

Politecnico di Torino



von KARMAN INSTITUTE FOR FLUID DYNAMICS

Master of Science in Aerospace Engineering

Academic Year 2025/2026

Development of a Data-Driven Framework for the Design and Multi-Objective Optimization of Transonic Turbine Cascades

---

Supervisors:
Prof. Andrea Ferrero
Prof. Sergio Lavagnoli - VKI

Co-supervisor:
Francesco Porta - VKI

Candidate:
Alessandro Pescosolido

Graduation Session March 2026

Thesis carried out in collaboration with the von Karman Institute for Fluid Dynamics

Acknowledgements

Ringrazio il Professor Andrea Ferrero per la disponibilità e la gentilezza dimostrate durante i corsi e in questi mesi di tesi, e soprattutto per avermi dato l'opportunità di vivere un'esperienza straordinaria al von Karman Institute.

Ringrazio Sergio per il supporto costante, la fiducia e la pazienza con cui mi ha accompagnato lungo tutto questo percorso.

Ringrazio Francesco per i consigli preziosi — lavorativi e di vita — e per essere stato, più che un advisor, un amico.

Abstract

Turbomachinery remains a cornerstone technology for aviation, power generation, and industrial energy conversion. Aerodynamic design of turbine cascades is particularly challenging in the transonic regime, where supersonic regions, shock waves, and shock-boundary-layer interactions strongly affect performance, often leading to physically inconsistent designs or non-convergent simulations. These extreme non-linearities make brute-force design space exploration impractical even with fast 2D solvers.

This thesis develops a comprehensive data-driven pipeline for the rapid preliminary design and optimization of transonic 2D turbine cascades. A neural network surrogate is trained on a massive database of inviscid-viscous coupled simulations performed with `MISES`, learning to predict near-optimal blade geometries across a structured grid of 125 aerodynamic duty points defined by (α_1, α_2, M_2, Re). Blade geometries are parametrized using `ParaBlade`, an open-source constructive NURBS-based framework relying on 21 conventional geometric design variables.

A central contribution is a tailored Design of Experiments (DoE) strategy that ensures geometric feasibility. By extracting the Spearman rank-correlation structure from 30 transonic seed blades, the Iman-Conover algorithm and conditional sampling from a multivariate Gaussian distribution are used to generate realistic blade populations instantaneously.

The DoE database was evaluated and expanded through 1.25 million CFD simulations coupled with a multi-objective evolutionary algorithm (NSGA-II). By simultaneously minimizing kinetic energy loss (ξ_{kin}) and maximizing blade pitch (t/c_{ax}), distinct Pareto fronts of non-dominated designs were generated for each duty point. A Multi-Layer Perceptron (MLP) was subsequently trained on 12,500 uniformly binned samples extracted from these Pareto and near-Pareto sets.

Validation demonstrates robust predictive capabilities: the surrogate achieved an RMSE on the outlet flow angle ($\Delta\alpha_2$) of 0.360° on seen duty points, and 0.597° on unseen conditions. The model successfully recovered complex transonic phenomena, such as the transition from front-loaded to back-loaded distributions when sweeping the exit Mach number. Ultimately, this framework predicts near-optimal transonic geometries with negligible computational cost compared to direct CFD optimization, providing a highly effective tool for preliminary turbine design.

Keywords: Transonic turbine cascades, Multi-objective optimization, Design of Experiments, Blade parametrization, Surrogate modelling, Machine Learning, Computational Fluid Dynamics.

Table of Contents

1 Literature Survey 1

1.1 Literature Review . . . 1

1.1.1 Machine Learning and Surrogate Modeling in Aerodynamics . . 1

1.1.2 Forward and Inverse Design Strategies . . . 2

1.1.3 Challenges in Transonic Turbine Aerodynamics . . . 3

1.1.4 Geometric Parameterization and Sampling Strategy . . . 5

1.1.5 Optimization of Transonic Turbine Cascades . . . 6

2 Problem Formulation 9

2.1 Transonic Turbine Design . . . 9

2.1.1 Physical Challenges in the Transonic Regime . . . 10

2.1.2 Geometric Constraints on Shock Formation . . . 12

2.2 Design Objective and Duty Space . . . 14

2.2.1 Extension from Porta's Inverse Design Framework . . . 16

2.3 Methodology Overview . . . 20

2.3.1 Multivariate Gaussian Sampling . . . 20

2.3.2 MISES CFD Solver . . . 21

2.3.3 NSGA-II Multi-Objective Optimizer . . . 21

2.3.4 Neural Network Surrogate Model . . . 21

3 Blade Parametrization and CFD Solver 23

3.1 Introduction . . . 23

3.2 NURBS: Non-Uniform Rational B-Splines . . . 24

3.3 `ParaBlade`: 2D Blade Construction . . . 25

3.3.1 Camberline Definition . . . 26

3.3.2 Thickness Distribution . . . 26

3.3.3 Leading and Trailing Edge Curvature . . . 27

3.3.4 Blade Profile Assembly . . . 28

3.4 `MatchBlade`: Reverse Engineering Workflow . . . 29

3.4.1 Optimization Formulation . . . 29

3.4.2 Initial Guess Strategy . . . 29
3.4.3 Convergence and Accuracy . . . 30
3.4.4 Design Parameters . . . 31
3.5 MISES CFD Solver . . . 33
3.5.1 Introduction and Computational Approach . . . 33
3.5.2 Governing Equations and Numerical Method . . . 33
3.5.3 Input and Output File Structure . . . 36
3.6 Transonic Validation and Computational Setup Selection . . . 37
3.6.1 Mesh Type Sensitivity . . . 38
3.6.2 Grid Density Sensitivity . . . 40
3.7 Artificial Dissipation and MUCON Selection . . . 46
3.7.1 Independent Validation via ADS CFD Solver . . . 49

4 Design of Experiments 55
4.1 Reference Blade Database . . . 56
4.1.1 Real Blade Geometries . . . 56
4.1.2 Synthetic Blade Generation . . . 57
4.2 Parameter Correlation Analysis . . . 59
4.2.1 Correlation Theory . . . 59
4.3 Choice of Sampling Distribution . . . 62
4.3.1 Multivariate Gaussian Conditional Sampling . . . 63
4.3.2 Assumptions of the Gaussian Model . . . 65
4.4 The Iman-Conover Method . . . 67
4.5 Pitch-to-Axial-Chord Ratio Determination . . . 70
4.5.1 Limitations of Analytical Loading Criteria . . . 70
4.5.2 Empirical Pitch Selection Model . . . 71
4.6 Blade Generation Algorithm . . . 74
4.6.1 Parameter Partitioning and Boundary Conditions . . . 74
4.6.2 Pipeline Phases . . . 75
4.6.3 DoE Results . . . 78

5 The Optimizer 81
5.1 Preliminary Single-Objective Test . . . 81
5.1.1 Objective Function and Angle Constraint . . . 82
5.1.2 Optimizer Setup . . . 83
5.1.3 Results . . . 84
5.2 NSGA-II . . . 88
5.2.1 Genetic Algorithms . . . 88
5.2.2 The Pareto Front . . . 88

5.2.3 Non-Dominated Sorting . . . 89
5.2.4 Crowding Distance . . . 89
5.2.5 Selection, Elitism, and the Generation Cycle . . . 89
5.3 Sensitivity Study . . . 90
5.3.1 Choice of the Test Case . . . 90
5.3.2 Choice of Crossover Operator . . . 91
5.3.3 Choice of the Loading Objective . . . 95
5.3.4 Choice of the Angle Constraint Strategy . . . 100
5.3.5 Hard Constraint on Loss . . . 102
5.3.6 Hyperparameter Sensitivity Study . . . 103
5.3.7 Total CFD Budget . . . 107

6 The Surrogate Model 111
6.1 From Optimisation Results to a Training Dataset . . . 111
6.1.1 Parametrisation of the Pareto Front via Arc-Length . . . 112
6.2 Dataset Construction Pipeline . . . 113
6.2.1 Dataset composition . . . 114
6.3 Problem Formulation . . . 117
6.4 Network Architecture . . . 117
6.5 Hyperparameter Optimization . . . 118
6.5.1 Search Space . . . 119
6.5.2 Optimal Configuration . . . 120
6.6 Training Results . . . 120
6.6.1 Prediction Accuracy . . . 121

7 Results and Discussion 125
7.1 Inference Pipeline . . . 125
7.2 Results on Corner Points . . . 127
7.2.1 Surrogate Predictions in the Objective Space . . . 127
7.2.2 Effect of M_2 and p on Blade Geometry and Loading . . . 130
7.3 Outlet Flow Angle Accuracy . . . 134
7.4 Conclusions and Future Work . . . 137
7.4.1 Summary of Contributions . . . 137
7.4.2 Limitations . . . 138
7.4.3 Future Work . . . 139

Bibliography 141

Chapter 1

Literature Survey

1.1 Literature Review

The aerodynamic design of turbine blades is a complex **multi-objective optimization problem** characterized by high-dimensional geometric spaces and strongly non-linear fluid dynamics. High-fidelity CFD simulations are computationally expensive, making exhaustive design space exploration impractical. This has motivated the adoption of **data-driven methodologies**, which use **surrogate models** to replace costly simulations with fast approximations. This section reviews the state-of-the-art in surrogate modeling, inverse design, and statistical sampling techniques relevant to turbomachinery design.

1.1.1 Machine Learning and Surrogate Modeling in Aerodynamics

In aerodynamic shape optimization, the dominant bottleneck is the cost of repeated CFD evaluations in a high-dimensional design space. As the number of geometric degrees of freedom increases (typically $\mathcal{O}(10\text{–}20)$ for 2D turbine profiles), both gradient-based and evolutionary workflows require a rapidly growing number of function evaluations, which becomes prohibitive when multiple operating conditions must be considered [1].

Surrogate models alleviate this cost by replacing the CFD solver with a regression model trained on a limited set of high-fidelity samples. Common choices in aerodynamics include **Gaussian Process Regression (Kriging)** and **neural networks** [2]. However, the expected accuracy is often driven more by the *training data* than by the specific regression architecture: a poorly constructed **design of experiments** yields unreliable predictions even with sophisticated models [2].

This aspect is particularly critical in **transonic cascades**, where small geometric

perturbations may trigger abrupt regime changes due to shock formation and **shock-boundary-layer interaction** (SBLI). Supervised learning models are intrinsically interpolative and may fail when extrapolating beyond the domain covered by the training set [3]. Consequently, the present work prioritizes **feasibility-aware sampling strategies** that respect physical parameter dependencies, alongside **targeted dataset enrichment** in the low-loss region, where the surrogate must be most accurate for design purposes.

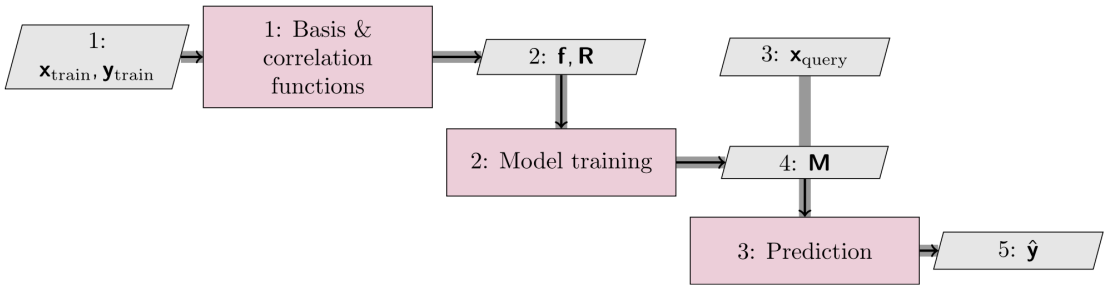


Figure 1.1: General framework for Kriging model (Li, Du, and Martins [1])

1.1.2 Forward and Inverse Design Strategies

Data-driven design frameworks in turbomachinery can be divided into two categories: **forward** and **inverse** approaches. In forward design, an optimizer iteratively modifies the blade geometry to minimize a loss function, evaluating each candidate through CFD. This is the classical approach used in most optimization workflows. Inverse design, by contrast, attempts to directly predict the optimal geometry given a set of aerodynamic requirements—commonly referred to as the **duty specification**. Inverse methods have attracted growing interest because they offer the possibility of near-instantaneous design generation once a surrogate model is trained.

Zhou et al. [4] developed a bi-directional design system that includes both a forward model (using Back-Propagation Neural Networks to predict performance from geometry) and an inverse model (using Radial Basis Function Networks to reconstruct geometry from performance targets). Their work highlights a fundamental difficulty with inverse design: the problem is often **ill-posed**. Multiple geometries can produce similar overall performance metrics, creating ambiguity for the learning algorithm. The model may struggle to distinguish between these solutions, leading to inconsistent predictions.

One strategy for addressing this ambiguity is to incorporate **physical priors** into the learning process. Chen et al. [5] proposed a two-stage framework that decouples aerodynamic physics from geometric representation. Rather than mapping operating conditions directly to blade coordinates, their system first uses a Multi-Output Gaussian Process (MOGP) to predict an optimal blade **loading distribution** (the pressure

curve along the blade surface). A Deep Neural Network then maps this target loading to the actual geometric parameters. By introducing the loading distribution as an intermediate constraint, this approach regularizes the inverse problem and ensures that the predicted geometries correspond to physically realizable flow fields.

A similar approach was demonstrated by Porta [6], who used Gaussian Process Regression (GPR) for inverse turbine blade design. His model was trained on approximately 3000 blade profiles to learn a mapping from both the aerodynamic duty and the desired loading style to the geometric parameters. The duty vector specifies the operating conditions (inlet angle, outlet angle, Mach number, and Reynolds number), while the style defines the target Mach number distribution along the blade surface through four control variables.

The present thesis extends this methodology to the **transonic regime**, where compressibility effects and shock waves impose stricter constraints on which geometries are physically feasible. Unlike Porta's work, which included the loading style as an input, the current approach focuses on learning the mapping from duty alone to near-optimal blade geometries.

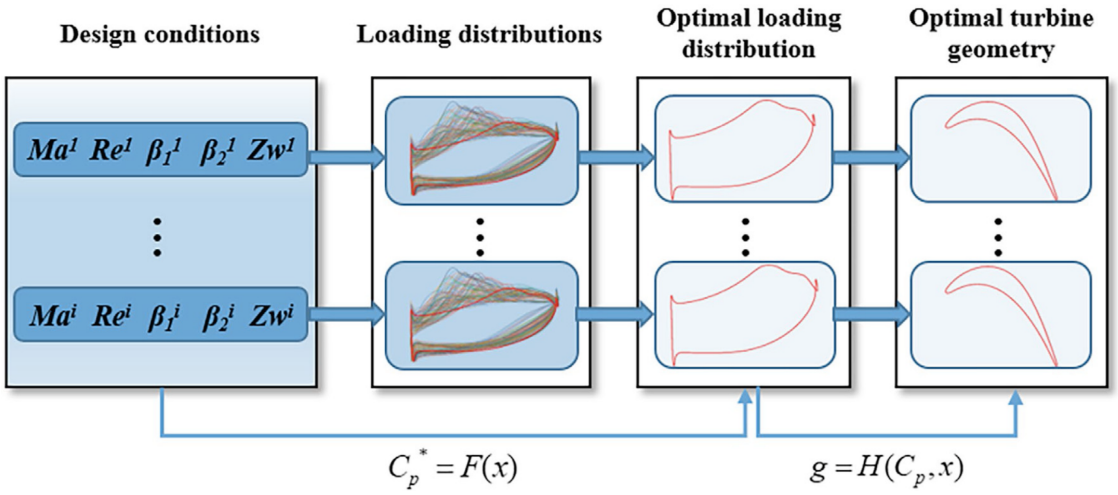


Figure 1.2: Overall framework for two-stage modeling (Chen et al. [5])

1.1.3 Challenges in Transonic Turbine Aerodynamics

The transonic regime introduces significant physical complexity that distinguishes it from subsonic design problems. In high-pressure turbine stages, the flow accelerates from subsonic to supersonic speeds, leading to shock wave formation and complex **shock-boundary layer interactions** (SBLI). These phenomena impose strict physical boundaries on the feasible design space and can cause CFD simulations to fail or produce non-physical results if the blade geometry is not carefully chosen.

One key constraint is the **limit load capacity** of the cascade. Fruchtman [7] defined this as the maximum tangential force that a blade row can generate before the flow **chokes**. For a given set of inlet conditions, there exists an upper bound on the turning that can be achieved. In an inverse design context, this means that certain duty specifications may be physically impossible to satisfy—no geometry exists that can produce the requested flow angles and Mach number simultaneously. A data-driven model must implicitly learn this non-linear boundary to avoid generating infeasible designs.

Compressibility effects also alter the loss mechanisms compared to low-speed flows. Ni et al. [8] showed that standard loading criteria, such as the **Zweifel coefficient**, must be adjusted to account for the outlet Mach number (M_2). As M_2 approaches unity, the optimal loading distribution changes, and designers must balance profile losses against shock losses. Simply minimizing total pressure loss is insufficient—the position and strength of the shock wave can have a strong impact on overall efficiency.

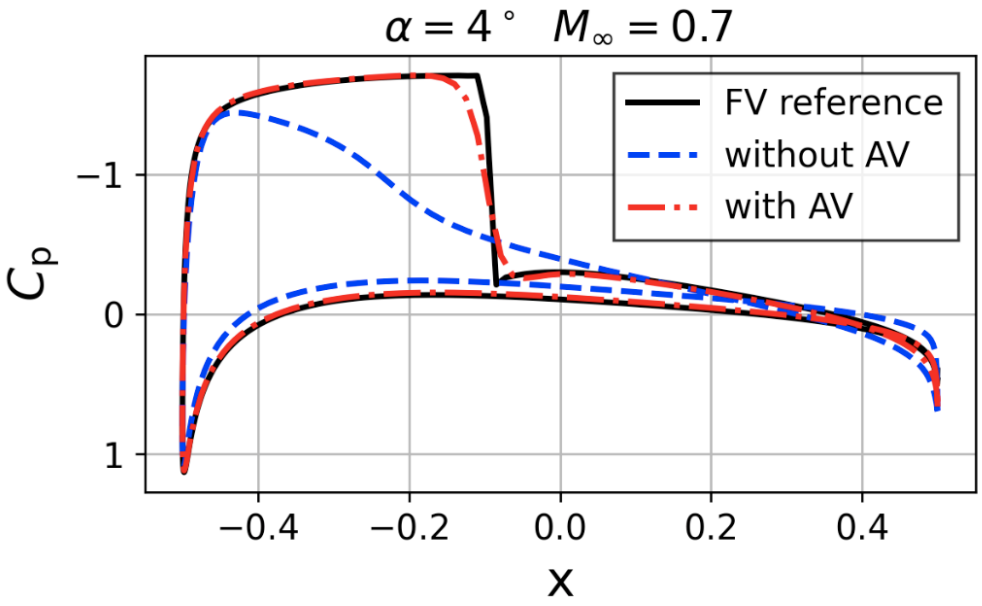


Figure 1.3: Application of artificial viscosity in Physics-Informed Neural Networks (PINNs) for transonic flows (Wassing et al. [9])

From a modeling perspective, capturing shock discontinuities is challenging. Wassing et al. [9] investigated **Physics-Informed Neural Networks (PINNs)** for transonic flows and found that standard neural network architectures struggle to resolve the sharp gradients associated with normal shocks. This suggests that a uniform sampling of the design space may not provide sufficient training data in the regions where the surrogate model needs to be most accurate—namely, the low-loss transonic designs near

the optimum.

