulations are indeed compulsory to obtain an affordable output statistics. Anyway, basic MC represents a benchmark technique against which other methods are compared for simple flow problems. A random-based Monte-Carlo uncertainty analysis is therefore chosen as reference case for the application investigated in this study.

6.2.3 Stochastic Expansion Methods

Stochastic expansion methods refer to a class of uncertainty quantification algorithms that try to approximate the functional relationship between a set

of output response metrics and a set of input random variables. Hereafter, we present two non-intrusive stochastic expansion methods, namely Non Intrusive Polynomial Chaos (NIPC) and Probabilistic Collocation (PCM), used for comparison with the AMoM in Section V.

Non Intrusive Polynomial Chaos (NIPC)
Polynomial Chaos methods construct the approximation of a functional by means of a truncated series of multivariate orthogonal basis polynomials, chosen according to the Askey scheme [120], so to best fit the statistical distribution of uncertain parameters $\boldsymbol{\alpha}$. The Chaos expansion of a QoI $J(\boldsymbol{\alpha})$, function of a n-dimensional random vector $\boldsymbol{\alpha}$ takes the form:

$$J(\boldsymbol{\alpha}) = \sum_{i=0}^{P} \eta_j \psi_j(\boldsymbol{\alpha}) \tag{6.15}$$

where η_j and $\psi_j(\boldsymbol{\alpha})$ are, respectively, the weights and the multivariate orthogonal basis of the expansion, related to the probability density of input random variables. In Dakota [91], the weights η_j can be estimated through a non intrusive procedure using either spectral projection or linear regression. The basic idea of the first approach is to project the response against each basis function using inner product and enforcing orthogonality. The resulting integrals [121] are evaluated through sampling techniques, quadrature formulas, or sparse grids integration based on Smolyak's construction. The second alternative uses instead a single linear least squares solution of the form $\psi\eta = J$ and is associated with a so-called total-order expansion comprising a total number of terms N_t in the series given by the formula:

$$N_t = P + 1 = r\frac{(n+p)!}{n!p!} \tag{6.16}$$

where p is the order of the polynomial chaos expansion and r is an oversampling factor (typically $1 \leq r \leq 2$).

In both cases the final objective is the calculation of a stochastic response surface that allows to accurately predict the statistical moments of the cost function by a reconstructed Monte-Carlo method applied to the inexpensive model (6.15). The stochastic problem is reduced to a set of deterministic simulations at each collocation point.

In the present work a linear regression method (with $r = 1$) on both full and sparse grids is considered to extract the coefficients ξ_j of the chaos expansion. Full grids are built by both structured (tensor grids) and unstructured approaches (by a latin-hypercube method). Spectral projection based on Smolyak's sparse grid technique is also investigated for reducing the number of collocation points.

The fully tensored grid (for quadrature order equal to 3) and Smolyak's sparse grids corresponding to Gauss-Patterson rules for two uniformly-distributed non-dimensional uncertainties a and b is depicted in Figure 6.1.

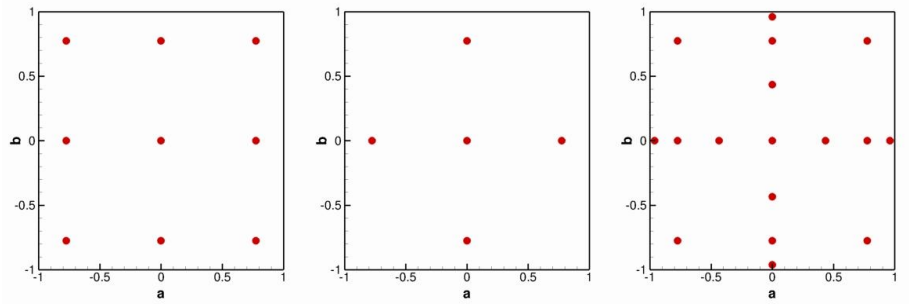


Figure 6.1. *Examples of stochastic grids. Full tensor grid (left), Gauss-Patterson sparse grid low resolution (middle), and Gauss-Patterson sparse grid high resolution (on the right).*

Probabilistic Collocation Method (PCM)

Probabilistic Collocation Method (PCM) is closely related to the polynomial chaos method. The key difference is that whereas NIPC employs bases of multivariate orthogonal polynomials, PCM uses Lagrange interpolation polynomials. In a single dimension (case of a single uncertain parameter $\boldsymbol{\alpha}$) Lagrange polynomials interpolate a set of collocation points α_j using the functional form:

$$J(\boldsymbol{\alpha}) = \sum_{j=1}^{N_p} J_j(\alpha_j) \prod_{k=1,k\neq j}^{m} \frac{\boldsymbol{\alpha} - \alpha_k}{\alpha_j - \alpha_k} \tag{6.17}$$

Equation (6.17) is used to build an approximation of the functional in the case of a higher dimensional uncertain vector by constructing a tensor product of 1-D polynomials of order $p-1$, being p the order of the quadrature polynomials, selected according the Askey scheme. The roots of the quadrature polynomials, permuted by a full factorial design, represent the knots of the global interpolating function. The result is a series of $N_p = p^n$ terms, corresponding to a multivariate interpolation function:

$$J(\boldsymbol{\alpha}) = \sum_{j=1}^{N_p} J_j(\alpha_j) \prod_{k=1}^{n} \left[\prod_{k=1,k\neq j}^{m} \frac{\boldsymbol{\alpha} - \alpha_k}{\alpha_j - \alpha_k} \right] \tag{6.18}$$

where the coefficient $J_j(\alpha_j)$ are the deterministic solutions (i.e. fitness function values) at collocation point α_j and n is the number of uncertain parameters. An total amount of p^n decoupled simulations is carried out for computing the weights of eq. (6.18). The output statistics are finally computed by a reconstructed Monte-Carlo method (RMC) applied to the interpolating function (6.18). Despite PCM is a simple and affordable UQ method, it suffers from severe limitations due to the *curse of dimensionality* (the computational cost grows exponentially with the number of uncertainties) and oscillatory behavior in presence of discontinuities (Runge's phenomenon). Various extensions to the tensor formulas have been proposed to overcome these difficulties. They include sparse grid techniques, simplex stochastic collocation [122] and the construction of the interpolating function by means of non-polynomial basis. In this work we employ PCM with a simple full factorial tensor formulation.

6.3 Numerical solvers

To explore the interest of AMoM for robust design optimization, a full package of quasi-1D codes for the Euler equations has been developed. It comprises an isentropic flow solver, here described, its adjoint, a linear code and numerical procedures for gradient and hessian computation. In the following, we provide the main features of these solvers.

6.3.1 Non-linear flow solver

The physical model used in the non-linear flow solver is represented by the well-known Euler equations for quasi-1D flows:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial f(\boldsymbol{u})}{\partial x} + B(\boldsymbol{u}) = 0 \qquad B(\boldsymbol{u}) = \begin{bmatrix} \rho v \\ \rho v h_t \\ \rho v^2 \end{bmatrix} \frac{1}{S}\frac{dS}{dx} \tag{6.19}$$

where $u_i = [\rho, \rho e_t, \rho v]$ and $f(u_i) = [\rho v, \rho v h_t, \rho v^2 + p]$ are, respectively, the conservative variable and the physical flux vectors, and $S = S(x)$ is the nozzle area at abscissa x. The system of equations is completed by a thermal and a caloric equation of state. For sake of simplicity in this work we use the perfect gas model with a constant specific heat ratio of 1.4 (air).

The governing equations are discretized by a cell-centered finite volume formulation. Three different numerical schemes (Rusanov's first-order scheme [123], Roe's first order scheme [123], and an high resolution extension of the Roe's scheme based on the van Albada-Van Leer's limiter [123]) are implemented within the code. A four-step explicit Runge-Kutta method is

adopted for time integration [124]. The code is constructed in a highly modular manner to facilitate its automatic derivation via TAPENADE [42, 39].

6.3.2 Adjoint and linear flow solvers

As pointed out in Section II, the adjoint and linear flow solvers are linearized versions of the baseline non-linear solver. They have been developed by applying the backward (adjoint) and forward (linear) mode of automatic differentiation to explicit parts of the native software. To construct the adjoint code, routines of the native nonlinear code calculating the fitness function J and the global residual R are inversely differentiated with respect to u in order to compute the vectors $(\frac{\partial J}{\partial u})^T$ and $(\frac{\partial R}{\partial u})^T v$, as used in equation (6.11). The gradient calculation is carried out by inversely differentiating the same routines with respect to the α_i to obtain the terms $(\frac{\partial J}{\partial \alpha_i})^T$ and $(\frac{\partial R}{\partial \alpha_i})^T v$ of equation (6.12).

The modular structure of the native solver is of paramount importance to ensure an efficient automatic differentiation of the non-linear routines. Furthermore it allows to easily change the objective function without significant modification of the overall numerical procedure.

The solution of the adjoint solver is converged by means of an explicit time-marching method (the same used for the non-linear solver). For the present problem, the algorithm proved to robustly converge the solution to a steady-state. Nevertheless, for more complex problems, convergence of adjoint solvers may require faster solvers, like GMRES [54]. These techniques will be considered in the future.

On the other hand, the linear solver computes the solution by constructing the matrix $\dfrac{\partial R}{\partial u}$ and the vector $\dfrac{\partial R}{\partial \alpha_i}$ through the direct differentiation of the numerical residual routine and by applying the LU factorization method (available in the LAPACK library) to solve equation (6.9). The resulting methodology is computationally efficient for a relevant number of sensitivity parameters. According to the definition of the Hessian components given in equation (6.13), the terms $D^2_{i,j}R$ and $D^2_{i,j}J$ are finally obtained directly by double differentiating the procedures that give J and R and providing the solution of the linear solver $\dfrac{\partial u}{\partial \alpha_i}$ as an input for the generated routines.

6.3.3 AMoM-based Uncertainty Quantification loop

In the UQ framework, all or part of the input variables α_i are considered as random variables with a known probability distribution function. In this case, information about the first and second sensitivity derivatives provided by the non-linear, adjoint, and linear solvers, are used to estimate how the randomness propagates through the system of governing equations and af-

fects the objective function J in terms of mean value and variance. For this purpose, we couple the numerical solvers described in the preceding subsections with the generalized Method of Moments, as sketched Fig. 6.2, specifically oriented to the evaluation of the low-order statistical moments.

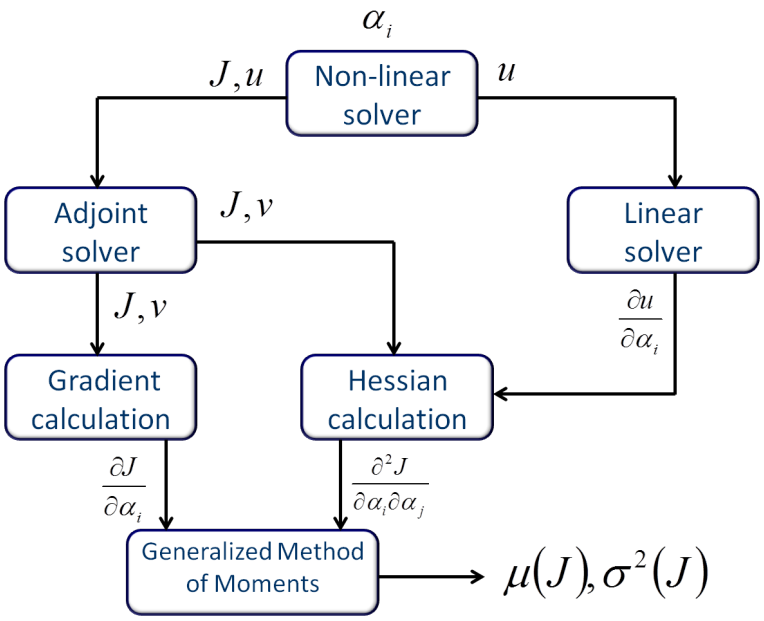


Figure 6.2. Automated coupling procedure for UQ for the quantity of interest (objective function).

Given an initial guess of the uncertain parameters α_i, the non-linear solver is run first, to evaluate the fitness function. Moreover, the computed values of the state variables u are passed-in to the adjoint and linear codes which return, respectively, the adjoint variable vector v and flow sensitivities with respect to the uncertain parameters, $\dfrac{\partial u}{\partial \alpha_i}$. This information is post-processed to obtain the gradient and the Hessian of the fitness function with respect to the stochastic input variables. Once the first and second-order derivatives are available, mean $\mu(J)$ and variance $\sigma^2(J)$ of the fitness function can be computed by the generalized method of moments. The low-order moments of the stochastic fitness function are finally made available as objectives for the robust optimization algorithms, presented hereafter.

6.4 Optimization algorithms

Robust optimization strategies investigated in this work are based on two underlying deterministic optimization algorithms: the first one is a Pareto-Based multi-objective Genetic Algorithm (MOGA), while the second one is a gradient-based BFGS (Broyden-Fletcher-Goldfarb-Shanno) method [125]. The MOGA is employed in a bi-objective search in which low-order moments, provided by an UQ method, are taken as separate fitness functions, in accordance with the formulation of the optimization problem. The gradient-based method is then used to refine the best-so-far solution of the MOGA[126], chosen on the Pareto front according to the following criterion: first the individual with minimum expectancy is chosen, then the objective is constructed by a linear combination of mean and standard deviation, each multiplied by a weighted factor of 0.5. In this way the gradient-based algorithm will tend to simultaneously optimize statistical moments, thus improving the best individual provided by MOGA with a bi-objective optimization.

In this work, we make use of the implementations of these algorithms available in the open source software Dakota. For the whole calculations carried out, the genetic algorithm starts with an initial population comprising 50 individuals which evolves for 30 generations to ensure a fully converged Pareto front.

6.5 Numerical Applications

This section presents the numerical applications of AMoM in UQ and robust optimization context extensively debating the results against other strategies. The section is divided into three parts: the first defines the test case while the second and the third one report the UQ analysis and the robust design optimization results for the stated problem.

6.5.1 Problem statement

In the following we consider transonic flow through a quasi-1D diverging nozzle. The flow equations are discretized by using a computational grid of 300 uniformly spaced cells, and Rusanov's spatial discretization scheme. The nozzle area distribution is parametrized by the following function:

$$S(x) = a + b \cdot tanh(cx - d) \tag{6.20}$$

where a, b, c, d are free coefficients that define the geometry.

The moderate computational expense of the flow solver, of the order of six CPU seconds on a standard PC, enables to efficiently compare the various stochastic solutions with a reference Monte-Carlo simulation based on 500000

Table 6.1. *Distributions and statistical parameters of the input uncertain variables.*

α	Distribution	Parameter 1	Parameter 2	Group
a	Normal	1,75 (mean)	1,0 % (CoV)	G
b	Normal	0,699 (mean)	1,0 % (CoV)	G
c	Normal	0,8 (mean)	1,0 % (CoV)	G
d	Normal	4,0 (mean)	1,0 % (CoV)	G
p_T	Uniform	0,9 (low bound)	1,1 (up bound)	O
T_T	Uniform	0,9 (low bound)	1,1 (up bound)	O
γ	Uniform	1,39 (low bound)	1,40 (up bound)	T

samples, which ensures full convergence of the output statistics. Several sources of uncertainties, summarized in Table 6.1, are taken into account. They include geometric uncertainties (group G), operating conditions (group O), and thermodynamic uncertainties (group T). In this study, the first group of uncertainties is represented by the four coefficients of eq. (6.20), the second one comprises the reservoir pressure p_T and temperature T_T, and the latter includes the specific heat ratio γ. Geometric uncertainties are assumed as normally distributed, with given mean and coefficient of variation (CoV $= \frac{\sigma}{\mu}$); the other uncertain parameters are assumed to be uniformly-distributed with imposed lower and upper bounds.