To address this, the present work adopts a two-stage data generation strategy. A static design of experiments is utilized solely to provide a physically sound geometric initialization for the design space exploration. Rather than training the surrogate model on this broad initial sampling, multi-objective optimization is deployed to systematically map the high-performance boundary of the design space. The final training set is therefore constructed exclusively from Pareto-optimal and near-Pareto blades that have been explicitly optimized to minimize aerodynamic loss and maximize blade loading. This approach ensures that the surrogate achieves maximum accuracy precisely in the region of engineering interest, mitigating the difficulty of regressing strongly non-linear phenomena, such as shock waves, across the entire unconstrained design space.

1.1.4 Geometric Parameterization and Sampling Strategy

The representation of blade geometry plays a central role in enabling data-driven design. The choice of parameterization determines both the flexibility of the design and the structure of the training dataset. A poor parameterization can lead to geometries that are difficult to manufacture or that produce unphysical flow fields. Agromayor et al. [10] provide a comprehensive framework for turbomachinery blade parameterization, advocating for constructive approaches based on **NURBS** or B-splines rather than discrete point definitions. These methods ensure smooth curvature continuity (C^2), which is essential for avoiding spurious flow features in CFD simulations.

The present work adopts the `ParaBlade` framework, an open-source implementation of these principles. `ParaBlade` extends classical parameterization schemes (such as the 11-parameter model by Pritchard [11]) to a flexible system where the blade is defined by **21 variables** controlling metal angles, radii of curvature, and thickness distributions.

However, defining the geometry is only part of the challenge. To train a surrogate model, a **Design of Experiments (DoE)** must be constructed to populate the design space with CFD evaluations. The choice of sampling strategy has a direct impact on surrogate accuracy. In turbomachinery applications, the geometric parameters are not independent—they exhibit physical correlations that reflect manufacturing constraints and aerodynamic feasibility. For example, the stagger angle is typically related to the throat area, and the leading edge radius is influenced by the local thickness distribution.

Prots et al. [12] argue that standard sampling methods like **Latin Hypercube Sampling (LHS)**, which treat all parameters as independent, are inefficient for turbomachinery design. LHS explores the corners of the design hypercube, generating combinations of parameters that often correspond to geometrically invalid or aerodynamically nonsensical blades. This is particularly problematic in the transonic regime, where unrealistic geometries frequently cause CFD simulations to diverge or produce

non-physical shock patterns.

To address this issue, the sampling strategy must incorporate prior knowledge of parameter dependencies. Iman and Conover [13] developed a method for inducing a specified rank correlation structure on a set of samples while preserving their marginal distributions. This approach allows the construction of a multivariate sampling distribution that respects the physical correlations observed in real turbine blades. By measuring the **Spearman correlations** from a reference set of existing transonic blade designs, a covariance matrix Σ can be constructed and used to generate new samples that lie within the manifold of physically plausible geometries.

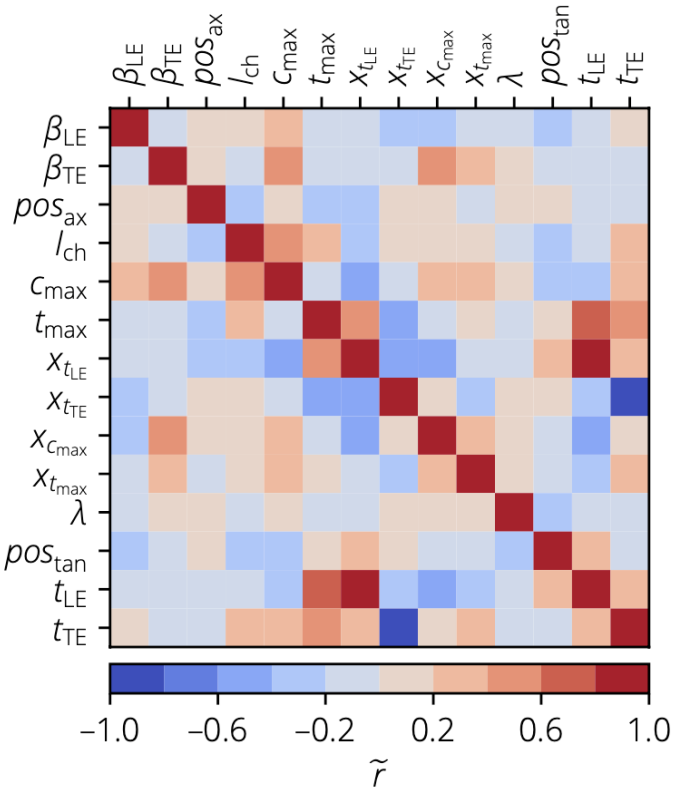


Figure 1.4: Correlation matrix of the profile parameters for a turbomachinery test case (Prots et al. [12])

1.1.5 Optimization of Transonic Turbine Cascades

The aerodynamic design of transonic turbine cascades presents unique optimization challenges due to the presence of shock waves, shock-boundary layer interactions, and discontinuities in the objective function landscape. These characteristics have motivated extensive research into gradient-free, gradient-based, and hybrid optimization strategies.

Gradient-Free Optimization Methods

Evolutionary algorithms have been applied to turbomachinery optimization due to their ability to perform global exploration without requiring derivative information. Shelton et al. [14] applied a **Genetic Algorithm (GA)** combined with Artificial Neural Networks and CFD to optimize a transonic turbine airfoil, demonstrating that evolutionary approaches could successfully navigate the non-convex design space characteristic of transonic flows. The study highlighted the ability of GA to escape local minima but noted the high computational cost due to the large number of function evaluations required.

Sonoda et al. [15] investigated advanced high-loaded transonic turbine airfoils using **multi-objective evolutionary algorithms**. Their work emphasized that evolutionary approaches were essential for exploring the **Pareto front** in multi-objective problems where competing design objectives (such as efficiency, turning angle, and structural constraints) must be balanced simultaneously. The study demonstrated successful optimization of transonic cascades but confirmed the computational expense inherent to population-based methods.

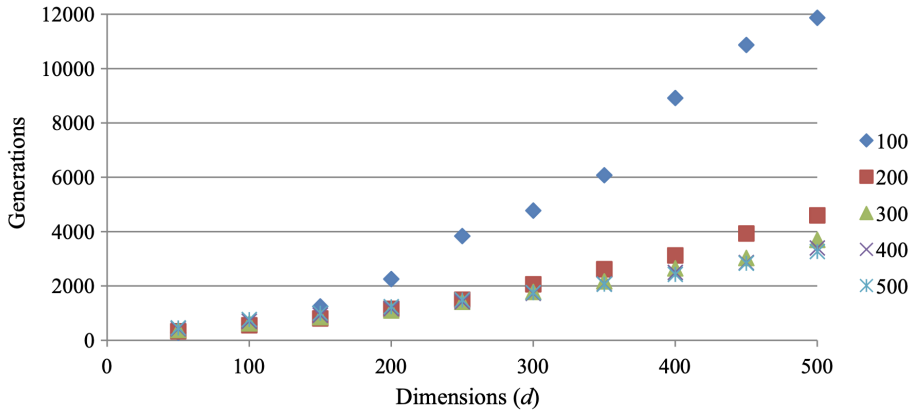


Figure 1.5: Generations required for DE to converge varying the number of population (Chen et al. [16])

Chen and Montgomery [16] provided a theoretical analysis of the **curse of dimensionality** and its effects on Particle Swarm Optimization and Differential Evolution. Their work quantified how the convergence performance of gradient-free methods degrades as the number of design variables increases, confirming that while these methods can identify promising regions of the design space, the final convergence toward the optimum becomes progressively slower in high-dimensional problems typical of turbine blade parametrization.

Gradient-Based Optimization with Adjoint Methods

The development of **discrete adjoint methods** for CFD has enabled efficient gradient computation at a cost independent of the number of design variables. Jameson [17] pioneered the application of control theory and adjoint methods to aerodynamic design, establishing the theoretical foundation for gradient-based shape optimization in compressible flows.

In the context of transonic turbines, gradient-based optimization using adjoint methods has been applied successfully to reduce profile losses. However, several studies have reported challenges related to the sensitivity of shock position to geometric perturbations. A previous optimization of the LS89 cascade using a classical gradient-based method achieved only an 11.6% reduction in entropy generation [18], significantly less than the 21.3% improvement obtained with the hybrid evolutionary approach in the same study. This difference was attributed to the gradient-based method converging to a local minimum rather than exploring the global design space.

	LS89	GBM [1] Variation	Hybrid Variation
$J\ [Pa/(kg/m^3)]$	826.7	731 -11.6 %	650.5 -21.3%
$\alpha_{out}[°]$	74.89	74.89 -0 %	74.89 -0%
Number of evaluations		70	1600

Figure 1.6: Hybrid Algorithm vs Gradient Based Method (Châtel et al. [18])

Chapter 2

Problem Formulation

2.1 Transonic Turbine Design

Modern gas turbines operate at increasingly **high cycle pressure ratios** to maximize thermal efficiency. This trend, illustrated in Figure 2.1, has driven the flow through **high-pressure (HP) turbine stages** into the **transonic regime**, where locally supersonic flow and shock waves are unavoidable.

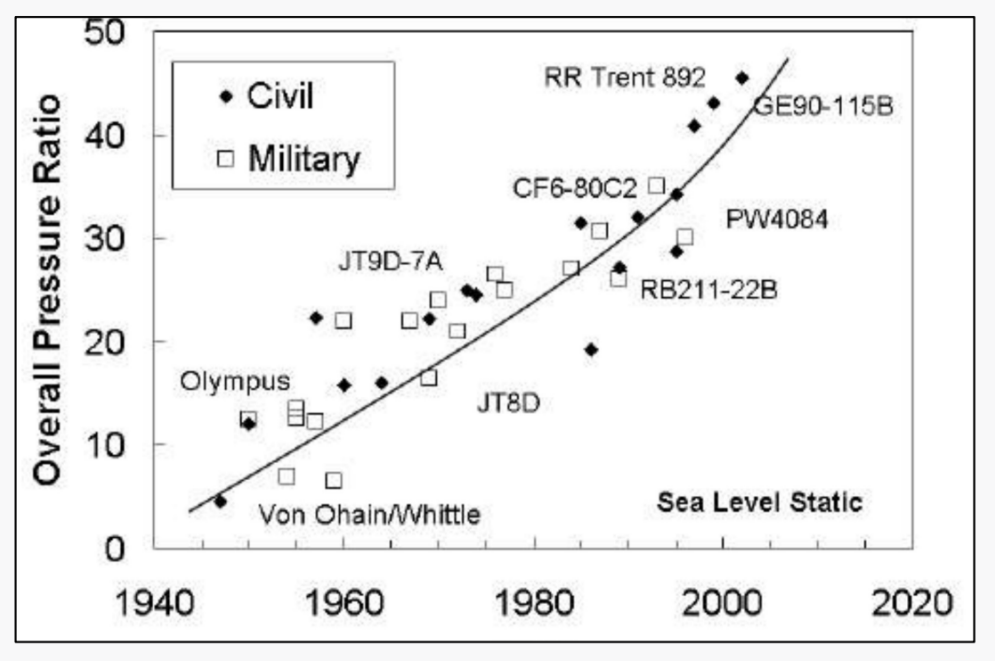


Figure 2.1: Evolution of overall pressure ratio (OPR) in gas turbine engines over the past decades for civil and military applications - *public source*

The transonic regime introduces significant **design complexity**. The flow accelerates from subsonic conditions at the inlet to locally supersonic speeds on the blade suction side, typically terminating in a **shock wave** that returns the flow to subsonic before

cascade exit. **Shock-boundary layer interactions** can trigger flow separation, and aerodynamic losses are highly sensitive to shock position and strength. Small geometric changes can therefore have disproportionate effects on performance.

The challenge is not whether to design transonic blades — modern thermodynamic cycles demand such high work extraction that transonic operation is inevitable — but rather how to design them **efficiently**. Traditional iterative workflows, where geometries are proposed, evaluated through CFD, and manually refined, are time-consuming and require significant expertise. **Data-driven methods** offer an alternative: by training surrogate models on databases of CFD simulations, near-optimal geometries can be predicted directly for specified operating conditions, bypassing the iterative refinement loop.

2.1.1 Physical Challenges in the Transonic Regime

The flow through a turbine cascade is governed by the **compressible Navier–Stokes equations**, expressing conservation of mass, momentum, and energy. For a Newtonian fluid with constant transport properties, these equations can be written in conservative form as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{W}) = 0 \tag{2.1}$$

$$\frac{\partial (\rho \vec{W})}{\partial t} + \nabla \cdot (\rho \vec{W} \otimes \vec{W}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} \tag{2.2}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot [(\rho E + p)\vec{W}] = \nabla \cdot (\boldsymbol{\tau} \cdot \vec{W}) + \nabla \cdot (k \nabla T) \tag{2.3}$$

where:

- ρ is the fluid density
- $\vec{W}$ is the velocity vector; $\vec{W} = (u, v, w)$
- p is the static pressure
- E is the total energy per unit mass
- $\boldsymbol{\tau}$ is the viscous stress tensor
- k is the thermal conductivity
- T is the temperature

In turbomachinery applications, these equations are typically solved using **Reynolds-Averaged Navier–Stokes (RANS)** methods coupled with turbulence closure models. However, for the CFD simulations performed in this work, a different approach was

adopted. The `MISES` code, developed at MIT specifically for turbomachinery analysis, solves the **Euler equations** on the blade-to-blade stream surface and couples them with **integral boundary layer equations** through a transpiration boundary condition. This hybrid approach models viscous effects via body forces in the Euler solver while maintaining computational efficiency suitable for design space exploration. A detailed description of `MISES` and its validation for transonic flows is provided in Chapter 3.

Mixed Elliptic-Hyperbolic Character

Transonic flow presents a fundamental mathematical challenge: the governing equations change character across the sonic line. In subsonic regions where $M < 1$, the equations are **elliptic**. Disturbances propagate instantaneously throughout the domain, and the solution is globally coupled. A geometric perturbation at one location affects the pressure field both upstream and downstream.

When the flow becomes locally supersonic with $M > 1$, the equations become **hyperbolic**. Information now propagates along **characteristic lines** (Mach Line) inclined at the **Mach angle** $\mu = \arcsin(1/M)$ relative to the local flow direction. Causality is restored: a point in the supersonic region is influenced only by conditions within its upstream Mach cone. Downstream influence persists, but upstream influence is lost.

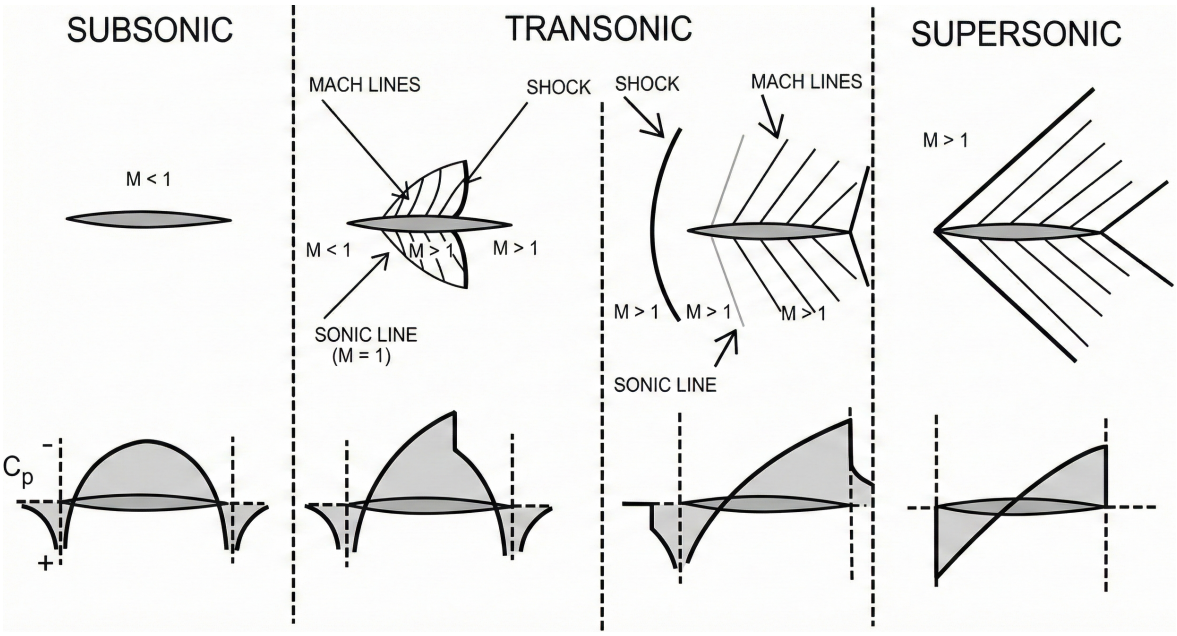


Figure 2.2: Distributions of Mach Lines in different regimes

Transonic turbine cascades exhibit a **mixed flow field** where subsonic and supersonic regions coexist, separated by a **sonic line** and terminated by shock waves. This has direct consequences for aerodynamic design. The surface Mach distribution in the

supersonic patch cannot be freely prescribed—it is constrained by the global pressure ratio and by the characteristic structure emanating from the sonic line. This mixed elliptic–hyperbolic behavior is why subsonic inverse-design methods, which prescribe target loading distributions, cannot be extended to fully transonic conditions without fundamental modifications.

2.1.2 Geometric Constraints on Shock Formation

The mathematical constraints described above are further compounded by geometric limitations inherent to the cascade architecture. The flow passage between adjacent turbine blades behaves as a **convergent-divergent nozzle**, with the effective flow area first decreasing from inlet to throat and then increasing toward exit. This divides the cascade into two aerodynamically distinct regions.