The range of variation of the total pressure is such that a shock is always created in the divergent. In the following, we consider a robust design problem where the average pressure distribution along the nozzle provided by the stochastic flow solver is required to match a target pressure distribution. Precisely, the cost function is defined as the mean quadratic error with respect to the target all over the nozzle:

$$J = \frac{1}{2}\sum_{i=1}^{n}(p_i - p_{target})^2 \tag{6.21}$$

where p_i represents the pressure at the ith mesh cell. We require the mean value of J to be zero, its variance being as low as possible. The robust optimization problem is then set as follows:

$$\text{Find} \quad \overline{S}(x) \quad \text{such that:} \quad E[J(\overline{S})] = 0, \quad Var[J(\overline{S})] = \min_{S} Var[J(S).]$$

In this work, the two statistical moments are minimized simultaneously by means of a MOGA in a robust optimization loop; the process iteratively modifies the values of the coefficients a, b, c, d of equation (6.20) until convergence.

6.5.2 Assessment of the AMoM as an UQ method

This Session assesses the capabilities of the AMoM as UQ method against other methods commonly used in the literature. Initially, the AMoM is applied to the model problem described in the preceding section to estimate the low-order statistics (mean and variance) of the QoI (6.21) under multiple uncertainties. Then the capability of AMoM of efficiently providing accurate information on the contribution of each uncertain parameter to the global variance of a QoI is investigated. This approach, close to analysis of variance (ANOVA), can provide useful information for reducing the size of a stochastic problem, i.e. neglecting its dependencies on the less influential uncertain parameters, while preserving a high level of accuracy of the output statistical moments.

First, we assess the UQ capabilities of the AMoM for two cases. The first case consists in computing the moments of functional J (6.21) by taking into account all of the variables given in Tab. 6.1 as uncertain parameters; in the second case, only uncertainties belonging to group G are taken into account. The results are summarized in Tables 6.2 and 6.3. The computations were carried out with both the first-order (FOSM) and the second-order (SOSM) method of moments. Nevertheless, for the former method, J evaluated at the design point is equal to zero, as well as the first-order derivatives. Hence, statistical moments computed according to eqs. (6.2), (6.3) are identically null. For this reason, FOSM was discarded and only SOSM computations are presented in the following. Moreover, in the SOSM method the first and second-order derivatives of J were computed using both the proposed Adjoint Method and finite differences. The results obtained using second-order NIPC on both structured (NIPC2QFF) and unstructured full grids (NIPC2LHS), accurate sparse grid approach (NIPC2QS), second- and third-order PCM (PCM2 and PCM3), and Monte Carlo sampling (MC) are also reported for comparison.

When all groups of uncertain parameters are taken into account simultaneously, SOSM provides high error levels with respect to the reference Monte Carlo computation, compared to NIPC and PCM. Results of the AMoM are very similar to those of the finite difference SOSM, apart from minor numerical errors. Nevertheless, the second one is 16 times more expensive, for this highly dimensional stochastic problem. In spite of its relative inaccuracy, the computational cost of AMoM is much lower (one to five orders of magnitude) compared to the other methods. The inaccuracy of AMoM for this case is essentially due to the fact that small fluctuations of the reservoir pressure (parameter p_T) lead to large displacements of the shock wave across the nozzle divergent. Such large displacement cannot be approximated accurately by means of Taylor-series expansions.

Table 6.2. *Results of UQ analysis for 6 uncertainties.*

Method	Error μ (%)	Error σ (%)	Time (s)
SOSM fin. diff.	32,25	56,34	80
SOSM adjoint	32,25	56,30	5
PCM2	4,09	35,53	60
PCM3	2,29	3,35	700
NIPC2QFF	3,74	36,20	54
NIPC2LHS	1,36	2,25	54
NIPC2QS	1,43	1,42	98
Random MC	0,0	0,0	10^5

Table 6.3. *Results of UQ analysis for 4 uncertainties.*

Method	Error μ (%)	Error σ (%)	Time (s)
SOSM fin. diff.	9,91	16,93	40
SOSM adjoint	10,13	17,00	5
PCM2	3,28	17,32	15
PCM3	1,43	1,35	75
NIPC2QFF	3,57	17,53	15
NIPC2LHS	2,02	7,17	15
NIPC2QS	1,34	1,66	40
Random MC	0,0	0,0	10^5

In fact, for the second case, where only geometrical uncertainties are taken into account, the accuracy of the SOSM method increases greatly, even if error levels remain large compared to other methods, because of the presence of a shock wave, that invalidates the use of Taylor-series expansions locally. The computational cost of AMoM is still much lower (3 to 25 times) than NIPC or PCM methods, and orders of magnitude lower than that of Monte Carlo sampling.

Computations (not shown here) were also run for a smooth flow problem (no shock in the divergent). In this case, AMoM results are in excellent agreement with Monte Carlo ones for both cases (all uncertainty groups or only G group taken into account), for a computational cost that is drastically lower than any other method.

In summary, AMoM allows for a cheap but rough estimate of low-order moment of the QoI, its accuracy being greatly reduced for non-smooth flows with large shock fluctuations.

Then, we investigate the capability of AMoM of efficiently predicting the

contribution of a given uncertain parameter to the randomness of a QoI, in order to rank the importance of input uncertainties. To this purpose, analysis of variance (ANOVA) is usually adopted to preliminarily quantify the contributions of each uncertainty to the global variance of the QoI.

In the following, we use information available via the AMoM to calculate the so-called importance factors of each uncertain parameter, expressed as:

$$\chi_i = \frac{\sigma^2(J)\mid_i}{\sigma^2(J)} \tag{6.22}$$

Indicators defined by (6.22) measure the fraction of variance of a QoI that is due to a specific input variable. They are calculated afterwards the automated coupling procedure, using available information about output variance.

The results are compared with those of a classical ANOVA analysis using NIPC based on sparse grid integration (NIPC2QS) and the corresponding variance based decomposition, according to the Sobol's indexes[94].

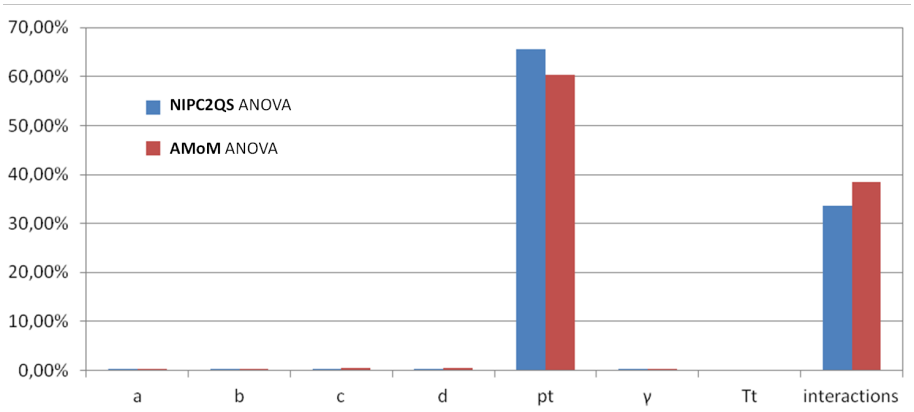


Figure 6.3. *Results of the adjoint-based and sparse grid integration preliminary ANOVA.*

Figure 6.3 depicts the results of the ANOVA analysis for the whole set of parameters listed in Table 6.1. Both the NIPC2QS and AMoM approach give similar trends. Nevertheless the adjoint-based methodology requires a fraction of the sparse grid computational cost (of the order of 10^{-1}), and appears to be much more efficient for stochastic problems with multiple uncertainties. Hystograms displayed in Figure 6.1 clearly show that the highest contribution to the variance of the fitness function J is due to the reservoir

pressure. Interaction effects between p_T and the remaining uncertain variables (represented by the bars at the right) are correctly captured by the adjoint-based approach. Since the main objective of the ANOVA is to rank the parameters according to their relative impact on the variance of the QoI, more than obtaining the exact importance factors, the present investigation demonstrates the interest of AMoM for quick ANOVA analysis prior to more detailed UQ or robust optimization calculations.

6.5.3 Robust Design of an Inverse Flow Problem

Results shown in the preceding Section show that the AMoM enables for a cheap estimate of low-order moments of a QoI, even if with lower accuracy compared to other UQ methods, namely for shock-dominated flows. In the following, we investigate the suitability of AMoM as UQ method within a robust optimization loop: despite the fact that statistical moments are not evaluated as accurately as by other UQ methods AMoM is still expected to provide proper and cheap information of search for optimization purposes.

In the following, we apply AMoM to the solution of the inverse flow problem under uncertainty described in subsection V.A. For this purpose, the mean $E[J]$ and the variance $(\sigma[J])^2$ of the quadratic error with respect to the target pressure are simultaneously minimized by means of a MOGA (see Section IV). The optimization is carried out for both cases discussed in the preceding Section, i.e. by taking into account all of the uncertain input parameters (case 1) or by considering only group G uncertainties (case 2). Statistical distributions used for random parameters are those of Table 6.1. Hereafter, we discuss results obtained by using AMoM coupled with a MOGA (referred as AMoM-MOGA) compared to several robust optimization strategies:

1. Second-order NIPC based on full factorization of the collocation points coupled to a MOGA (NIPC2QFF-MOGA).

2. NIPC using accurate sparse grid quadrature, coupled to a MOGA (NIPC2QS-MOGA).

3. Two-step strategy based on preliminary selection of the most influential uncertain parameters via ANOVA, then robust optimization by applying NIPC2QFF to the reduced set of random parameters along with MOGA (2STEP). For the present problem, ANOVA shows that p_T is largely the most influential variable, and the only one that should be retained for subsequent optimization, as more than 90% of the fitness variance is due to total pressure. For case 2, where only geometric uncertainties are considered, the most influential parameters are the

geometrical coefficients c and d, so that a, b are taken as deterministic in the optimization loop.

Approach 1 is considered hereafter as the reference in terms of accuracy, since NIPC2QFF ensures good accuracy compared to Monte-Carlo, at least for expectancy estimates. Approach 2 still provides accurate estimates of the statistics with reduced computational cost with respect to NIPC2QFF. Finally, approach 3 drastically reduces computational cost, at the expense of neglecting most of the sources of uncertainty.

For all optimization runs, a fully converged Pareto front is obtained by means of the MOGA. Then, to investigate the possibility of improving the best configuration of the AMoM-MOGA approach, a second optimization step is applied; the solution is then refined by a local search algorithm, coupling the second-order AMoM with a BFGS method (referred to AMoM-BFGS). To make it compatible with a gradient-based algorithm, the bi-objective optimization problem is reformulated in a single-objective one by constructing the combined cost function:

$$J = \lambda\mu + (1 - \lambda)\sigma \tag{6.23}$$

In a similar fashion, the gradient becomes:

$$\boldsymbol{\nabla} J = \lambda\boldsymbol{\nabla}\mu + (1 - \lambda)\boldsymbol{\nabla}\sigma \tag{6.24}$$

where λ has been fixed to 0.5, and $\boldsymbol{\nabla}\mu, \boldsymbol{\nabla}\sigma$ are calculated by deriving the first order expectancy and variance equations, namely relations (6.2), (6.3) with respect to uncertain parameters. As a consequence second-order derivatives are used to estimate the gradient of the cost function.

The resulting optimal solutions given by the overall optimization process are subsequently validated *a posteriori* by computing their fitness functions by means of a third order polynomial chaos.

Simulation results, at the end of the validation process, are provided in Tables 6.4 and 6.5. The last column provides computational costs of the simulations with the MOGA. The gradient-based refinement applied to the AMoM-MOGA best individual, that typically takes about one hundred seconds, slightly improves the objectives. As a consequence solutions of the genetic solver are considered as the reference and the second optimization step is not accounted for hereafter. The AMoM-MOGA approach is found to be far cheaper than the other methods; the cost is also nearly independent from the number of uncertainties taken into account. The last rows of Tables 6.4 and 6.5 also provide the mean and variance of the cost function for a geometry obtained by means of a deterministic optimization. Deterministic design offers poor off-design behavior compared to robust optimization approaches, the standard deviation of the error with respect to target pressure

Table 6.4. *Robust design results for 6 uncertainties.*

Method	Mean	Std. Dev.	a	b	c	d	Time (s)
NIPC2QFF-MOGA	2.79e-01	2.54e-01	1.503	0.719	0.762	4.04	20000
NIPC2QS-MOGA	2.86e-01	2.59e-01	1.507	0.719	0.755	3.99	3400
2STEP	2.81e-01	2.54e-01	1.505	0.714	0.778	4.09	800
AMoM-MOGA	2.79e-01	2.53e-01	1.502	0.719	0.762	4.04	1450
AMoM-BFGS	2.75e-01	2.50e-01	1.5	0.73	0.759	4.04	130
deterministic	3.64e-01	3.26e-01	1.75	0.699	0.8	4.0	-

being approximately 50% larger. For both optimization cases (6 uncertainties or 2 G-uncertainties), all robust optimization approaches return very similar results in terms of mean, standard deviations, and design variables; the resulting nozzle profiles are superposed to within plotting accuracy, see Figure 6.4.

The computational cost of the NIPC2QFF methodology is about one order of magnitude greater than that of the other approaches, due to the highly expensive estimate of low-order moments by full factorial polynomial chaos (64 deterministic simulations for design required). As a consequence it is not suitable for robust designs of realistic problems involving a moderate number of random parameters, unless massive parallelization is applied like in [106]. The cost drastically decreases by adopting sparse grid techniques, since mean and standard deviation are evaluated after a lower number of deterministic calculations (13 for each individual of the genetic algorithm). The two-step strategy, leading to just 1 or 2 deterministic simulations for each individual, is the more efficient one for the present test case. Its accuracy and efficiency strongly relies on the preliminary analysis of variance: to this aim, AMoM provides the required information quickly and accurately.

The second series of runs (4 uncertainties) leads to similar conclusions. Note that the optimal nozzle contour, sketched with a red line, is completely different from that obtained by taking into account fluctuations of the total pressure. This confirms again the high impact of this parameter on the solution. Differently from the previous case, the NIPC2QS-MOGA and the 2STEP methodologies require now a similar computational effort as NIPC2QFF-MOGA, as only two variables, a, b, over a total of four, are omitted in the uncertainty analysis. The NIPC2QFF-MOGA strategy remains nevertheless the most costly.