The **covered turning region** extends from the leading edge to approximately the trailing edge of the adjacent blade's pressure surface. Here the flow is confined between two solid surfaces on both sides, forming a convergent channel. The **uncovered turning region** begins downstream of this point, where the flow can expand freely with only one bounding surface, corresponding to the divergent section of the nozzle.

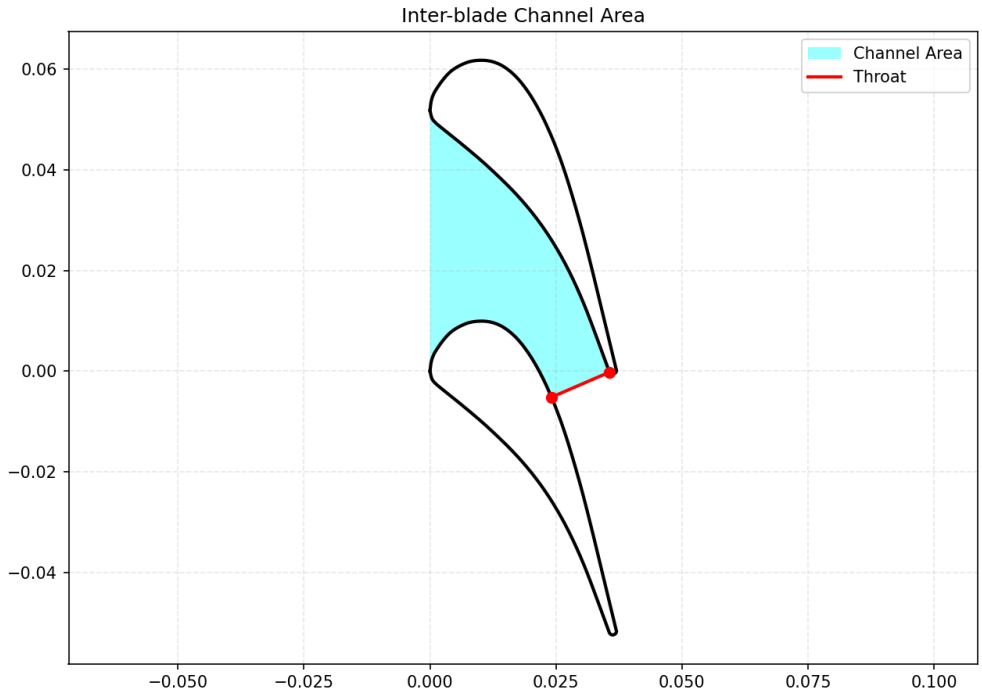


Figure 2.3: Flow passage in the VKI LS89 test case

In typical transonic turbine cascades, the **throat**—the location of minimum flow

area—coincides with the trailing edge of the adjacent blade. Since the flow enters at subsonic conditions and accelerates through the convergent covered region, it can reach sonic conditions ($M = 1$) at the throat when the cascade operates near its **limit load**. Any supersonic flow and shock waves can therefore only occur downstream of the throat, in the uncovered turning region where the divergent area distribution permits supersonic expansion.

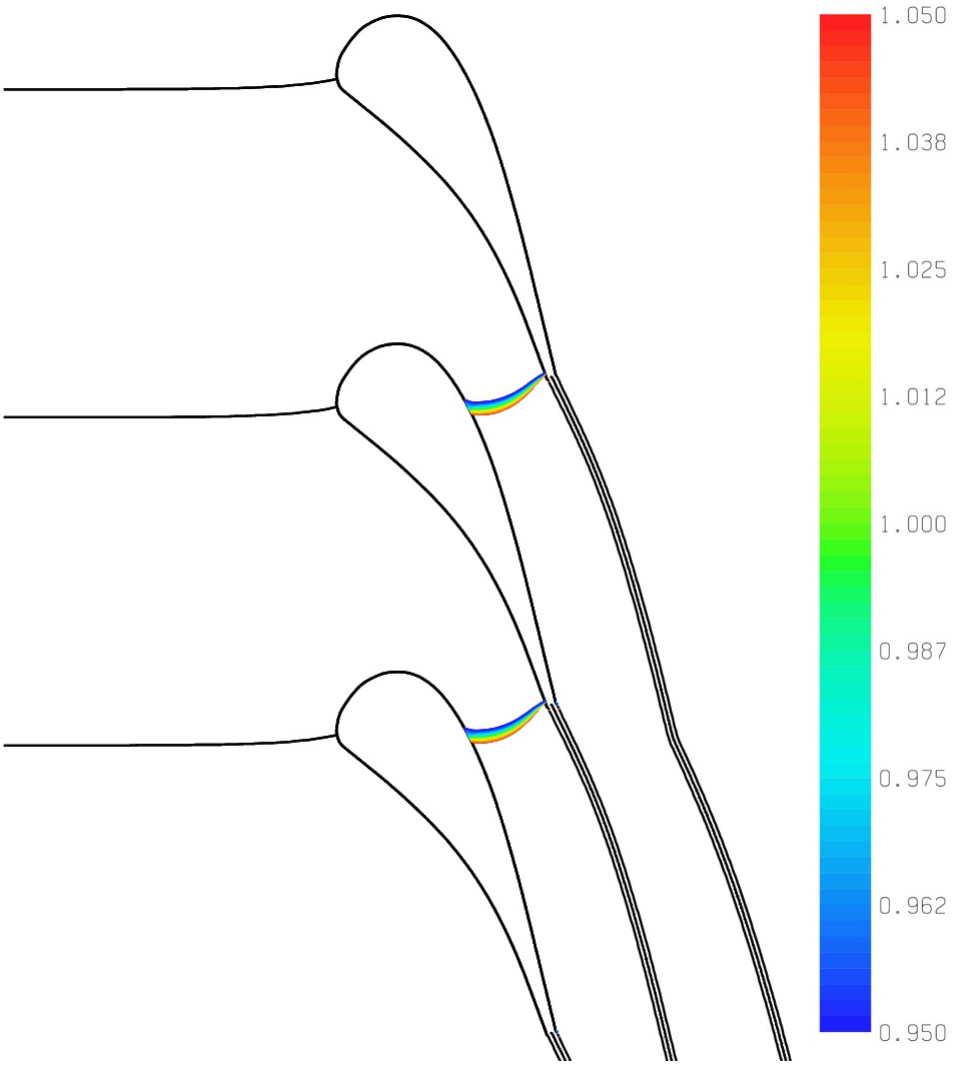


Figure 2.4: Mach contour for throat identification via `MISES` on the LS89 test case ($M_2 = 1.2$)

This geometric constraint has profound implications for transonic blade design. The covered region, being convergent with a subsonic inlet, is inherently protected from shock formation regardless of blade geometry. Shock waves, when present, must appear in the uncovered region on the blade suction side. The designer's geometric degrees of freedom are therefore most critical in shaping the **rear portion of the suction surface**, where small changes in curvature or thickness can significantly alter

shock position, **strength**, and the resulting losses. This explains why **trailing edge geometry** and **aft-loading distribution** are particularly sensitive parameters in the transonic regime, and why traditional design approaches based on experience with subsonic cascades often fail when extended to transonic operating conditions.

2.2 Design Objective and Duty Space

The objective of this work is to construct a surrogate model that maps prescribed operating conditions to blade geometries exhibiting near-optimal aerodynamic performance. This mapping is defined as:

$$\text{duty} = (\alpha_1,\ \alpha_2,\ M_2,\ Re_2) \quad \longrightarrow \quad \mathbf{X}_{\text{opt}} \tag{2.4}$$

where $\mathbf{X}_{\text{opt}}$ represents the geometric design parameters that yield favorable performance for the specified duty. The **trailing-edge thickness** is held constant at $t_{\text{TE}}/c_{\text{ax}} = 3\%$ throughout the entire campaign and is therefore excluded from the mapping. Performance is assessed through both **aerodynamic efficiency** (kinetic energy loss coefficient) and **aerodynamic loading** (blade pitch), with the exact optimization formulation detailed in Chapter 5.

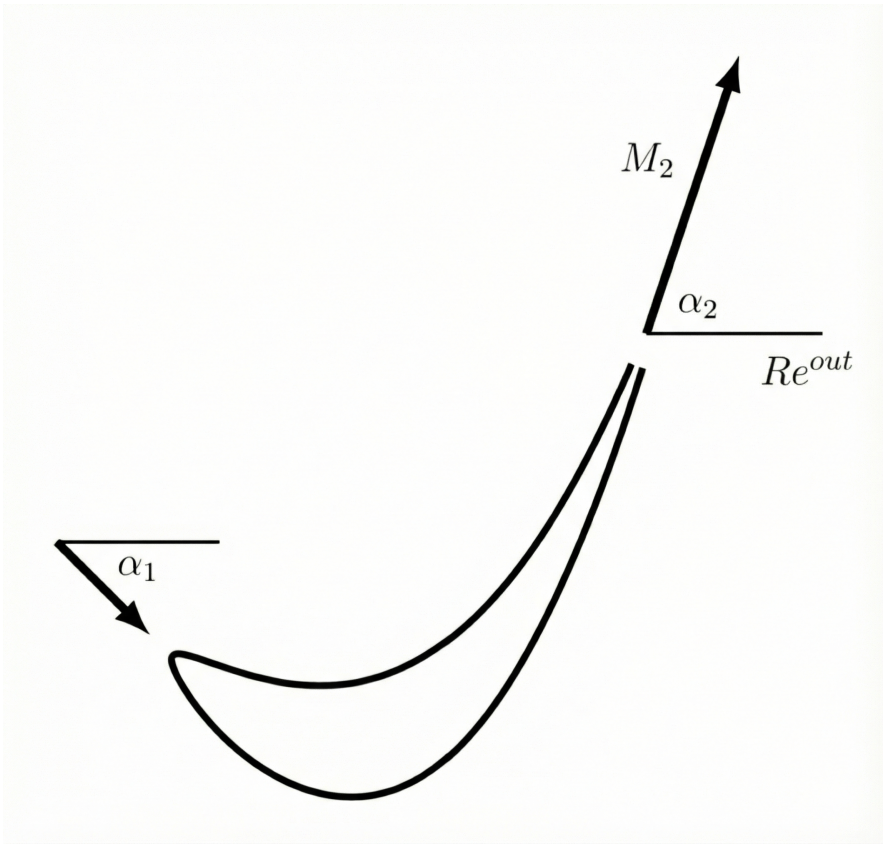


Figure 2.5: Aerodynamic duty scheme (Porta [6])

Duty Vector Definition

The **duty vector** completely specifies the operating conditions of the cascade:

- **Inlet flow angle** α_1 and **outlet flow angle** α_2, measured from the axial direction — their difference defines the blade turning and work extraction.
- **Outlet Mach number** M_2, evaluated at the mixed-out state downstream of the cascade.
- **Reynolds number** Re_2, based on axial chord and outlet conditions.

The **turning angle** $\alpha_1 - \alpha_2$ is directly related to stage work extraction through the Euler turbine equation. Higher turning generally corresponds to higher loading but increases susceptibility to losses, particularly in the transonic regime where shock-boundary layer interactions become significant.

Design Space Coverage

The **design space** explored in this work spans the operating conditions representative of high-pressure turbine stages in modern aero-engines and industrial gas turbines. The **Reynolds number** is fixed at $Re_2 = 10^6$ and the **trailing-edge thickness** at $t_{\mathrm{TE}}/c_{\mathrm{ax}} = 3\%$ throughout the entire campaign, reducing the free parameters to the three aerodynamic quantities:

- **Inlet flow angle**: $\alpha_1 \in [0°, 50°]$
- **Outlet flow angle**: $\alpha_2 \in [-50°, -75°]$
- **Isentropic exit Mach number**: $M_2 \in [0.8, 1.2]$

The resulting **turning angle** $\alpha_1 - \alpha_2$ spans from 50° to 125°, covering three representative regimes:

- **Low turning** (50°–75°): mild flow acceleration, shocks appear only at the highest outlet Mach numbers.
- **Medium turning** (75°–100°): stronger adverse pressure gradients on the suction side, increasing sensitivity to shock position.
- **High turning** (100°–125°): substantial flow diffusion, earlier shock formation, and higher susceptibility to flow separation and boundary layer interaction.

Duty Grid

The continuous design space is discretized into a structured **grid of 125 duty points**, obtained by combining 25 angle pairs (α_1, α_2) with 5 Mach numbers. The angle grid is defined on a 5×5 matrix:

- $\alpha_1 \in \{0°, 15°, 30°, 40°, 50°\}$
- $\alpha_2 \in \{-50°, -60°, -65°, -70°, -75°\}$

The four **corner points** — extreme combinations $(0°, -50°)$, $(50°, -50°)$, $(0°, -75°)$, $(50°, -75°)$ — are labeled **C1**, **C2**, **C3**, **C4**. The remaining 21 interior points are labeled **A1**–**A21**. Combined with five Mach numbers $M_2 \in \{0.8, 0.9, 1.0, 1.1, 1.2\}$, this yields **125 duty points** in total, illustrated in Figure 2.6.

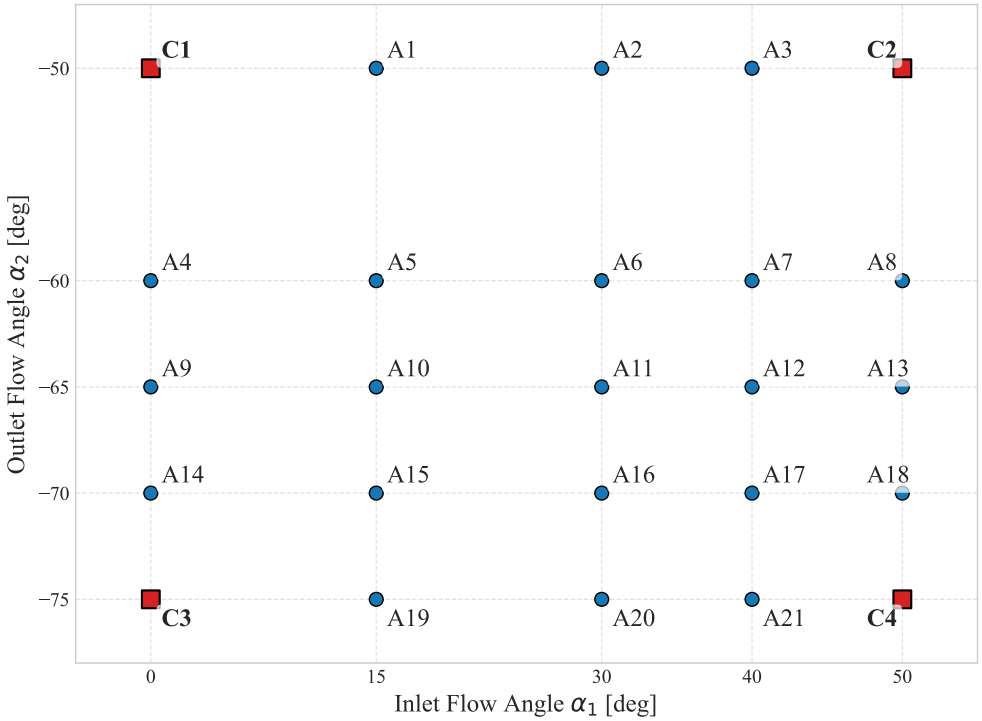


Figure 2.6: Discretized duty grid in the (α_1, α_2) plane. Interior points A1–A21 are shown in blue; corner points C1–C4 are highlighted in red.

2.2.1 Extension from Porta's Inverse Design Framework

The present work builds directly upon the methodology developed by Porta [6], who demonstrated the feasibility of **inverse turbine blade design** using **Gaussian Process Regression (GPR)**. His framework was trained on a comprehensive database of approximately 3000 blade profiles, each evaluated using the `MISES` viscous cascade solver, to learn a direct mapping from prescribed aerodynamic requirements to blade

geometry. The fundamental innovation was a **dual-input structure** that captured both operating conditions and the desired aerodynamic character of the blade.

Porta's Dual-Input Architecture

Porta's approach treated the design problem as a mapping from two distinct categories of inputs:

- **Aerodynamic duty** $(\alpha_1, \alpha_2, M_2, Re_2)$: the set of operating conditions the blade must satisfy, as defined in Section 2.
- **Aerodynamic style**: four control parameters prescribing the target Mach number distribution along the blade surface. The style defined the blade loading—the pressure difference between suction and pressure sides—and controlled where along the chord the peak Mach number occurred and how rapidly the flow accelerated to reach it.

By varying the style while holding the duty fixed, the designer could generate blades with identical overall flow deflection but distinctly different loading distributions. The trained GPR surrogate could instantaneously produce geometries for any combination of duty and style within the training domain, effectively acting as an inverse design tool.

Demonstrated Performance in the Subsonic Regime

Porta's database covered a range of subsonic to mildly compressible operating conditions, with outlet Mach numbers extending up to approximately $M_2 = 0.7$ and peak suction-side Mach numbers reaching $M_{\text{peak}} \approx 0.98$, just below the threshold where shock formation becomes dominant.

The trained GPR model achieved remarkable accuracy in this regime:

- **Outlet angle prediction**: Average error $\Delta\alpha_2 = 0.078°$, with RMS error of $0.009°$.
- **Mach distribution matching**: RMS error between target and predicted surface Mach distribution below 1%.
- **Loss prediction**: Total pressure loss coefficient estimated within $\sim 2\%$ deviation.
- **Generation time**: Blade geometry produced in ~ 0.01 seconds.

The success of this methodology convincingly demonstrated that inverse design via surrogate modeling represents a viable and practical alternative to traditional forward optimization approaches, at least for subsonic and mildly compressible flows.

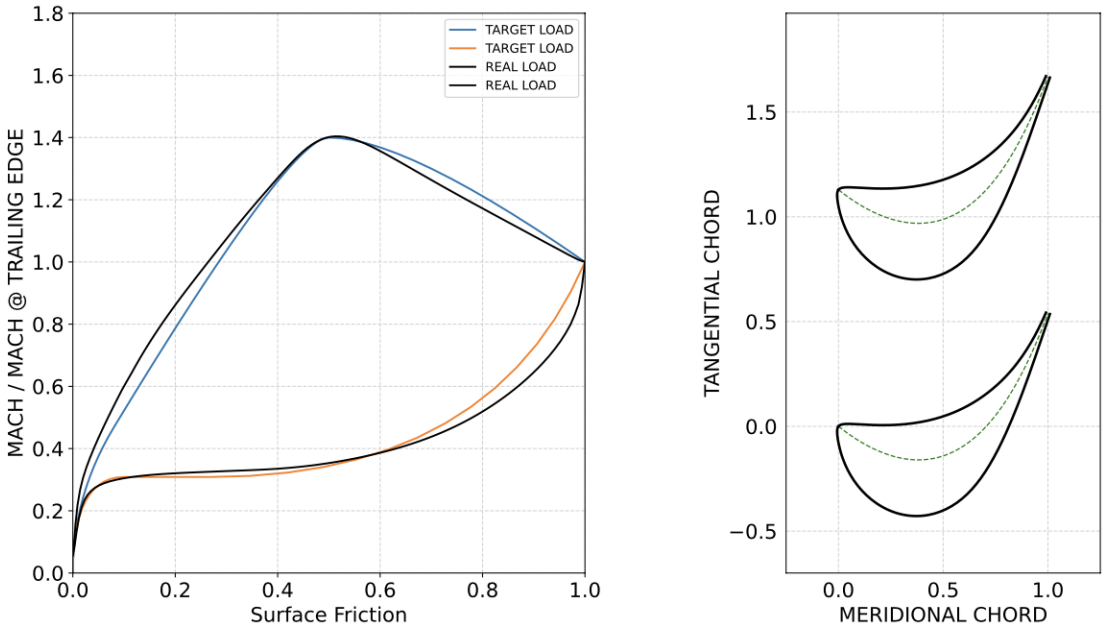


Figure 2.7: Blade geometry and Mach distribution from Porta's framework at $M_2 = 0.7$. The normalized peak gives $M_{\text{peak}}^{\text{abs}} = 1.4 \times 0.7 = 0.98$ (Porta [6])

Physical Limitations in the Transonic Regime

However, extending this framework to **fully transonic conditions**—where locally supersonic flow regions, shock waves, and shock-boundary layer interactions become the dominant phenomena—reveals a **fundamental physical constraint**. In the transonic regime, the designer's ability to freely prescribe a target loading distribution is severely limited by the **mixed elliptic-hyperbolic** nature of the governing equations discussed in Section 2.

Figure 2.8 illustrates this constraint by examining a family of blade geometries originally generated by Porta's framework. Each blade geometry is held fixed while the outlet Mach number M_2 is varied across a range spanning subsonic to transonic conditions. As M_2 increases beyond approximately 0.8, the axial position of the **peak Mach number** on the suction side exhibits a **systematic drift downstream**.

This drift is observed consistently across the entire blade family, demonstrating that it is not a consequence of any particular blade shape but rather a **universal feature of transonic flow physics**. The peak Mach location is controlled by the characteristic structure of the transonic field—specifically, by the hyperbolic regions downstream of the sonic line that dictate where the supersonic patch can form and where the flow must decelerate through a shock. The designer's geometric degrees of freedom cannot override these physical constraints. A style prescription that might be achievable at $M_2 = 0.7$ becomes physically infeasible at $M_2 = 1.0$, regardless of how the blade geometry is

modified.

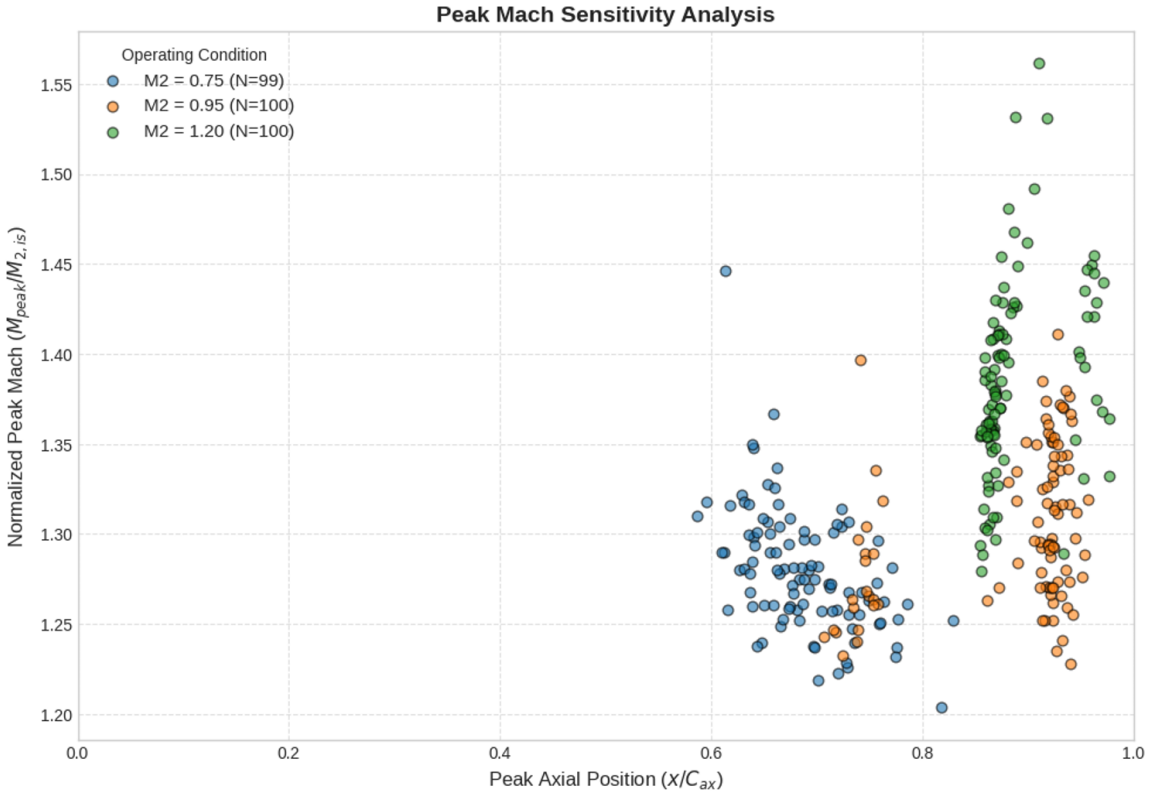


Figure 2.8: Axial position of peak Mach number versus outlet Mach number for a family of blade geometries

This observation has direct implications for the design strategy. In subsonic flow, the style vector provides meaningful control over the blade's aerodynamic behavior. In transonic flow, attempting to impose a prescribed loading distribution conflicts with the physical constraints imposed by the shock structure and characteristic lines. The **inverse problem becomes ill-posed**: for a given transonic duty, there may be no blade geometry capable of simultaneously achieving the prescribed exit conditions *and* the target style.

Departure from Style-Based Design

The fundamental constraint revealed by Figure 2.8 motivates the departure from Porta's dual-input architecture. While the duty-plus-style framework proved highly effective in the subsonic regime, extending it to transonic conditions encounters a fundamental physical barrier.

The mixed elliptic-hyperbolic character of the transonic field, combined with the geometric constraints on shock formation discussed in Section 2.2, severely limits the designer's ability to prescribe a target loading distribution independently of the

operating conditions. For a given transonic duty, the blade loading is largely dictated by the physics: the location and strength of the shock, the extent of the supersonic region, and the pressure recovery downstream are all constrained by the characteristic structure of the flow field.

Attempting to impose a prescribed style conflicts with these constraints, rendering the inverse problem ill-posed. In many cases, no blade geometry exists that can simultaneously achieve the prescribed exit conditions *and* the target loading distribution.

The present work therefore focuses exclusively on the **duty-to-geometry mapping**. The loading distribution is no longer prescribed as an input—it emerges as a consequence of the **physics-constrained optimization** process. This approach acknowledges the reality of transonic design: the designer specifies what the blade must achieve (the operating conditions), and the physics largely determines how it achieves it (the resulting loading characteristics).

The development of a transonic inverse design framework capable of reliably prescribing loading style while respecting the physical constraints imposed by shock formation remains an **open and challenging problem** for future research.

2.3 Methodology Overview

To address the transonic blade design challenge outlined in the preceding sections, this work develops a data-driven framework integrating four principal computational tools. Each tool addresses a specific aspect of the problem: ensuring physical consistency of the design space, efficient aerodynamic evaluation, exploration of performance trade-offs, and rapid prediction of near-optimal geometries. Figure 2.9 illustrates the complete workflow and the interaction between these components.

2.3.1 Multivariate Gaussian Sampling

The exploration of the design space relies on a sampling strategy based on **multivariate Gaussian distributions** fitted to a reference family of transonic turbine blades. Rather than employing traditional space-filling methods such as **Latin Hypercube Sampling**—which do not preserve the physical correlations between geometric parameters—this approach constructs a statistical model that captures the relationships between camber, thickness, pitch, and other parameters observed in physically realizable blades. Samples drawn from this distribution naturally respect the **parameter dependencies**, reducing the occurrence of non-convergent CFD simulations and unphysical flow features. Chapter 4 provides a detailed description of the sampling methodology and its validation.

2.3.2 MISES CFD Solver

Aerodynamic performance is evaluated using `MISES`, a specialized two-dimensional cascade solver capable of capturing transonic flow physics including shock waves, shock-boundary layer interaction, and viscous losses. `MISES` achieves computational times of approximately **2–3 seconds** per simulation through a **viscous-inviscid interaction** formulation, making it suitable for the large-scale database generation required by the data-driven approach. The solver's capabilities and validation for transonic flows are discussed in Chapter 3.

2.3.3 NSGA-II Multi-Objective Optimizer

The optimization strategy employs the **Non-dominated Sorting Genetic Algorithm II (NSGA-II)** to explore the trade-off between aerodynamic efficiency and blade loading. Rather than collapsing multiple objectives into a single scalar fitness function, the **multi-objective formulation** explicitly treats **loss minimization** and **loading maximization** as competing goals, generating a **Pareto front** of non-dominated solutions. This allows the designer to select the most appropriate compromise for a given application. A **smart selection filter** is applied to the optimization results to extract a diverse set of high-performance blades for surrogate model training. Chapter 5 describes the optimization formulation, algorithm configuration, and selection strategy.

2.3.4 Neural Network Surrogate Model

The final component is a **neural network surrogate model** that predicts blade geometries exhibiting near-optimal performance for prescribed operating conditions. Unlike Porta's Gaussian Process approach, which was suited to the smaller subsonic database ($\sim$3000 blades), the present work employs a **Multi-Layer Perceptron (MLP)** architecture capable of scaling to the larger datasets generated by the multi-objective optimization. The network is trained exclusively on high-quality designs extracted from the Pareto fronts, ensuring that predictions lie in regions of the design space where aerodynamic performance is favorable. Once trained, the surrogate enables **blade generation in milliseconds** without CFD evaluation. Chapter 6 presents the model architecture, training strategy, and validation.

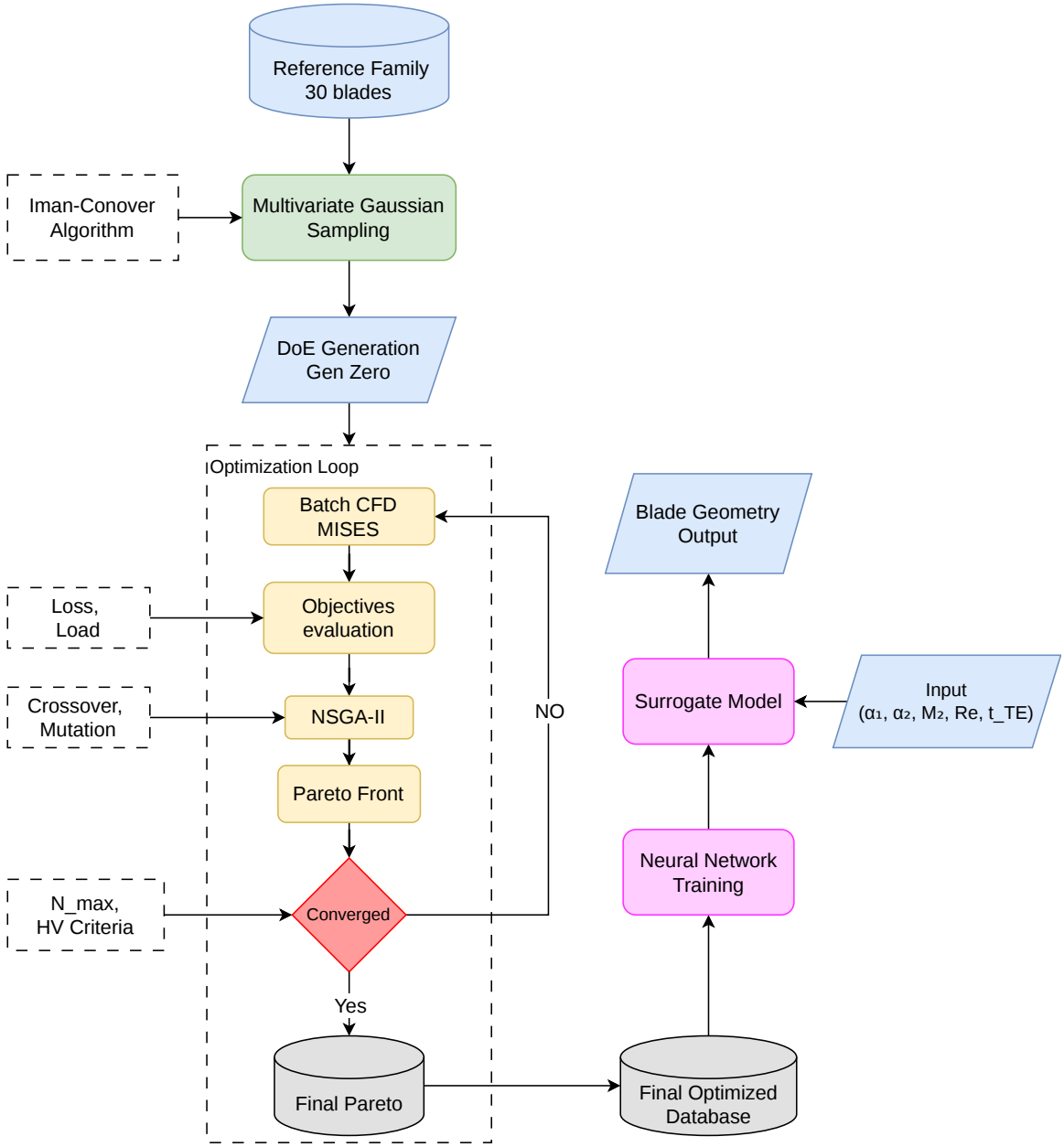


Figure 2.9: Complete workflow of the data-driven transonic blade design framework, showing the interaction between the four principal tools: multivariate Gaussian sampling for DoE generation, `MISES` for CFD evaluation, NSGA-II for multi-objective optimization, and the neural network surrogate for rapid design prediction.

Chapter 3

Blade Parametrization and CFD Solver

3.1 Introduction

The aerodynamic optimization and surrogate modeling framework developed in this work requires a parametric representation capable of accurately describing existing transonic turbine cascades. The parametrization must balance geometric fidelity with a manageable number of design variables, and critically, must support **reverse engineering**: extracting parametric descriptions from discrete point clouds of validated reference blades.

This chapter presents the `ParaBlade` framework [10], a constructive **NURBS-based parametrization** system. `ParaBlade` was selected primarily for its `MatchBlade` module, which performs the reverse-engineering operation required to populate the design space with real transonic blade geometries. The mathematical foundation ensures G^2 **continuity** everywhere—a property essential in transonic flows where curvature discontinuities can trigger premature shock formation or boundary layer separation.

Section 3.2 introduces the Non-Uniform Rational B-Splines (NURBS) mathematical framework. Section 3.3 describes the blade construction procedure, detailing how camberline and thickness distributions are combined to produce complete blade profiles. Section 3.4 presents the reverse-engineering workflow and defines the design parameters used throughout this work. Section 3.5 discusses the `MISES` flow solver and the computational setup.

3.2 NURBS: Non-Uniform Rational B-Splines

Non-Uniform Rational B-Splines (NURBS) are parametric curves and surfaces that have become the standard representation in Computer-Aided Design systems since their development in the late 1960s and early 1970s. Their widespread adoption is due to a combination of favorable mathematical properties, intuitive geometric interpretation, and efficient computational algorithms.

A NURBS curve is defined as a parametric curve

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\, w_i\, \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u)\, w_i}, \quad 0 \le u \le 1, \tag{3.1}$$

where p is the degree of the curve, the coefficients $\mathbf{P}_i$ and w_i are the coordinates and weights of the $n+1$ control points, and $N_{i,p}$ are B-spline basis functions.

The basis functions are defined on a non-decreasing, clamped **knot vector**

$$\mathbf{U} = [\underbrace{0, \ldots, 0}_{p+1}, u_{p+1}, \ldots, u_n, \underbrace{1, \ldots, 1}_{p+1}] \in \mathbb{R}^{r+1} \tag{3.2}$$

with $r = n + p + 1$. The term "clamped" indicates that the first and last knot values are repeated $p+1$ times, which ensures that the curve interpolates the first and last control points.

The B-spline basis functions are computed recursively according to the Cox-de Boor formula

$$N_{i,0}(u) = \begin{cases} 1, & \text{if } u_i \le u < u_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

and

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \tag{3.4}$$

NURBS curves possess several mathematical properties that make them particularly well-suited for turbomachinery blade parametrization. First, they exhibit **affine invariance**, meaning that geometric transformations such as rotations, translations, and scalings can be applied to a NURBS curve by simply transforming its control points. Second, NURBS curves satisfy the **convex hull property**: the curve is always contained within the convex hull of its control points, which prevents the geometry from exhibiting unphysical behavior far from the region defined by the control points. Third, NURBS curves interpolate their endpoints and are tangent to the control polygon at the endpoints, a property that simplifies the enforcement of geometric constraints at the leading and trailing edges of turbine blades. Finally, NURBS represent a generalization of both Bézier curves (obtained when $p = n$) and B-splines (obtained when all weights

are equal), providing a unified mathematical framework.