Table 6.5. *Robust design results for 4 uncertainties.*

Method	Mean	Std. Dev.	a	b	c	d	Time (s)
NIPC2QFF-MOGA	5.04e-02	6.61e-02	1.986	0.709	0.83	4.03	4300
NIPC2QS-MOGA	5.04e-02	6.62e-02	1.916	0.69	0.83	4.01	2400
2STEP	5.05e-02	6.64e-02	1.915	0.704	0.83	4.04	2100
AMoM-MOGA	5.05e-02	6.62e-02	1.996	0.705	0.83	3.99	1400
AMoM-BFGS	5.04e-02	6.61e-02	1.996	0.706	0.83	3.983	125
deterministic	5.09e-02	6.77e-02	1.75	0.699	0.8	4.0	-

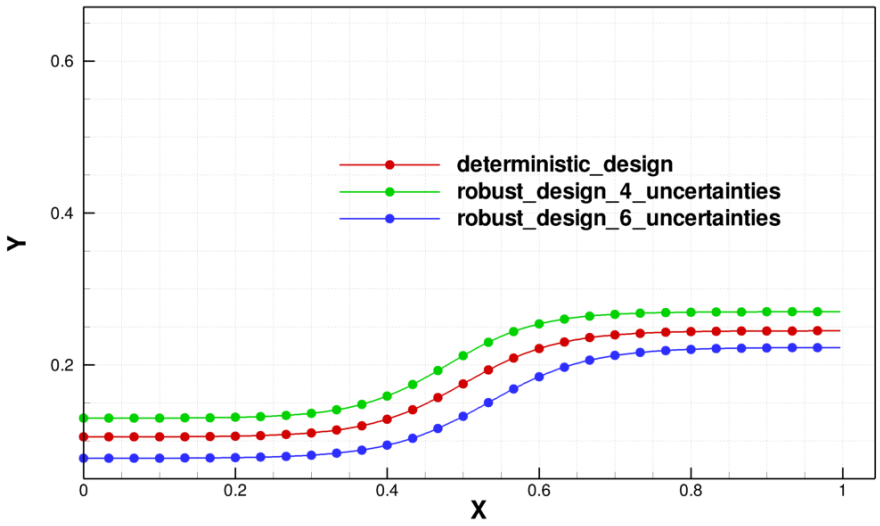


Figure 6.4. *Baseline and robust nozzle contours for the inverse flow problem.*

6.6 Conclusive Remarks

This chapter has illustrated an interesting application of the adjoint for UQ and robust design approaches in CFD. In particular a novel robust design approach, based on an Adjoint Method of Moments (AMoM) combined with a Multi-Objective Genetic Algorithm (MOGA) has been proposed. The performances of AMoM have been assessed for a transonic flow problem against with some widely-used stochastic methods (Monte-Carlo, NIPC, and PCM). The first- and second-order sensitivity derivatives have been computed by both an adjoint approach and finite differences.

For problems characterized by large fluctuations of the input parameters, and in the transonic regime (where shock waves are present in the flow), the method of moments provided high error levels with respect to the other UQ

techniques taken into account (Monte-Carlo, NIPC and PCM). This defect is related to the intrinsic nature of the method, which makes use of Taylor-series expansion. Nevertheless, in all numerical tests carried out, the use of the adjoint formulation leads to a dramatic reduction of computational cost. As a consequence the AMoM appears to be a suitable candidate for quick estimate of the statistical moments for stochastic problems involving an arbitrarily large number of random parameters.

Subsequently, the AMoM was coupled with a multi-objective genetic solver (MOGA) to build a robust design strategy, and the resulting hybrid robust optimization algorithm (AMoM-MOGA) was applied to an inverse design exercise for transonic 1-D flows through a diverging channel.

Numerical tests show the interest of the proposed methodology in terms of solution accuracy and computational burden against other investigated robust strategies. The final results point out that the coupling between AMoM and a genetic solver makes an efficient robust optimization tool, despite the scant accuracy of expectancy and standard deviation provided by the uncertainty propagation method in case of non-smooth flows. Present numerical experiments show that the AMoM-based robust optimization returns results that are close to those obtained by using NIPC as UQ method in terms of accuracy (final values of the objective functions and of the design variables), with a much lower computational expense. The latter is shown to be almost independent on the number of uncertain parameters that are taken into account. In conclusion, AMoM-based robust optimization is an effective and promising design tool, that is worth further investigations for more complex applications.

Chapter 7

Conclusions and Future Perspectives

This research work has presented an efficient optimization framework for improving the design of fluid-dynamic devices operating under ideal and real gas flows. The iterative update of the baseline configuration has been obtained by means of a gradient method based on a discrete inviscid adjoint, state-of-the-art parameterization techniques and a preconditioned steepest descent searching algorithm. Differently from standard approaches, the present methodology is capable to effectively deal with real-gas effects adopting, alternatively, built-in equations of state or a fast look-up table method. The last one results in a non-intrusive coupling between the numerical solvers and the thermodynamic library, allowing for the inclusion of any thermodynamic models (provided by the library) into the design loop with a relatively low effort.

The geometry of interest is deformed according to surface gradients, i.e. gradients with respect to NURBS control points, or active gradients, i.e. gradients with respect to active boundary mesh nodes. Whenever necessary, the smoothness of the sensitivities can be augmented by recurring to elliptic smoothing techniques applied at mesh level.

Test cases have concerned the optimization of wind tunnel nozzles and turbine cascades operating under flows characterized by relevant real-gas effects. The re-design of a supersonic cascade for ORC applications has been proposed as original test case. To address off-design conditions, a standard multi-point methodology has been coupled to a stochastic algorithm for the choice of weights and samples in which evaluating the objective function. An ANOVA analysis has been also used to preliminarily select the most influent uncertain parameters.

The results achieved generally proved a quick convergence of the design algorithm to the local minimum. Regarding the supersonic turbine opti-

mization, considerable improvements of the cascade performance have been obtained by weakening the shocks appearing in the initial configuration, in reference as well as at off-design conditions. The off-design performances of the optimized blades have been finally investigated from a probabilistic perspective to demonstrate the validity of the multi-point strategy. The outcomes of the stochastic analysis has revealed the potential of the proposed multi-point optimization as a viable strategy to improve the robustness of supersonic turbomachinery cascades exposed to relevant variations of the operating conditions.

The work conducted in this research has led to a better understanding of the adjoint method for shape optimization and uncertainty quantification algorithms applied to aerodynamics and turbomachinery problems. The issues encountered during this work suggested a number of possible ideas to improve the quality of the research. Among these, to the author's opinion, the most favorable paths for future new achievements can be oriented in the following directions:

- A comprehensive analysis on gradient accuracy is planned as a first step to further consolidate the results achieved in this work. In this perspective, a thorough study of the effect of gradient approximation on the optimized layout, currently ongoing, will be extensively performed. In fact there are still a few studies on this topic in the literature.

- Viscous and turbulent adjoints will be considered. Precisely, algebraic turbulence models could be the first candidates to be fully differentiated and embedded within design loop. The capability of matrix-free techniques for solving viscous and turbulent adjoint problems remains one of the trickiest issues of the method. Indeed, numerical studies on the convergence of these methods for advection-diffusion problems could be of paramount importance for the whole adjoint community.

- The optimization method could be extended to handle explicit constraints using the Lagrangian formulation. In addition, state-of-the-art optimization solvers, such as BFGS or SQP, could be taken into account to compare their performances against the ones offered by the steepest-descent method.

- A comparison between the present robust multi-point approach and fully stochastic design strategies could be of great interest to point out advantages and disadvantages of both solutions. As an example, the hybrid algorithm for stochastic optimizations proposed in the last chapter represents a valuable strategy to efficiently manage uncertain

operating conditions. Meta-modeling techniques could also offer a viable contribution for enhancing the efficiency of robust optimization methods.

References

[1] D. Thevenin and G. Janiga. *Optimization and Computational Fluid-Dynamics*. Springer, 2005.