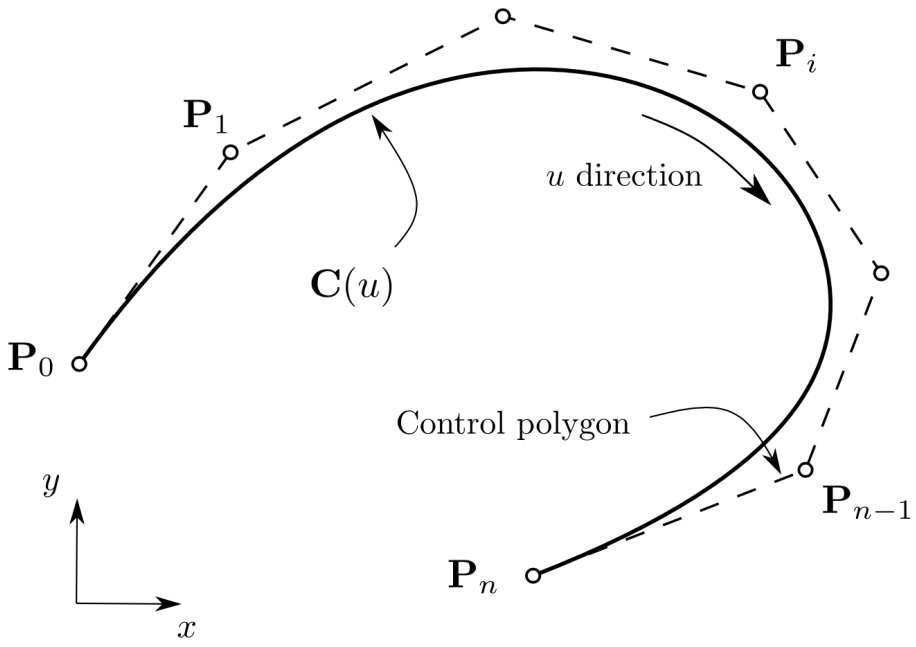


Figure 3.1: NURBS curve construction (Agromayor et al. [10])

An additional advantage of NURBS is their ability to exactly represent conic sections such as circles and ellipses through appropriate choice of control point weights [19]. This capability is particularly useful for defining the leading and trailing edge radii of turbine blades, where circular arcs are commonly prescribed.

In contrast, polynomial-based representations can only approximate such curves. The control point weights w_i in the NURBS definition provide this additional degree of freedom, although in the `ParaBlade` framework the weights are typically set to unity (reducing NURBS to B-splines) unless specific geometric requirements dictate otherwise.

For turbomachinery applications, the ability of NURBS to guarantee high-order geometric continuity is critical. G^2 **continuity**, which requires that both the tangent vector and the curvature are continuous along the blade surface, is essential to avoid pressure distribution anomalies and flow separation in transonic flows. As will be shown in the following section, the `ParaBlade` method constructs blade geometries using NURBS curves in a way that ensures G^2 continuity everywhere, including at the leading and trailing edges where discontinuities are most likely to occur in other parametrization methods.

3.3 `ParaBlade`: 2D Blade Construction

The two-dimensional blade parametrization in `ParaBlade` is based on typical turbomachinery design variables. The blade geometry is generated by first defining a camberline

and subsequently imposing independent upper and lower thickness distributions in the direction normal to the camberline. The construction procedure ensures G^2 continuity at the junction between the upper and lower surfaces, including at the leading and trailing edges.

3.3.1 Camberline Definition

The camberline $\mathbf{C}_c(u)$ is defined as a cubic B-spline curve with four control points $\mathbf{P}_0^c, \mathbf{P}_1^c, \mathbf{P}_2^c, \mathbf{P}_3^c$, as illustrated in Figure 3.1. The coordinates of the control points are determined from the design variables according to

$$\mathbf{P}_0^c = \begin{bmatrix} x_{\text{in}} \\ y_{\text{in}} \end{bmatrix}, \tag{3.5}$$

$$\mathbf{P}_1^c = \mathbf{P}_0^c + d_{\text{in}} \begin{bmatrix} c\cos(\theta_{\text{in}}) \\ c\sin(\theta_{\text{in}}) \end{bmatrix}, \tag{3.6}$$

$$\mathbf{P}_2^c = \mathbf{P}_3^c - d_{\text{out}} \begin{bmatrix} c\cos(\theta_{\text{out}}) \\ c\sin(\theta_{\text{out}}) \end{bmatrix}, \tag{3.7}$$

$$\mathbf{P}_3^c = \mathbf{P}_0^c + \begin{bmatrix} c\cos(\xi) \\ c\sin(\xi) \end{bmatrix}, \tag{3.8}$$

where ξ is the stagger angle (also known as the setting angle), c is the chord length, θ_{in} and θ_{out} are the inlet and outlet metal angles measured from the axial direction, and d_{in} and d_{out} are dimensionless tangent proportions that control the curvature of the camberline near the leading and trailing edges. The axial chord length is given by $c_{\text{ax}} = c\cos(\xi)$.

This construction ensures that the camberline has the specified axial chord length and that the slope at the leading and trailing edges matches the input metal angles, thanks to the endpoint tangency property of B-spline curves.

3.3.2 Thickness Distribution

The upper and lower surfaces of the blade, denoted $\mathbf{C}_u(u)$ and $\mathbf{C}_l(u)$ respectively, are defined as B-spline curves of degree four. This degree is the minimum that guarantees continuous rate of change of curvature (G^2 continuity) at the spline knots. The

coordinates of the control points $\{\mathbf{P}_i^u\}$ and $\{\mathbf{P}_i^l\}$ are computed according to

$$\mathbf{P}_i^{u/l} = \begin{cases} \mathbf{C}_c(\hat{u}_i), & \text{for } i = 0 \\ \mathbf{C}_c(\hat{u}_i) \pm \mathbf{n}(\hat{u}_i) \cdot f(r_{\text{in}}), & \text{for } i = 1 \\ \mathbf{C}_c(\hat{u}_i) \pm \mathbf{n}(\hat{u}_i) \cdot t_{u/l}(\hat{u}_i), & \text{for } i = 2 : n-2 \\ \mathbf{C}_c(\hat{u}_i) \pm \mathbf{n}(\hat{u}_i) \cdot g(r_{\text{out}}), & \text{for } i = n-1 \\ \mathbf{C}_c(\hat{u}_i), & \text{for } i = n \end{cases} \tag{3.9}$$

where the sampling values $\hat{u}_i$ are given by

$$\hat{u}_i = \begin{cases} 0, & \text{for } i = 0 \\ \frac{i-1}{n-2}, & \text{for } i = 1 : n-1 \\ 1, & \text{for } i = n. \end{cases} \tag{3.10}$$

The upper and lower thickness distributions, $t_u(u)$ and $t_l(u)$, are defined as B-spline polynomials of degree three with control points $\{t_i^u\}$ and $\{t_j^l\}$ specified by the user. The unit vectors normal to the camberline $\mathbf{n}(u)$ are computed from the unit tangent vector $\boldsymbol{\tau}(u)$ according to

$$\mathbf{n}(u) = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} -\tau_y \\ \tau_x \end{bmatrix}, \quad \text{with} \quad \boldsymbol{\tau}(u) = \frac{\dot{\mathbf{C}}_c(u)}{\|\dot{\mathbf{C}}_c(u)\|}, \tag{3.11}$$

where $\dot{\mathbf{C}}_c(u)$ is the derivative of the camberline with respect to u, computed analytically using B-spline derivative formulas.

3.3.3 Leading and Trailing Edge Curvature

The functions $f(r_{\text{in}})$ and $g(r_{\text{out}})$ appearing in Equation (3.9) are used to impose the radii of curvature r_{in} and r_{out} at the leading and trailing edges, ensuring that the parametrization satisfies G^2 continuity by construction. This feature is critical for the aerodynamic design of turbomachinery blades because a sudden change in curvature can cause a spike in the surface pressure distribution or trigger local flow separation, particularly in transonic regimes where shock-boundary layer interaction is sensitive to surface geometry.

The derivation of the functions $f(r)$ and $g(r)$ is based on endpoint curvature formulas for NURBS curves. For a parametric curve $\mathbf{C}(u)$, the curvature is given by

$$\kappa(u) = \frac{\|\ddot{\mathbf{C}}(u) \times \dot{\mathbf{C}}(u)\|}{\|\dot{\mathbf{C}}(u)\|^3}. \tag{3.12}$$

By evaluating the first and second derivatives of a NURBS curve at its endpoints and imposing a specified radius of curvature $r = 1/\kappa$, the location of the second and second-to-last control points can be determined such that the blade profile exhibits the desired curvature at the leading and trailing edges. The detailed mathematical derivation is provided by Agromayor et al. [10].

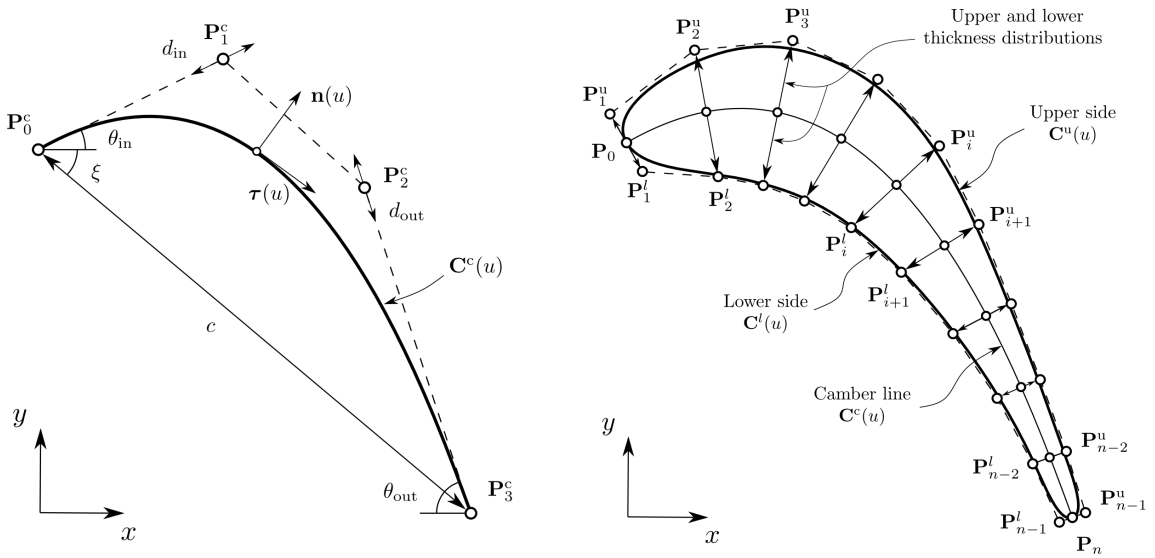
3.3.4 Blade Profile Assembly

Once the upper and lower sides are defined according to Equation (3.9), they are combined into a single B-spline curve

$$\mathbf{C}_b(u) = \mathbf{C}_l \cup \mathbf{C}_u \tag{3.13}$$

that represents the entire blade profile. The construction guarantees that the blade surface is G^2 continuous everywhere, including at the leading edge ($u = 0$), the trailing edge ($u = 1$), and at the junction between the upper and lower surfaces.

This property distinguishes `ParaBlade` from many other parametrization methods that achieve G^2 continuity along most of the blade but introduce curvature discontinuities at the trailing edge, which is particularly problematic for transonic turbine design where even small geometric irregularities can significantly affect shock wave position and strength.



(a) Camber line construction. **(b)** Blade profile construction.

Figure 3.2: Camber line and blade construction

3.4 MatchBlade: Reverse Engineering Workflow

The primary motivation for adopting the ParaBlade framework was the availability of the MatchBlade module, which solves the inverse problem of extracting parametric representations from existing blade geometries. Given a discrete point cloud describing a real turbine cascade—typically provided as (x, y) coordinates in a text file—MatchBlade recovers the 21 design parameters that reproduce the geometry within the ParaBlade construction framework. These parameters can then be systematically varied using sampling strategies (Chapter 4), reconstructed into blade geometries via the complementary MakeBlade module, and evaluated through CFD simulation in MISES.

3.4.1 Optimization Formulation

The blade matching problem is formulated as a nonlinear least-squares minimization:

$$\min_{\boldsymbol{\theta}} \quad f(\boldsymbol{\theta}) = \|\mathbf{x}_{\texttt{ParaBlade}}(\boldsymbol{\theta}) - \mathbf{x}_{\text{target}}\|_2^2 \tag{3.14}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{21}$ is the vector of design parameters, $\mathbf{x}_{\text{target}}$ represents the discrete point cloud of the reference blade, and $\mathbf{x}_{\texttt{ParaBlade}}(\boldsymbol{\theta})$ denotes the corresponding points on the parametrized blade surface generated by ParaBlade. The objective function quantifies the root-mean-square (RMS) deviation between the target geometry and its parametric approximation.

The optimization is performed using the L-BFGS-B algorithm [**Byrd1995**], a limited-memory quasi-Newton method capable of handling bound constraints on the design variables. This algorithm was selected for its superior convergence properties compared to gradient-free methods, despite the increased computational cost per iteration due to finite-difference gradient approximations.

3.4.2 Initial Guess Strategy

The 21-dimensional parameter space exhibits a fundamental degeneracy: infinitely many parameter combinations can produce the *identical* blade geometry. This non-uniqueness arises because the camberline in constructive parametrization methods does not correspond to the physical meanline of the blade but rather serves as a geometric scaffold upon which asymmetric thickness distributions are imposed. Multiple combinations of camberline definition and thickness control points can yield the same final (x, y) coordinates and, consequently, identical aerodynamic behavior in CFD.

To guide the optimizer toward a consistent and interpretable solution, an initial guess strategy is employed:

1. **Normalization**: The target blade coordinates are normalized to unit axial chord ($c_{\text{ax}} = 1$).

2. **Camberline parameter initialization**: The stagger angle ξ, inlet metal angle θ_{in}, and outlet metal angle θ_{out} are measured directly from the target blade geometry and imposed as initial values. This ensures that the recovered camberline parameters remain close to the physical orientation and flow angles of the original cascade, providing a geometrically meaningful baseline even though the camberline itself is a constructive artifact.

This approach is standard practice in blade parametrization and does not eliminate the degeneracy—it simply constrains the solution to lie in a region of parameter space where camberline angles have straightforward geometric interpretation.

3.4.3 Convergence and Accuracy

Starting from the initialized parameter vector, the L-BFGS-B algorithm typically converges within 200 iterations. The matching accuracy achieved is sufficient to ensure that the parametric representation is indistinguishable from the original geometry for CFD purposes. Deviations remain below 0.01% of the blade meanline length on average, with maximum local errors under 0.3%—well within the precision of typical blade fabrication processes and negligible relative to mesh discretization errors in `MISES`. Figure 3.3 illustrates a representative matching result, showing the original discrete point cloud overlaid with the optimized `ParaBlade` geometry.

The `MatchBlade` procedure was applied to multiple reference transonic cascades to extract their parametric representations. This reverse-engineering capability proved essential for populating the design space with geometries known to produce *feasible* flow solutions—blades that converge in CFD and exhibit physically realistic aerodynamic behavior.

An important observation emerged during this process: the 21 design parameters extracted from successful transonic blades are **strongly correlated**. Certain combinations of stagger angle, metal angles, and thickness distributions recur across different cascade families, while other mathematically valid combinations consistently produce non-convergent or aerodynamically pathological geometries when reconstructed via `MakeBlade` and simulated in `MISES`. This correlation structure, quantified and exploited in Chapter 4, became the foundation for the intelligent sampling strategy used to generate the surrogate model training database.

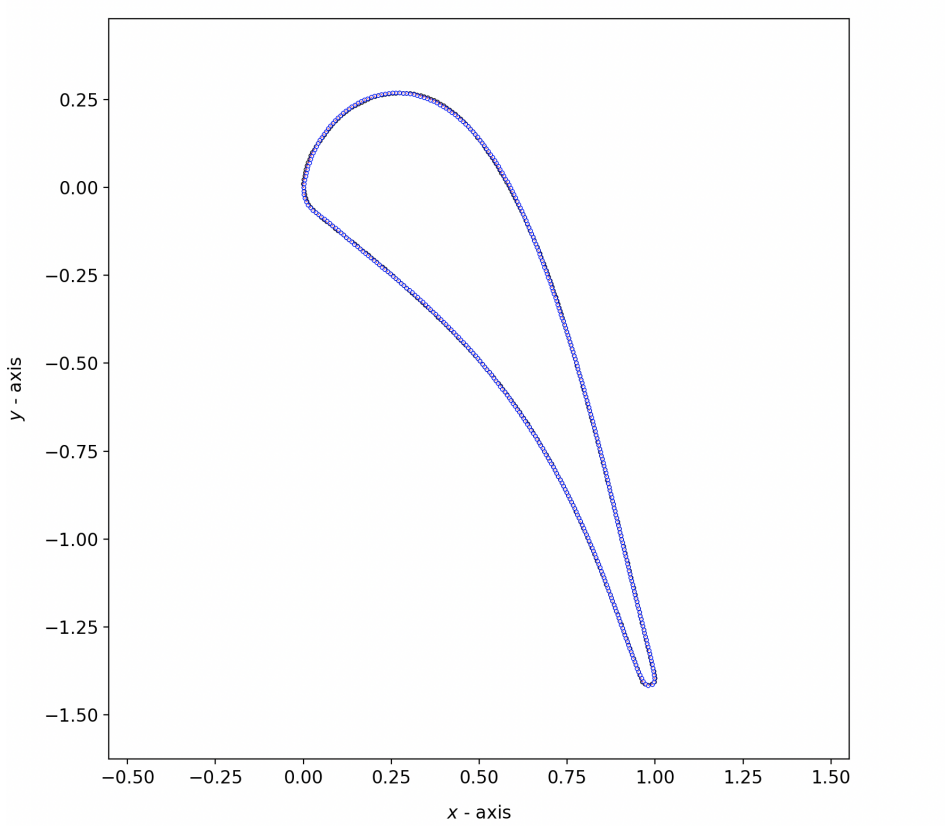


Figure 3.3: Example of `MatchBlade` output (LS89): original blade geometry (blue discrete points) overlaid with optimized `ParaBlade` representation (red discrete points). Maximum deviation = 0.004025 m ($c_{ax} = 1$ m)

3.4.4 Design Parameters

The complete blade geometry is defined by 21 design variables, organized into three functional groups:

Camberline Geometry (5 parameters):

- ξ: Stagger angle (setting angle)
- θ_{in}, θ_{out}: Inlet and outlet metal angles, measured from the axial direction
- d_{in}, d_{out}: Dimensionless tangent proportions controlling camberline curvature near the leading and trailing edges

Edge Radii (2 parameters):

- r_{in}: Leading edge radius of curvature
- r_{out}: Trailing edge radius of curvature

Thickness Distribution (14 parameters):

- $t_1^u, \ldots, t_8^u$: Control points defining the thickness distribution on the suction (upper) side
- $t_1^l, \ldots, t_6^l$: Control points defining the thickness distribution on the pressure (lower) side

The asymmetry in thickness control points—8 on the suction side versus 6 on the pressure side—reflects the aerodynamic requirements of transonic cascades. The suction surface experiences more complex flow phenomena, including acceleration to locally supersonic speeds, shock wave formation, and shock-boundary layer interaction. These features are sensitive to subtle variations in surface curvature, particularly in the rear half of the blade where the supersonic region terminates.