[2] D. Pasquale. *Turbomachinery Design Optimization based on CFD, Metamodels and Evolutionary Algorithms*. PhD thesis, 2012.

[3] M.M. Rai. Aerodynamic Design Using Neural Networks. *AIAA Journal*, 38(1):173–182, 2000.

[4] M.M. Rai and N.K. Madavan. Application of artificial neural networks to the design of turbomachinery airfoils. *Journal of Propulsion and Power*, 17(1):176–183, 2001.

[5] M.M. Rai. Three-Dimensional Aerodynamic Design Using Artificial Neural Networks. Technical Report January, AIAA-2002-0987, 2002.

[6] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.

[7] S. Zhen-Jun. Convergence of line search methods for unconstrained optimization. *Applied Mathmatics and Computation*, 157(2):393–405, 2004.

[8] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford Science Publications, 2010.

[9] M. Pini, G. Persico, E. Casati, and V. Dossena. Preliminary design of a centrifugal turbine for ORC applications. *ASME Journal of Engineering for Gas Turbines and Power*, 135(042312), 2013.

[10] M. V. Petrovic, G. S. Dulikravich, and T. J. Martin. Optimization of multistage turbines using a through-flow code. *Journal of Power and Energy*, 215(5):559–569, 2001.

[11] F. Larocca. Multiple objective optimization and inverse design of axial turbomachinery blade. *Journal of Propulsion and Power*, 24(5):1093–1099, 2008.

[12] D. Pasquale, G. Persico, and S. Rebay. Optimization of Turbomachinery Flow Surfaces Applying a CFD-Based Troughflow Method. *Journal of Turbomachinery*, 136(3):031013 1–11, 2014.

[13] S. Pierret and R.A. Van Den Braembussche. Turbomachinery Blade Design Using a Navier-Stokes Solver and Artificial Neural Network. *Journal of Turbomachinery*, 121(2):326, 1999.

[14] O. Leonard and R.A. Van Den Braembussche. Design Method for Subsonic and Transonic Cascade With Prescribed Mach Number Distribution. *Journal of Turbomachinery*, 114(3):553, 1992.

[15] A. Demeulenaere, O. Leonard, and R.A. Van Den Braembussche. A two-dimensional Navier-Stokes inverse solver for compressor and turbine blade design. *Journal of Power and Energy*, 211(4):299–307, jan 1997.

[16] C.A.C. Coello. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2):109–143, 2000.

[17] T. Verstraete, Z. Alsalihi, and R.A. Van Den Braembussche. Multidisciplinary Optimization of a Radial Compressor for Microgas Turbine Applications. *Journal of Turbomachinery*, 132:031004, 2010.

[18] D. Pasquale, A. Ghidoni, and S. Rebay. Shape Optimization of an Organic Rankine Cycle Radial Turbine Nozzle. *ASME Journal of Engineering for Gas Turbines and Power*, 135(4):042308 (13 pages), 2013.

[19] S. Y. Lee and K. Y. Kim. Design optimization of axial flow compressor blades with three-dimensional Navier-Stokes solver. *ASME International Journal*, 14(9):1005–1012, 2000.

[20] A. Oyama. Transonic Axial-Flow Blade Optimization: Evolutionary Algorithms/Three-Dimensional Navier-Stokes Solver. *Journal of Propulsion and Power*, 20(4):612–619, 2004.

[21] N. Chen, H. Zhang, W. Huang, and Y. Xu. Study on aerodynamic design optimization of turbomachinery blades. *Journal of Thermal Science*, 14(4), 2005.

[22] S. Pierret, R.F. Coelho, and H. Kato. Multidisciplinary and multiple operating points shape optimization of three-dimensional compressor blades. *Structural and Multidisciplinary Optimization*, 33(1):61–70, 2006.

[23] J.E.V. Peter and R.P. Dwight. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers & Fluids*, (39):373–391, 2010.

[24] M. Ferlauto A. Iollo and L. Zannetti. An Aerodynamic Optimization Method Based on the Inverse Problem Adjoint Equations. *Journal of Computational Physics*, page 29, 2001.

[25] D.I. Papadimitriou and K.C. Giannakoglou. Compressor Blade Optimization Using a Continuous Adjoint Formulation. In *ASME Turbo-Expo 2006*, page 9, 2006.

[26] M.C. Duta, S. Shahpar, and M. Giles. Turbomachinery Design Optimization Using Automatic Differentiated Adjoint Code. In *ASME Turbo-Expo 2007*, page 10, 2007.

[27] J. Luo, J. Xiong, F. Liu, and I. McBean. Three-Dimensional Aerodynamic Design Optimization of a Turbine Blade by Using an Adjoint Method. *Journal of Turbomachinery*, 133:11, 2011.

[28] S. Shankaran, A. Marta, P. Venugopal, B. Barr, and Q. Wang. Interpretation of Adjoint Solutions for Turbomachinery Flows. In *ASME Turbo Expo 2012*, number 85, page 14, 2012.

[29] I. Obernberger, H. Carlsen, and F. Biedermann. State-of-the-art and future developments regarding small-scale biomass CHP systems with a special focus on ORC and Stirling engines technologies. In *International Nordic Bioenergy Conference*, 2003.

[30] E. Prabhu. Solar trough organic rankine electricity system (STORES). Stage 1. Power plant optimization and economics. Technical Report NREL/SR-550-39433, National Renewable Energy Laboratory (US), 2006.

[31] A. Verneau. Emploi des fluids organiques dans les turbines solaires (The use of organic fluids in the solar turbines). *Entropie*, 82:9–18, 1978.

[32] G. Angelino, M. Gaia, and E. Macchi. A review of Italian activity in the field of organic Rankine cycles. In *International VDI-Seminar (ORC-HP-Technology, Working Fluid Problems)*, pages 465–482, 1984.

[33] H. Tabor and L. Bronicki. Establishing criteria for fluids for small vapor turbines. In *SAE National Transportation, Powerplant, and Fuels and Lubricants Meeting*, October 1964.

[34] P. Colonna, A. Guardone, N. R. Nannan, and C. Zamfirescu. Design of the dense gas flexible asymmetric shock tube. *ASME Journal of Fluids Engineering*, 130, 2008.

[35] A. Spinelli, V. Dossena, P. Gaetani, C. Osnaghi, and D. Colombo. Design of a Test Rig for Organic Vapours. In *ASME Turbo Expo 2010*, 2010.

[36] D. Pasquale, A. Ghidoni, and S. Rebay. Shape Optimization of an Organic Rankine Cycle Radial Turbine Nozzle. *Journal of Engineeering for Gas Turbines and Power*, 2013.

[37] P. Cinnella and P.M. Congedo. GA-Hardness of Dense-Gas Flow Optimization Problems. 2006.

[38] T.R. Bewley. Flow control: new challenges for a new Renaissance. *Progress in Aerospace Sciences*, 37(1):21–58, 2001.

[39] M. Martinelli and R. Duvigneau. On the use of second-order derivatives and metamodel-based Monte-Carlo for uncertainty estimation in aerodynamics. *Computers & Fluids*, (39):953–964, 2011.

[40] D. Ghate and M.B. Giles. Inexpensive Monte Carlo Uncertainty Analysis. 2005.

[41] M. Giles and N. Pierce. An introduction to the adjoint approach to design. Technical report, Oxford Computing Laboratory, 2000.

[42] L. Hascoet and V. Pascual. *Tapenade 2.1 user's guide*, 2004.

[43] E. Nielsen and M. Park. Using An Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design. *AIAA Journal*, page 10, 2006.

[44] G.E. Farin. *Curves and surfaces for CAGD: a practical guide.* Academic Press, 5^{th} edition, 2002.