The additional control points on the suction side provide the geometric resolution necessary to capture these details, while the pressure side, which typically exhibits a smoother, more benign pressure distribution, requires fewer degrees of freedom. This parametrization provides the foundation for all subsequent design space exploration, optimization, and surrogate modeling activities presented in the following chapters.

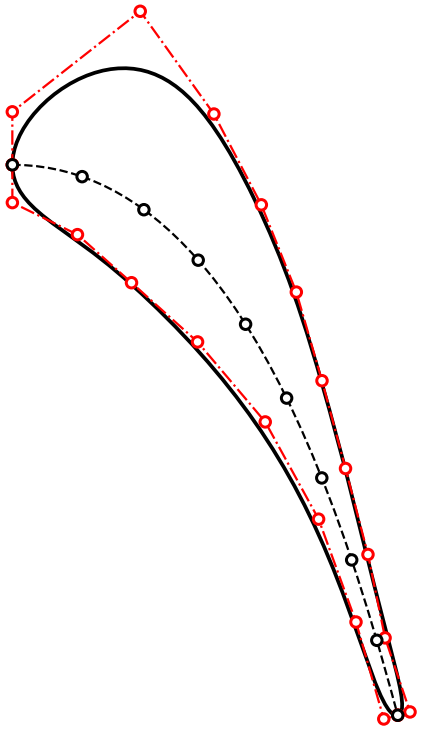


Figure 3.4: LS89 parametric representation via `MatchBlade`. The blade geometry is shown with its underlying NURBS control points (red circles) defining the camberline and asymmetric thickness.

3.5 MISES CFD Solver

3.5.1 Introduction and Computational Approach

The data-driven design framework requires evaluating tens of thousands of blade geometries across the transonic operating range. Generating such a database using three-dimensional Reynolds-Averaged Navier-Stokes (RANS) solvers would demand prohibitive computational resources—months of wall-clock time even on modern computing clusters. This work employs MISES (Multiple-blade Interacting Streamtube Euler Solver), a specialized two-dimensional cascade analysis code developed by Drela and Youngren at MIT [20, 21].

MISES achieves computational efficiency through a **viscous-inviscid interaction** formulation. The inviscid core flow is solved using the Euler equations formulated in streamtube coordinates, where the computational grid aligns with streamlines. Viscous effects—boundary layer development, transition, and separation—are captured through **integral boundary layer equations** coupled to the Euler solution via transpiration velocities.

This hybrid approach retains the essential physics governing transonic cascade aerodynamics (shock waves, shock-boundary layer interaction, profile losses) while providing simulation times of **2–3 seconds** per case: two to three orders of magnitude faster than RANS solvers. The streamtube formulation simplifies shock capturing and naturally accommodates the blade-to-blade periodicity of cascade flows.

The computational domain extends from inlet to outlet planes typically positioned one axial chord upstream and downstream of the blade row, respectively. The solution procedure employs the Newton-Raphson method to converge the coupled nonlinear system, with convergence typically achieved in 50–100 iterations depending on flow complexity and grid resolution.

3.5.2 Governing Equations and Numerical Method

The inviscid flow field is governed by the two-dimensional steady **Euler equations**. These are obtained from the compressible Navier-Stokes equations presented in Section 2 by setting the viscous stress tensor and thermal conduction terms to zero ($\boldsymbol{\tau} = 0$, $k = 0$). For steady flow, the conservation equations reduce to:

$$\nabla \cdot (\rho \vec{V}) = 0 \tag{3.15}$$

$$\rho(\vec{V} \cdot \nabla)\vec{V} = -\nabla p \tag{3.16}$$

$$\nabla \cdot [(E + p)\vec{V}] = 0 \tag{3.17}$$

where:

- ρ is the fluid density
- $\vec{V}$ is the velocity vector
- p is the static pressure
- $E = \rho(e + \frac{1}{2}V^2)$ is the total energy per unit volume with e the internal energy

For an ideal gas, the system is closed by the equation of state $p = \rho RT$ and the caloric relation $e = c_v T$. `MISES` solves these equations in a **streamtube-coordinate system** where one computational coordinate aligns with the local streamline direction. This formulation simplifies the treatment of curved blade passages and provides robust shock capturing in transonic flows.

The key distinction from the Navier-Stokes equations is the absence of viscous diffusion and heat conduction. Consequently, the Euler equations cannot directly model boundary layer development, flow separation, or viscous dissipation. These effects are instead captured through the integral boundary layer equations described below and coupled to the Euler solution.

Viscous Effects Modeling

Viscous effects in the thin regions adjacent to blade surfaces are modeled using **integral boundary layer equations** rather than resolving the full Navier-Stokes viscous terms. The integral formulation tracks the momentum thickness θ, displacement thickness δ^*, and shape factor $H = \delta^*/\theta$ as the boundary layer develops from the leading edge through transition and potential separation to the trailing edge. Closure requires empirical correlations for skin friction and shape factor as functions of Reynolds number and boundary layer state.

The coupling to the inviscid Euler solution is achieved through **transpiration velocities**. The boundary layer displaces the inviscid streamlines away from the physical surface by approximately the displacement thickness δ^*, imposed on the Euler solver as a normal velocity component at the effective body. The Newton-Raphson method solves the coupled system iteratively, typically converging in a few cycles for attached flows. The wake downstream of the **blunt trailing edge** is treated as a viscous layer with the Kutta condition applied at the trailing edge corners, which slightly underpredicts base pressure but avoids resolving the complex base region flow.

Transition Model

The **Abu-Ghannam and Shaw** bypass transition model is employed, calibrated for turbomachinery flows with elevated freestream turbulence (3–10%). For all database simulations, transition was forced at 3% axial chord on both surfaces (`XTRANS1` = `XTRANS2` = 0.03), ensuring fully turbulent boundary layers representative of high-Reynolds-number aircraft engine conditions. This eliminates sensitivity to turbulence intensity specification across the diverse blade geometries in the database.

Boundary Conditions

`MISES` automatically selects the appropriate boundary condition formulation based on the local Mach number regime. For the high-subsonic to transonic cascades analyzed in this work, the flow remains subsonic-axial throughout the domain ($M_{\text{axial}} < 1$), even when the local relative Mach number exceeds unity ($M > 1$). The applied boundary conditions are:

- **Inlet Boundary:** At the inlet plane, the flow state is determined by specifying the total pressure (P_{01}), total temperature (T_{01}), and the inlet flow angle (α_1). It is worth noting that `MISES` solves the flow equations in non-dimensional form. Therefore, the inlet total quantities are set to unity ($P_{01} = 1.0$, $T_{01} = 1.0$), and all thermodynamic variables in the field are computed relative to these reference values. For rotating cascades, the code enforces the conservation of rothalpy and relative swirl across the inlet domain to properly set the stagnation quantities on each streamline.

- **Outlet Boundary:** In the transonic regime investigated ($0.8 \leq M_2 \leq 1.2$), the axial Mach number remains subsonic ($M_{\text{axial}} < 1$), requiring the imposition of a single thermodynamic variable at the exit. Unlike incompressible formulations where the exit flow angle might be fixed, in this work the isentropic exit Mach number M_2 is prescribed. Given the normalization, this is equivalent to imposing a fixed back-pressure ratio (P_2/P_{01}). The exit flow angle α_2 is not constrained but is part of the solution, determined by the Kutta condition at the trailing edge and mass flow conservation. This approach allows the solver to correctly predict the deviation angle and the shock system structure without spurious constraints on the flow direction.

3.5.3 Input and Output File Structure

`MISES` operates through a suite of programs communicating via formatted text files. The primary modules and their associated files are:

`ISET`: Grid Generation and Initialization

- **Reads**: `blade.xxx` (blade coordinates), `gridpar.xxx` (mesh parameters)
- **Generates**: `idat.xxx` (initial solution file with computational grid)

The grid generation process constructs a streamline-aligned mesh based on parameters specified in `gridpar.xxx`, including number of streamtubes (pitchwise direction), axial discretization upstream/downstream of the blade, and blade surface point distribution. Grid quality significantly affects solution accuracy and convergence robustness, particularly in transonic cases with strong shock waves.

`ISES`: Flow Solver

- **Reads**: `idat.xxx` (grid + solution), `ises.xxx` (flow conditions and solver parameters)
- **Writes**: `idat.xxx` (updated solution)

The `ises.xxx` file specifies inlet/outlet boundary conditions (angles, Mach numbers, Reynolds number), convergence tolerances, transition parameters, and various solver options. The solution file `idat.xxx` can serve as a restart for subsequent calculations, enabling continuation from partially converged solutions or exploration of neighboring operating points with minimal computational overhead.

`IPLOT`: Post-Processing and Visualization

- **Reads**: `idat.xxx` (converged solution)
- **Displays**: Blade surface pressure distributions, Mach contours, boundary layer development, wake profiles

For automated database generation, post-processing extracts key aerodynamic quantities (outlet flow angle, total pressure loss coefficient, blade loading) programmatically from the solution file without invoking the graphical interface.

3.6 Transonic Validation and Computational Setup Selection

The computational setup employed throughout this work was established through a systematic validation and sensitivity study based on the methodology of Andrew and Kahveci [22]. However, while that study focused on demonstrating `MISES` accuracy against experimental data, the present investigation extended the analysis to identify the optimal trade-off between solution accuracy and computational speed.

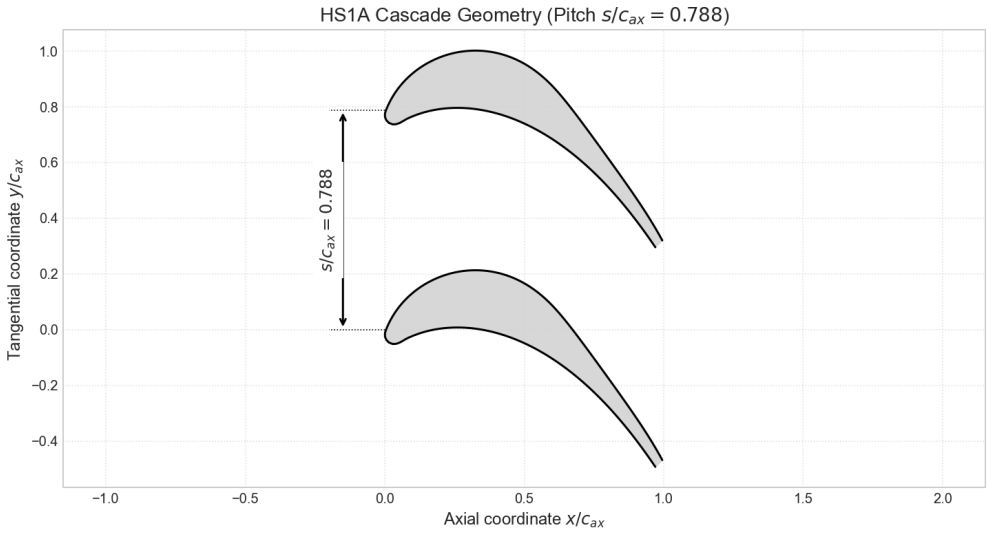


Figure 3.5: HS1A cascade geometry

HS1A Test Case and Operating Conditions

The validation study employed the HS1A high-pressure turbine cascade, a front-loaded midspan section extensively characterized in experimental campaigns by Jouini et al. [23] and Corriveau and Sjolander [24]. Three exit **static pressure** (Mach numbers) were analyzed:

- $\frac{p_2}{p_{01}} = 0.535 \quad \rightarrow \quad M_2 = 0.96$
- $\frac{p_2}{p_{01}} = 0.479 \quad \rightarrow \quad M_2 = 1.06$
- $\frac{p_2}{p_{01}} = 0.429 \quad \rightarrow \quad M_2 = 1.14$

All validation cases were performed at the design incidence of the HS1A cascade, corresponding to an effective incidence of $-4.5°$. The transition model was set to Abu-Ghannam and Shaw with 4% turbulence intensity. Outlet flow quantities were measured at 40% axial chord downstream of the trailing edge.

3.6.1 Mesh Type Sensitivity

MISES offers two primary grid topologies for cascade flows, controlled by the gridpar.xxx parameter ITYPE:

- F: Periodic H-type grid aligned with the blade passage
- T: Offset I-type grid

These settings are specified independently for the inlet (ITYPE(1)) and outlet (ITYPE(2)) portions of the domain. Three practical combinations were tested:

- F-F: Periodic grid throughout (fastest)
- F-T: Periodic inlet, offset outlet (recommended by MISES User Guide for high-speed turbines)
- T-F: Offset inlet, periodic outlet (selected in the validation paper)

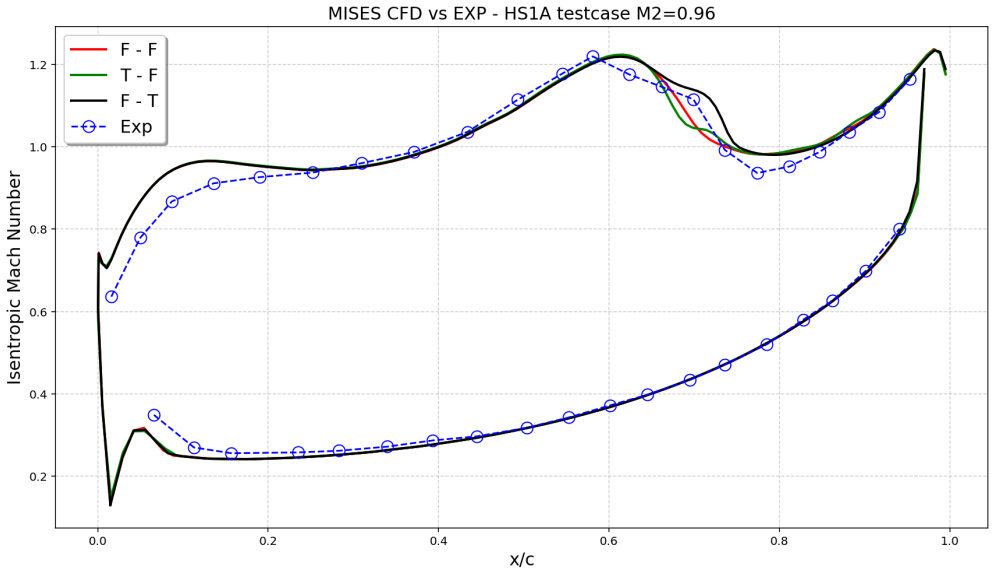


Figure 3.6: Comparison of blade surface isentropic Mach number distributions with F-F, F-T, and T-F grid configurations against experimental data ($M_2 = 0.96$)

Figures 3.6, 3.7, and 3.8 show blade loading distributions at the three test Mach numbers for all three grid configurations compared against experimental data. When MISES accurately captures the blade loading (cases $M_2 = 0.96$ and $M_2 = 1.14$), all three mesh types produce nearly identical results with minimal deviation from measurements.

At $M_2 = 1.06$—where complex shock reflection patterns challenge the solver—all three configurations exhibit comparable discrepancies relative to experiments, with F-T showing marginally better agreement. This slight accuracy advantage of F-T comes at the cost of approximately 10× longer computational time compared to F-F.

Given the minimal differences in predictive capability and the overwhelming importance of simulation throughput for database generation, the remainder of the grid sensitivity study employed the `F-F` configuration. This choice prioritizes computational efficiency while maintaining acceptable accuracy, and as demonstrated below, proves to be the final selection for all database simulations.

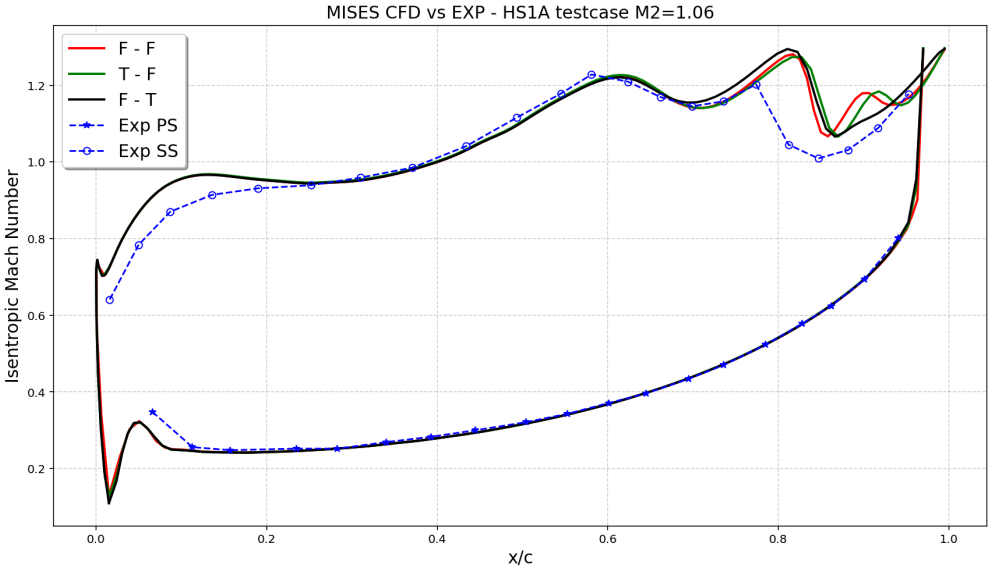


Figure 3.7: Comparison of blade surface isentropic Mach number distributions with `F-F`, `F-T`, and `T-F` grid configurations against experimental data ($M_2 = 1.06$)

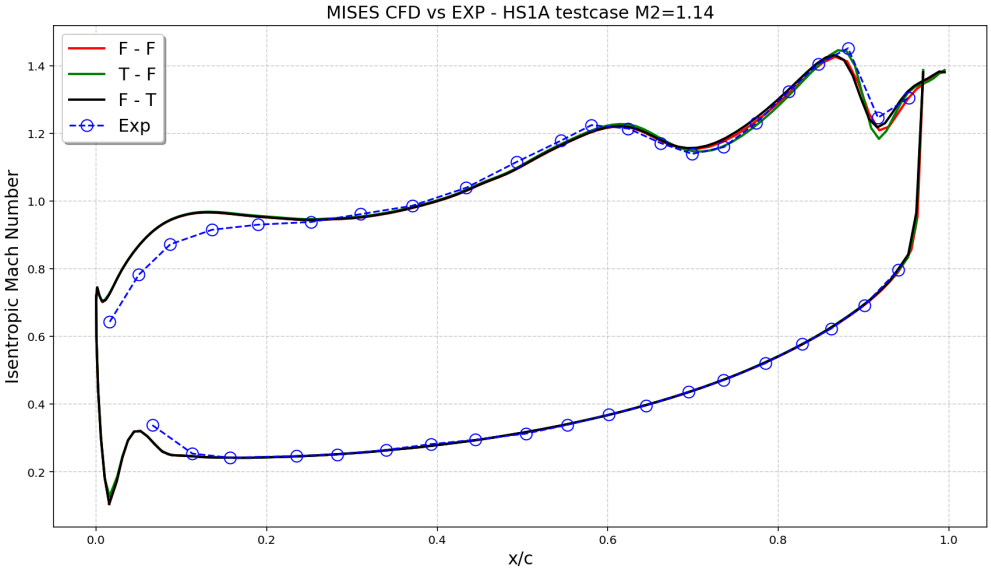


Figure 3.8: Comparison of blade surface isentropic Mach number distributions with `F-F`, `F-T`, and `T-F` grid configurations against experimental data ($M_2 = 1.14$)

3.6.2 Grid Density Sensitivity

With the `F-F` mesh topology fixed, grid density parameters were tuned using the $M_2 = 1.06$ case—the most challenging condition where `MISES` exhibited the greatest sensitivity to numerical parameters. Three grid refinement variables were investigated.

Inlet/Outlet Axial Discretization (`NIN`, `NOUT`)

These parameters control the number of axial grid points upstream and downstream of the blade row. Three refinement levels were tested:

- 20 points
- 30 points
- 40 points

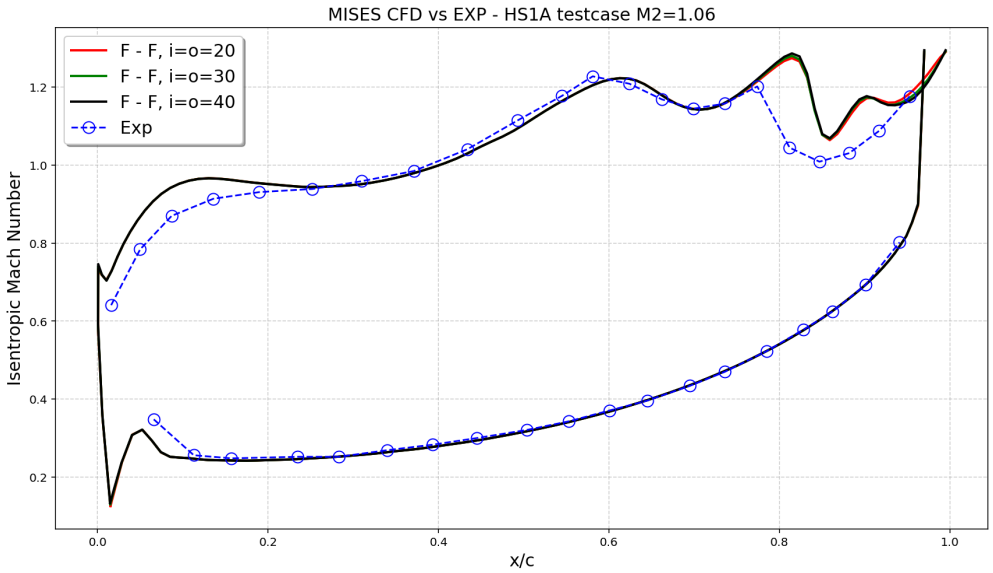


Figure 3.9: Blade surface loading at $M_2 = 1.06$ for three axial discretization levels ($i = o = 20, 30, 40$)

Figure 3.9 shows that blade loading predictions are insensitive to this parameter within the tested range. The baseline value of 30 points (matching the validation paper) was retained for both inlet and outlet regions.

Streamtube Discretization (`NSTRM`)

The number of streamtubes in the pitchwise direction is one of the most influential parameters for computational cost. Three levels were evaluated:

- 20 streamtubes
- 30 streamtubes
- 40 streamtubes

A modest accuracy improvement was observed between 20 and 30 streamtubes, with 40 showing saturation. However, the trade-off analysis favored `NSTRM` = 20 as the optimal choice for this work's objectives. The gain in precision from additional streamtubes was deemed insufficient to justify the proportional increase in simulation time given the database scale requirements.

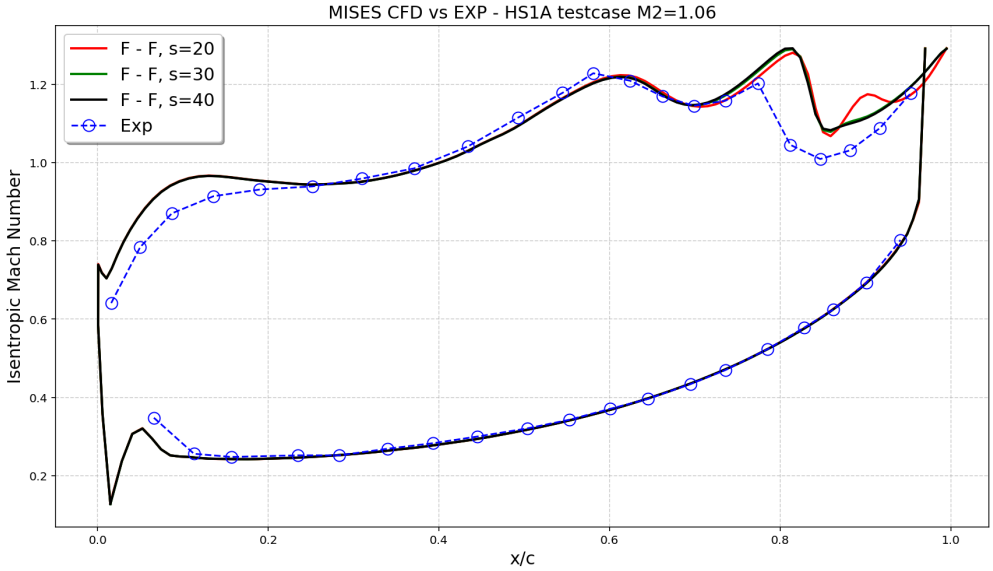


Figure 3.10: Blade surface loading at $M_2 = 1.06$ for three streamtube discretization levels ($s = 20, 30, 40$)

Blade Surface Discretization (`N`)

The number of points distributed along the blade surface was found to be the most effective parameter for improving solution quality with minimal computational overhead. Three refinement levels were tested:

- $N = 100$ (validation paper baseline)
- $N = 150$
- $N = 200$

Figure 3.11 demonstrates that blade loading predictions show negligible sensitivity to N within this range—all three cases capture the pressure distribution equally well.

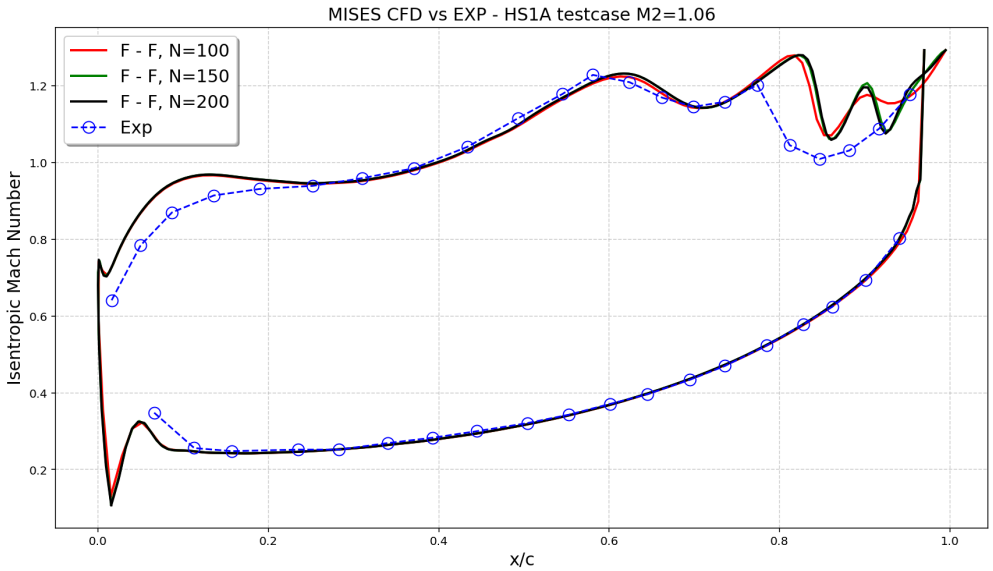


Figure 3.11: Blade surface loading at $M_2 = 1.06$ for three blade surface discretization levels ($N = 100, 150, 200$)

The critical difference emerges in loss prediction accuracy. Loss coefficients were computed using the total pressure loss coefficient W_t defined as:

$$W_t = \frac{P_{01} - P_{02}}{P_{02} - P_2} \tag{3.18}$$

where:

- P_{01} is the inlet total pressure;
- P_{02} is the exit total pressure;
- P_2 is the exit static pressure.

This formulation was selected because it corresponds directly to the experimental measurement protocol employed in the HS1A cascade tests [22], enabling quantitative comparison with measured loss levels. Figure 3.12 presents the total pressure loss coefficient W_t as a function of exit Mach number for the three blade surface discretizations alongside experimental measurements. Three key observations emerge from this comparison:

- The `F-T` mesh configuration (with $N = 100$) provides the closest agreement with experimental loss levels, establishing the benchmark for accuracy.
- The `F-F` grid with $N = 100$ underpredicts losses at all Mach numbers, consistent with its faster but less refined flow resolution.
- Increasing to $N = 150$ while maintaining the `F-F` topology yields good improvement.

Blade surface resolution is the most effective parameter for improving loss accuracy in `F-F` grids without sacrificing computational speed. The choice of $N = 150$ represents the optimal compromise: simulation times remain an order of magnitude faster than `F-T` configurations while achieving loss prediction quality sufficient for surrogate model training.

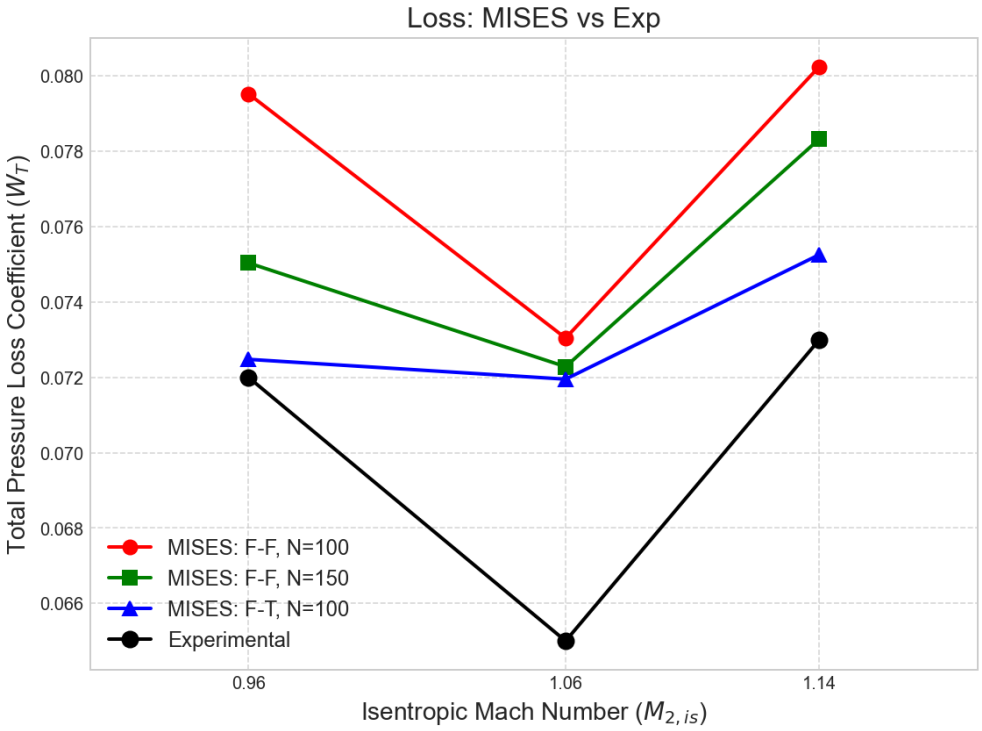


Figure 3.12: Total pressure loss coefficient W_t versus exit Mach number. The plot compares three grid configurations against experimental measurements.

Exit flow angle and exit Mach number predictions depend more strongly on streamtube discretization (`NSTRM`) and inlet/outlet axial grid density (`NIN`, `NOUT`) than on blade surface resolution N. The analysis presented in Figures 3.13 and 3.14 verifies that the selected setup (`F-F`, $N = 150$) substantially improves loss prediction accuracy without compromising the accuracy of other aerodynamic quantities.

Figure 3.13 shows that exit flow angle predictions deteriorate slightly with increasing exit Mach number. However, all three configurations remain within approximately 0.5° of experimental measurements across the full range.

Figure 3.14 presents exit Mach number predictions. The most challenging case is $M_2 = 1.06$, where complex shock reflection structures prevented accurate loading capture. Even for this difficult condition, all configurations predict exit Mach within 0.75% of experimental values.

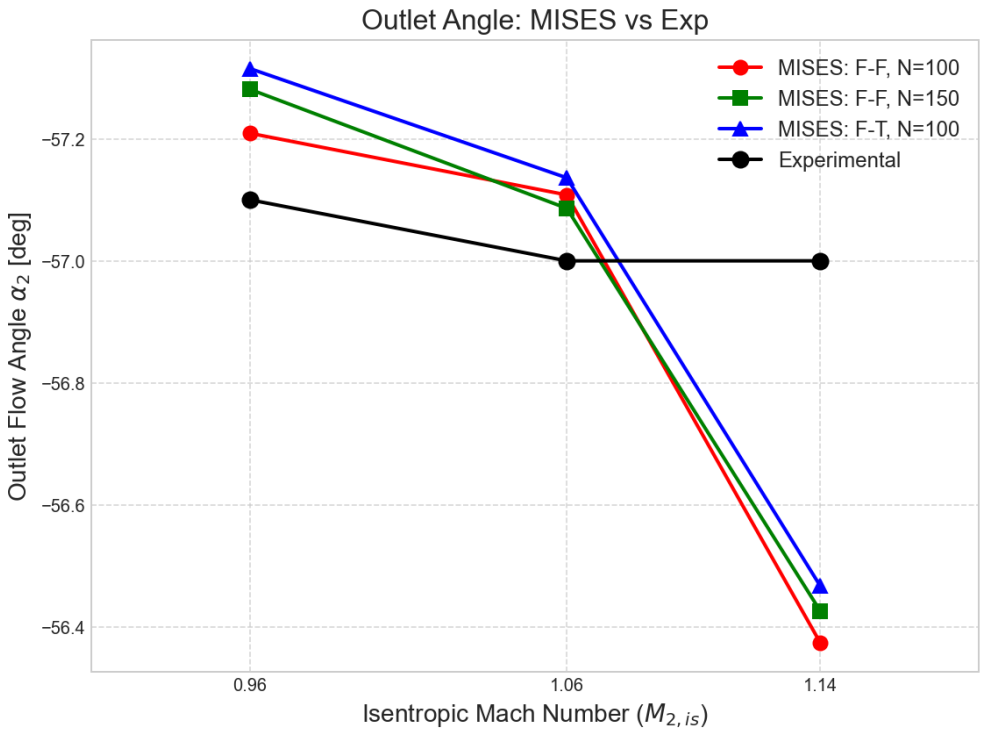


Figure 3.13: Exit flow angle predictions across the transonic Mach number range for the three final mesh configurations against experimental data.

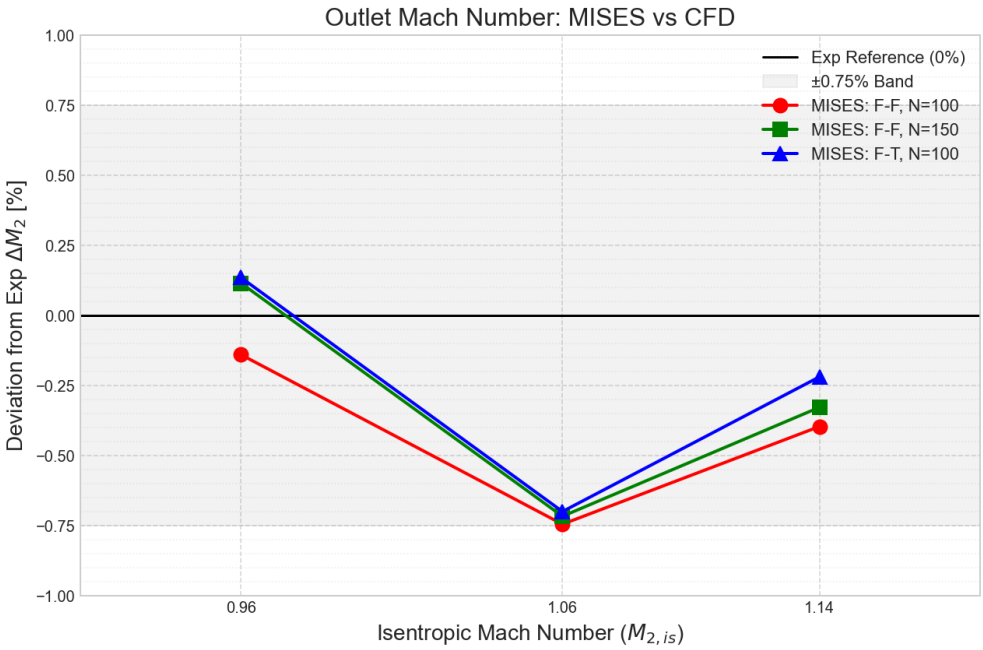


Figure 3.14: Exit Mach number predictions across the transonic Mach number range for the three final mesh configurations against experimental data.

The selected computational parameters (`F-F` grid topology, $N = 150$ blade surface points, `NSTRM` = 20 streamtubes, `NIN` = `NOUT` = 30 inlet/outlet points) provide the optimal balance for database generation. Loss prediction accuracy approaches the `F-T` benchmark while maintaining an order-of-magnitude computational speedup. Predictions of exit flow angle and Mach number remain within experimental uncertainty.