[45] J. Hoschek, D. Lasser, and L.L. Schumaker. *Fundamentals of computer aided geometric design.* A.K. Peters, 1996.

[46] C. Degand and C. Farhat. A three dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers & Structures*, 80(3-4):305–316, 2002.

[47] A. de Boer, M.S. van der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers & Structures*, (85):784–795, 2007.

[48] S. Kim, Obayashi S., and K. Nakahashi. Aerodynamic Optimization of Supersonic Transport Wing using Unstructured Adjoint Method. *AIAA Journal*, (39):1011–1020, 2001.

[49] V. Selmin. The node-centered finite volume approach: bridge between finite differences and finite elements. *Comput. Methods Appl. Mech. Eng.*, (102):107–138, 1993.

[50] V. Selmin and L. Formaggia. Unified construction of finite element and finite volume discretizations for compressible flows. *Int. Journal Numer. Meth. Eng.*, (39):1–32, 1996.

[51] P. Colonna and S. Rebay. Numerical simulation of dense gas flows on unstructured grids with an implicit high resolution upwind Euler solver. *International Journal for Numerical Methods in Fluids*, 46:735–765, 2004.

[52] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, (43):357–362, 1981.

[53] M. Vinokur and J.L. Montagnè. Generalized flux-vector splitting and Roe average for an equilibrium real gas. *Journal of Computational Physics*, 89(2):276–300, 1990.

[54] Y. Saad and M. Schultz. Gmres: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. ScI. STAT. COMPUT.*, 24, issue 2:282–294, 1986.

[55] P. Colonna, S. Rebay, and P. Silva. Computer Simulations of Dense Gas Flows Using Complex Equations of State for Pure Fluids and Mixtures and State of the Art Numerical Schemes. Technical report, 2002.

[56] E. J. Nielsen, J. Lu, M. A. Park, and D. L. Darmofal. An implicit, exact dual adjoint solution method for turbulent flows on unstructured grids. *Computers & Fluids*, (33):24, 2004.

[57] N. Nemec and D.W. Zingg. Towards efficient aerodynamic shape optimziation based on the Navier-Stokes equations. In *15th AIAA Computational Fluid-Dynamics Conference* , page 12, 2001.

[58] . M. Giles, M. Duta, J-D Muller, and N. Pierce. Algorithm developments for discrete adjoint methods. *AIAA Journal*, (41):198–205, 2003.

[59] S. Kim, K. Hosseini, K. Leoviriyakit, and Jameson A. Enhancement of the Adjoint Design Methods via Optimization of Adjoint Parameters. *AIAA Journal*, 2005.

[60] A. Jameson and S. Kim. Reduction of the Adjoint Gradient Formula in the Continuous Limit. *AIAA Journal*, 2003.

[61] *Minfx Optimisation Library, version 1.05*, 2013.

[62] A. Ghidoni, E. Pelizzari, S. Rebay, and V. Selmin. 3D anisotropic unstructured grid generation. *Int. J. Numer. Meth. Fluids*, (51):1097–1115, 2006.

[63] M. B. Kennel. KDTREE 2: Fortran 95 and C++ software to efficiently search for near neighbors in a multi-dimensional Euclidean space. 2004.

[64] S. Schmidt, C. Ilic, V. Schultz, and N. R. Gauger. Airfoil design for compressible inviscid flow based on shape calculus. *Optim. Eng.*, 12, 2011.

[65] R.P. Dwight and J. Brezillon. Effect of Approximations of the Discrete Adjointon Gradient-Based Optimization. *AIAA Journal*, (44):24, 2006.

[66] G. Carpentieri, M.J.L. van Tooren, and B. Koren. Comparison of exact and approximate discrete adjoint for aerodynamic shape optimization. In *ICCFD*, number 41, 2006.

[67] M. Nemec, D.W. Zingg, and T.H. Pulliam. Multipoint and Multi-Objective Aerodynamic Shape Optimization. *AIAA Journal*, 42(6):1057–1065, 2004.

[68] M.S. Eldred and J. Burkardt. Comparison of Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Quantification. *AIAA Journal*, 2009.

[69] J.D. van der Waals. *On the continuity of gaseous and liquid states*, volume XIV. North-Holland, Amsterdam, 1988, Reprinted.

[70] O. Redlich and J. N. S. Kwong. On the thermodynamics of solutions. V. An equation of state. Fugacities of gaseous solutions. *Chem. Rev.*, 44(1):233–244, 1949.

[71] J. J. Martin and Y. C. Hou. Development of an equation of state for gases. *A.I.Che. J.*, 1(2):142–151, June 1955.

[72] D. Y. Peng and D. B. Robinson. A new two-constant equation of state. *Ind. Eng. Chem. Fundam.*, 15(1):59–64, 1976.

[73] R. Span and W. Wagner. Equations of state for technical applications. I. simultaneously optimized functional forms for nonpolar and polar fluids. *International Journal of Thermophysics*, 24(1):1–39, January 2003.

[74] T.-H. Chung, M. Ajlan, L. L. Lee, and K. E. Starling. Generalized multiparameter correlation for non-polar and polar fluid transport properties. *Ind. Eng. Chem. Res.*, 27:671–679, 1988.

[75] J. Harinck, P. Colonna, A. Guardone, and S. Rebay. Influence of thermodynamic models in 2D flow simulations of turboexpanders. *ASME Journal of Turbomachinery*, 132:pp. 011001–17, 2010.

[76] M. Pini, A. Spinelli, V. Dossena, P. Gaetani, and F. Casella. Dynamic simulation of a test rig for organic vapours. In *ASME 5th International Conference on Energy Sustainability, ES 2011 (PARTS A, B, AND C)*, pages 1977–1988, 2011.

[77] C.R. Laughman, Y. Zhao, and D. Nikovski. Fast refrigerant property calculations using interpolation-based methods. In *International Refrigeration and Air Conditioning Conference*, page 10, 2012.

[78] R. Pecnik, E. Rinaldi, and P. Colonna. Computational Fluid Dynamics of a Radial Compressor Operating with Supercritical CO2. *ASME Journal of Engineering for Gas Turbines and Power*, 134:8, 2012.

[79] P. Boncinelli, F. Rubechini, A. Arnone, M. Cecconi, and C. Cortese. Real Gas Effects in Turbomachinery Flows: A Computational Fluid-Dynamics Model for Fast Computations. *ASME Journal of Turbomachinery*, 126:268–276, 2004.

[80] T. Saaresti-Turunen, J. Tang, and J. Larjola. A Practical Real-Gas Model in CFD. In *European Conference on Computational Fluid-Dynamics (ECCOMAS)*, page 9, 2006.

[81] P. Colonna and T. P. van der Stelt. FluidProp: a program for the estimation of thermo physical properties of fluids. Software, 2004.

[82] S. Rebay, P. Colonna, D. Pasquale, and A. Ghidoni. Simulation of the turbulent dense gas flow through the nozzle of an organic Rankine cycle turbine. In *8-th European Conference on Turbomachinery, Fluid dynamics and thermodynamics 23–27 March 2009, Graz-Austria*, volume

unico, pages 1137–1148. F. Heitmeir, F. Martelli, M. Manna, March 23ï¿½27 2009.

[83] G. Persico and S. Rebay. A Penalty Formulation for the Throughflow Modeling of Turbomachinery. *Computers and Fluids*, 60:86–98, 2012.

[84] A.D. Boger. Efficient Method for Calculating Wall Proximity. *AIAA Journal*, 39(12):2404–2406, 2001.

[85] B. Roget and J. Sitaraman. Wall Distance Search Algorithm Using Voxelized Marching Spheres. *Journal of Computational Physics*, 241:76–94, 2013.

[86] E. P. Gyftopoulos and G. P. Beretta. *Thermodynamics: Foundations and Applications*. MacMillan Publishing Company, New York, 1991.

[87] P. Colonna, N.R. Nannan, and A. Guardone. Multiparameter equations of state for siloxanes: $[(CH_3)_3\text{-Si-}O_{1/2}]_2\text{-}[O\text{-Si-}(CH_3)_2]_{i=1,3}$, and $[O\text{-Si-}(CH_3)_2]_6$. *Fluid Ph. Eq.*, 263:115–130, October 2008.

[88] R. Span and W. Wagner. A new equation of state for carbon dioxide covering the fluid region from the triple-point temperature to 1100 k at pressures up to 800 mpa. *J. Phys. Chem. Ref. Data*, 6(24):1509–1596, January 1996.

[89] H. R. M. Craig and H. J. A. Cox. Performance estimation of axial flow turbines. *Proceedings of the Institution of Mechanical Engineers*, 185:407–424, 1971.

[90] G. Persico, M. Pini, V. Dossena, and P. Gaetani. AERODYNAMIC DESIGN AND ANALYSIS OF CENTRIFUGAL TURBINE CASCADES. In *ASME Turbo Expo 2013*, number GT2013-95770, 2013.

[91] B. Adams, K. Dalbey, M. Eldred, L. Swiler, W. Bohnhoff, J. Eddy, and D. Vigil. *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification and Sensitivity Analysis, Theory Manual, version 5.2*, 2011.

[92] A. Guardone, A. Spinelli, and V. Dossena. Influence of Molecular Complexity on Nozzle Design for an Organic Vapor Wind Tunnel. *ASME Journal of Engineering for Gas Turbines and Power*, 135, 2013.

[93] G.J.A. Loeven and H. Bijl. Probabilistic Collocation in a Two-Step approach for efficient uncertainty quantification in computational fluid dynamics. *CMES Journal*, 36(3):193–212, 2008.

[94] I.M. Sobol. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55:271–280, 2001.

[95] M. Drela. Pros & Cons of Airfoil Optimization. *Frontiers of Computational Fluid Dynamics*, pages 363–381, 1998.

[96] P. Cinnella, P.M. Congedo, and L. Parussini. Quantification of thermodynamic uncertainties in real gas flows. *International Journal of Engineering Systems Modelling and Simulation*, 2(1-2):12–24, 2010.

[97] M. Pini and P. Cinnella. Hybrid Adjoint-based Robust Optimization Approach for Fluid-Dynamic Problems. In *15th AIAA Non-Deterministic Approaches Conference*, page 16, 2012.

[98] P. Colonna, J. Harinck, S. Rebay, and A. Guardone. Real-gas effects in Organic Rankine Cycle turbine nozzles. *Journal of Propulsion and Power*, 24, issue 2:282–294, 2008.

[99] P. Colonna, N. R. Nannan, A. Guardone, and W. Lemmon. Multiparameter equations of state for selected siloxanes. *Fluid Phase Equilibria*, 244:193–211, 2006.

[100] P. Colonna, A. Guardone, and N. R. Nannan. Siloxanes: a new class of candidate Bethe-Zeldovich-Thompson fluids. *Physics of Fluids*, 19, 2007.

[101] G. Persico, A. Mora, P. Gaetani, and M. Savini. Unsteady Aerodynamics of a Low Aspect Ratio Turbine Stage: Modeling Issues and Flow Physics. *ASME Journal of Turbomachinery*, 134:061030 (10 pages), November 2012.

[102] S.L. Padula, C.R. Gumbert, and W. Li. Aerospace applications of optimization under uncertainty. *Optimization and Engineering*, 7:317–328, 2006.

[103] P. Cinnella and S. Hercus. Robust optimization of dense gas flows under uncertain operating conditions. *Computers & Fluids*, pages 1893–1908, 2010.

[104] P.M. Congedo, J. Witteveen, and G. Iaccarino. Simplex-Simplex Approach for Robust Design Optimization. In *Evolutionary and Deterministic Methods for Design, Optimization and Control*, page 16, 2011.

[105] J. Axerio-Cilies, G. Iaccarino, G. Petrone, and V. Sellappan. Extreme ensemble computations for optimization under uncertainty. In *Evolutionary and Deterministic Methods for Design, Optimization and Control*, pages 12–24, 2011.

[106] S. Hercus and P. Cinnella. Robust shape optimization of uncertain dense gas flows through a plane turbine cascade. In *Proceedings of ASME-JSME-KSME Joint Fluids Engineering Conference*, 2011.

[107] R. Duvigneau. Aerodynamic Shape Optimization with Uncertain Operating Conditions using Metamodels. Technical report, 2007.

[108] P.M. Congedo, C. Corre, and J.M. Martinez. Shape optimization of an airfoil in a BZT flow with multiple source uncertainties. *Comput. Methods Appl. Mech. Engrg.*, (200):216–232, 2011.

[109] P.M. Congedo, S. Hercus, P. Cinnella, and C. Corre. Efficient robust optimization techniques for uncertain dense gas flows. In *Proceedings of CFD and OPTIMIZATION-ECCOMAS*, number 68, 2011.

[110] D. Xiu and G. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, (187):137–167, 2003.

[111] G. Onorato, G.J.A. Loeven, G. Ghorbaniasl, H. Bijl, and C. Lacor. Comparison of Intrusive and Non-intrusive Polynomial Chaos Methods for CFD Applications in Aeronautics. In *Proceedings of ECCOMAS CFD*, 2010.

[112] Uncertainty propagation in CFD using polynomial chaos decomposition. *Fluid Dynamics Research*, 38:616–640, 2006.

[113] G.J.A. Loeven, J. Witteveen, and H. Bijl. Probabilistic Collocation: An Efficient Non-intrusive Approach for Arbitrarily Distributed Parametric Uncertainties. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007.

[114] R. Abgrall, P.M. Congedo, and G. Geraci. On the use of the Sparse Grid techniques coupled with Polynomial Chaos. Technical report, INRIA Bordeaux - Sud Ouest, 2011.

[115] X. Yang, M. Choi, G. Lin, and G. Karniadakis. Adaptive ANOVA decomposition of stochastic incompressible and compressible flows. *Journal of Computational Physics*, 231:1587–1614, 2010.

[116] H.-G. Beyer and B. Sendhoff. Robust Optimization - A Comprehensive Survey. *Comput. Methods Appl. Mech. Engrg.*, (196):3190–3218, 2007.

[117] A. Belme, M. Martinelli, L. Hascoet, V. Pascual, and A. Dervieux. AD-based perturbation methods for uncertainties and errors. 2009.

[118] Jameson A. Optimum Aerodynamic Design Using CFD and Control Theory. *AIAA Journal*, page 24, 1995.

[119] . M. Giles and D. Ghate. Hessian Calculation Using AD. Technical report, Oxford Computing Laboratory, 2006.

[120] D. Xiu and G. Karniadakis. The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations. Technical report, Brown University, 2003.

[121] M.S. Eldred and J. Burkardt. Comparison of Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Quantification. *AIAA Journal*, 2009.

[122] J. Witteveen and G. Iaccarino. Simplex Elements Stochastic Collocation in Higher-Dimensional Probability Spaces. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2010.

[123] R.J. Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge texts in applied mathematics edition, 2002.

[124] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta. *AIAA Journal*, 81, 1981.

[125] HR. Byrd, P. Lu, J. Nocedal, and C. Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. Technical report, Technical Report NAM-08, 1994.

[126] P. Cinnella and P.M. Congedo. GA-Hardness of Dense-Gas Flow Optimization Problems. In *15th IASTED International Conference on Applied Simulation and Modeling*, 2006.