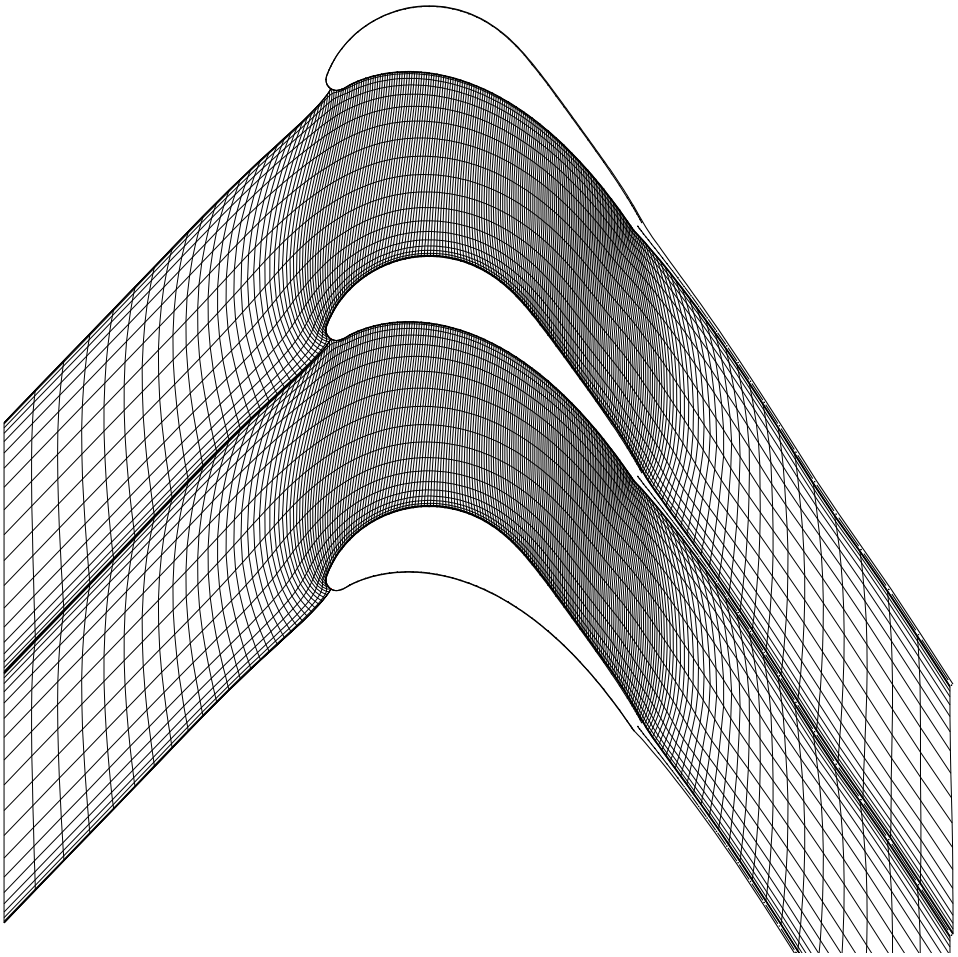


Figure 3.15: Selected computational grid generated by `MISES` for the HS1A cascade.

Table 3.1: Final `MISES` grid topology and density setup for database generation.

Parameter	**Value**
Grid Topology	
Grid type (inlet/outlet)	`F-F` (periodic H-type)
Grid Density	
Blade surface points (N)	150
Streamtubes (`NSTRM`)	20
Inlet axial points (`NIN`)	30
Outlet axial points (`NOUT`)	30

3.7 Artificial Dissipation and MUCON Selection

The `MUCON` parameter (C_μ in the `MISES` formulation) controls the magnitude of the artificial upwind dissipation applied to the streamwise momentum equation. Specifically, `MISES` computes the upwinded speed $\tilde{q}_i$ at each grid node i as

$$\tilde{q}_i = q_i - \mu_i^{(1)}(q_i - q_{i-1}) + \mu_i^{(2)}(q_{i-1} - q_{i-2})\,, \tag{3.19}$$

where the first- and second-order dissipation coefficients are defined as

$$\mu_i^{(1)} = \frac{C_\mu}{\gamma}\,(1 - M_{\text{crit}}) \log\!\left[1 + \exp\!\left(\frac{1 - 1/M^2}{1 - M_{\text{crit}}}\right)\right], \qquad M^2 = \tfrac{1}{2}\Big(M_i^2 + M_{i-1}^2\Big)\,, \tag{3.20}$$

$$\mu_i^{(2)} = \begin{cases} \mu_i^{(1)} & \text{second-order dissipation (}\texttt{MUCON} > 0\text{)}, \\ 0 & \text{first-order dissipation (}\texttt{MUCON} < 0\text{)}. \end{cases} \tag{3.21}$$

The default configuration is `MUCON` $= +1$, which activates second-order dissipation ($\mu^{(2)} = \mu^{(1)}$). This scheme produces crisper shock profiles, reduces spurious total pressure errors, and is the recommended setting for most transonic cascade problems. However, the `MISES` User Guide [21] explicitly warns:

> *"Second order dissipation is not recommended when strong shocks traverse quasi-normal grid lines — e.g. normal shocks on sheared grids, or strong oblique shocks on orthogonal grids."*

Setting `MUCON` $= -1$ disables the second-order term entirely, reverting to a first-order upwinding scheme that, while less accurate in smooth regions, is more robust in the presence of complex shock structures traversing the `F-F` (periodic–periodic) H-type grid adopted here.

Motivation: the C2 test case

The sensitivity of the dissipation scheme to convergence was identified during the early database generation campaigns. The investigation was anchored to a reference operating point designated **C2**, defined by

$$\Big(\alpha_1,\, \alpha_2,\, M_2,\, \text{TE thickness}\Big) = \Big(50^\circ,\, -50^\circ,\, 1.2,\, 3\%\Big), \tag{3.22}$$

which represents one of the most aerodynamically demanding corners of the design space: high inlet incidence, aggressive turning, and a strongly supersonic exit Mach number with complex shock reflection and boundary-layer interaction patterns. This point will be adopted as the reference test case for the hyperparameter tuning of the NSGA-II algorithm in Chapter 5.

Convergence study

Starting from an initial Design of Experiments (DoE) comprising 200 blades with physically acceptable geometries (generated as described in Chapter 4), `MISES` simulations were launched at the C2 operating condition under both dissipation configurations. The results were markedly different:

- `MUCON` $= +1$ (second-order): **51 out of 200** simulations converged (convergence rate: 25.5%);
- `MUCON` $= -1$ (first-order): **101 out of 200** simulations converged (convergence rate: 50.5%).

The doubling of the convergence rate with first-order dissipation is consistent with the User Guide's warning: at $M_2 = 1.2$ on an `F-F` periodic grid, the strong oblique and reflected shocks generate numerical instabilities that second-order upwinding amplifies rather than damps, leading to divergence of the Newton solver.

Impact on aerodynamic quantities

The improved convergence of `MUCON` $= -1$ raises an important question regarding the accuracy of the computed aerodynamic quantities. To quantify the systematic bias introduced by switching dissipation order, the 51 geometries that converged under *both* configurations were compared directly. Three output quantities were examined.

Kinetic energy loss coefficient ξ

The kinetic energy loss coefficient is defined as

$$\xi = 1 - \frac{1 - \left(\frac{p_2}{p_{02}}\right)^{\frac{\gamma-1}{\gamma}}}{1 - \left(\frac{p_2}{p_{01}}\right)^{\frac{\gamma-1}{\gamma}}} \tag{3.23}$$

The comparison between `MUCON` $= +1$ and `MUCON` $= -1$ yields a **MAPE (Mean Absolute Percentage Error) of 8.9%** and a **bias of −8.7%**, indicating that first-order dissipation systematically *underestimates* losses relative to the second-order reference. This bias is physically consistent: first-order upwinding smears the shock structure more aggressively, reducing the captured entropy rise and therefore under-reporting the aerodynamic loss.

Zweifel loading coefficient

The blade loading is quantified via the Zweifel coefficient, defined as the pressure integral around the blade profile normalized by the axial chord and the inlet-to-exit pressure

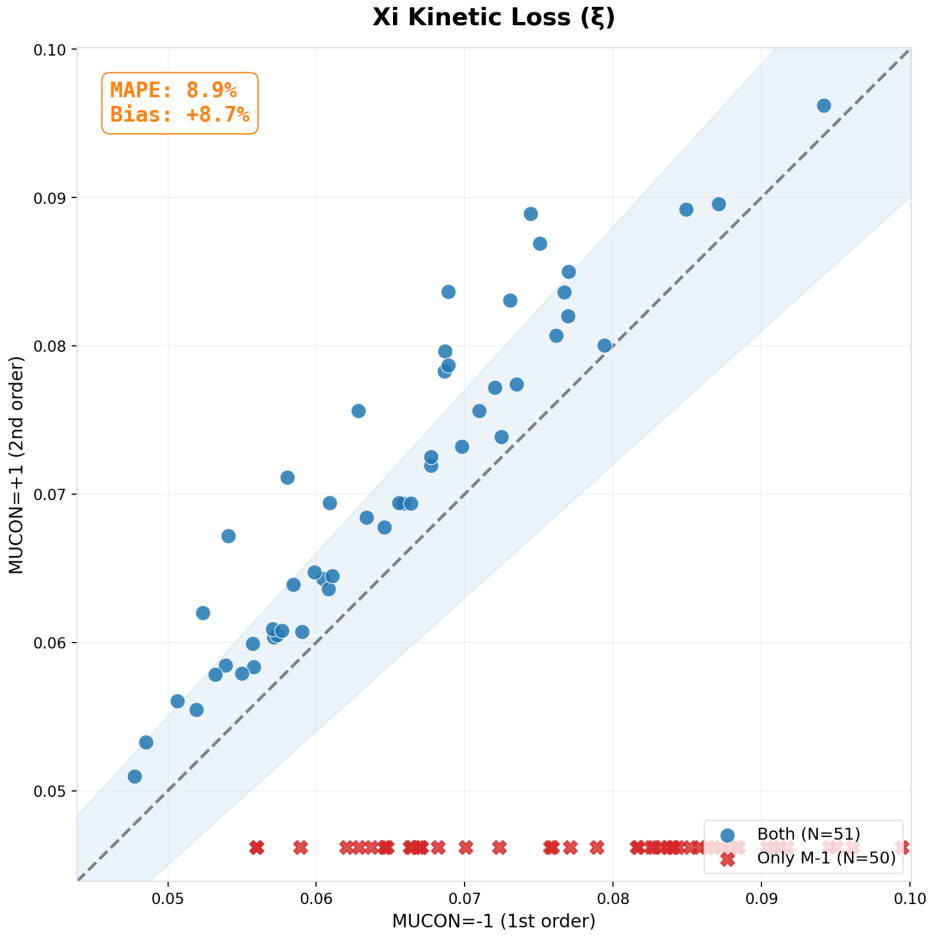


Figure 3.16: Comparison of the kinetic energy loss coefficient ξ computed with second-order ($\texttt{MUCON} = +1$) and first-order ($\texttt{MUCON} = -1$) dissipation.

head:

$$Z_w = \frac{\oint p \, \mathrm{d}x}{c_{\mathrm{ax}} \left(P_{01} - P_2 \right)}, \tag{3.24}$$

The Zweifel coefficient shows negligible sensitivity to the dissipation order: **MAPE of 0.9%** and **bias of −0.9%**. The slight systematic underestimation of loading is inconsequential for the purposes of this work.

Exit flow angle α_2

The predicted exit flow angle shows a **MAPE of 0.5%** and a **bias of +0.5%**: first-order dissipation marginally *overestimates* the exit angle, in the opposite direction to the loss bias, but at a level well within the experimental uncertainty of the cascade measurements.

MUCON selection for database generation

Despite the systematic underestimation of the kinetic energy loss ($\approx 8.7\%$ mean bias), the adoption of $\texttt{MUCON} = -1$ for database generation at high Mach numbers is justified

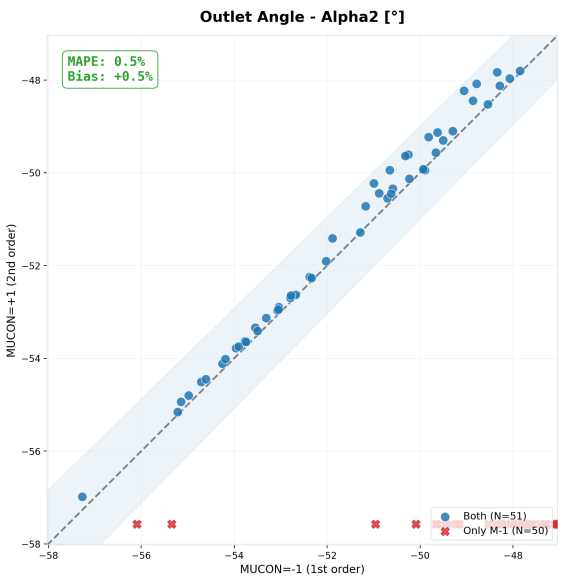


Figure 3.17: Comparison of exit flow angle α_2

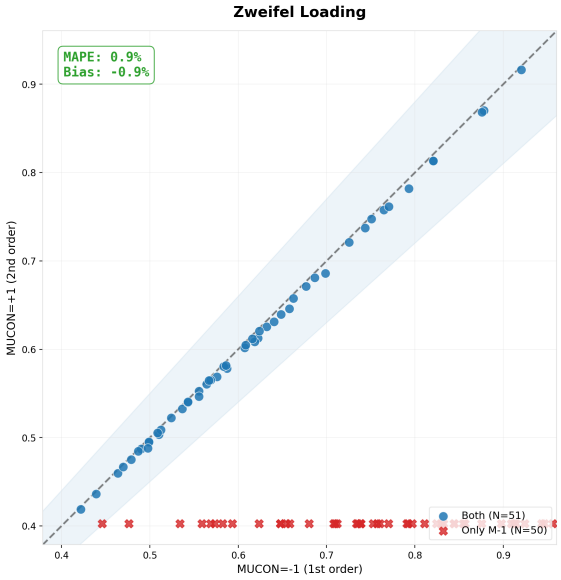


Figure 3.18: Comparison of Zweifel coefficient

by two complementary arguments:

- **Rank preservation.** The relative ordering of blades by aerodynamic efficiency is conserved: a geometry that is superior under $\texttt{MUCON} = -1$ is consistently superior under $\texttt{MUCON} = +1$. This property is essential for surrogate-based optimization, where the surrogate must capture the *topology* of the loss landscape rather than its absolute scale.

- **Dataset coverage.** Retaining only the 25.5% of simulations that converge with second-order dissipation would introduce a severe selection bias, privileging geometries that are—by construction—numerically easier to resolve, not necessarily aerodynamically superior. First-order dissipation preserves a representative and diverse sample of the design space.

The question of *why* certain geometries fail to converge under second-order dissipation—and whether the first-order scheme artificially regularizes their flow fields by smearing physical separations—is addressed in the following section through a comparison with the `ADS CFD` Reynolds-averaged Navier–Stokes solver.

3.7.1 Independent Validation via ADS CFD Solver

`ADS CFD` is a commercial turbomachinery solver built around the **`LEO`** code, described by its developers as a CFD solver with full steady and unsteady turbomachinery capabilities, optimized for GPU acceleration—achieving speedups of up to 120× compared to

conventional CPU-based solvers. The toolset includes the **WAND** meshing code for structured grid generation and the **LEO** solver for the flow computation.

ADS CFD is not the primary tool of this work, but is employed within the research group for data-driven turbomachinery applications. In the present thesis, it serves exclusively as an independent reference solver for the validation of the selected MISES setup for the transonic regime. The computational setup adopted—a structured mesh of approximately 150,000 cells with 15,000 iterations—was not developed from scratch, but inherits a configuration previously validated within the group for both RANS and URANS transonic cascade simulations. Steady RANS computations with the k-ω SST turbulence model were performed for the cases presented in this work.

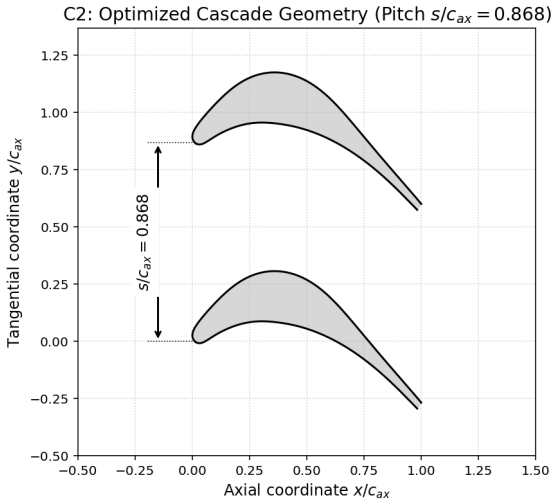


Figure 3.19: Profile of the C2 test blade selected

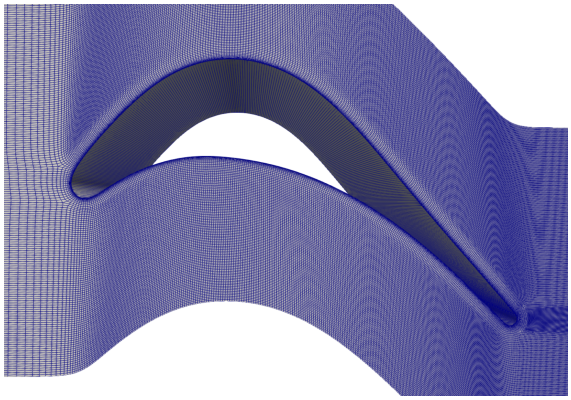


Figure 3.20: Detail of the structured mesh generated

Test geometry

The blade selected for this analysis was extracted from the NSGA-II multi-objective optimization run at the C2 operating point ($\alpha_1 = 50°$, $\alpha_2 = -50°$, $M_2 = 1.2$), described in detail in Chapter 5. Specifically, the geometry chosen is the blade minimizing the kinetic energy loss coefficient ξ across the entire optimization run, obtained after approximately 10,000 CFD evaluations with MUCON $= -1$. Figure 3.19 shows the blade profile, and Figure 3.20 provides a detail of the structured mesh generated with the WAND meshing code.

The defining characteristic of this geometry, from a numerical standpoint, is that it converges robustly under MUCON $= -1$, but fails to reach convergence with MUCON $= +1$, regardless of any adjustment to the remaining MISES grid parameters. This is precisely the class of geometries whose physical validity required external verification.

Blade loading comparison

The steady RANS simulation with the k-ω SST turbulence model was carried out using the `LEO` solver at the C2 operating conditions. Figure 3.21 compares the blade surface isentropic Mach number distributions predicted by `MISES` (`MUCON` $= -1$) and `ADS CFD`.

The agreement is encouraging. `MISES` does not resolve the full complexity of the reflected shock structures within the blade passage, but demonstrates a faithful prediction of the overall blade loading. In particular, the position of the normal shock on the suction side, located at approximately 15% of the axial chord, is accurately captured. The notably steep gradient visible in the `MISES` solution in that region is a direct consequence of the first-order dissipation scheme, which smears the shock profile more aggressively than second-order upwinding would. It should be noted that discrepancies near the trailing edge are expected, as `MISES` models an open trailing edge geometry, whereas `ADS CFD` resolves the physical trailing edge directly.

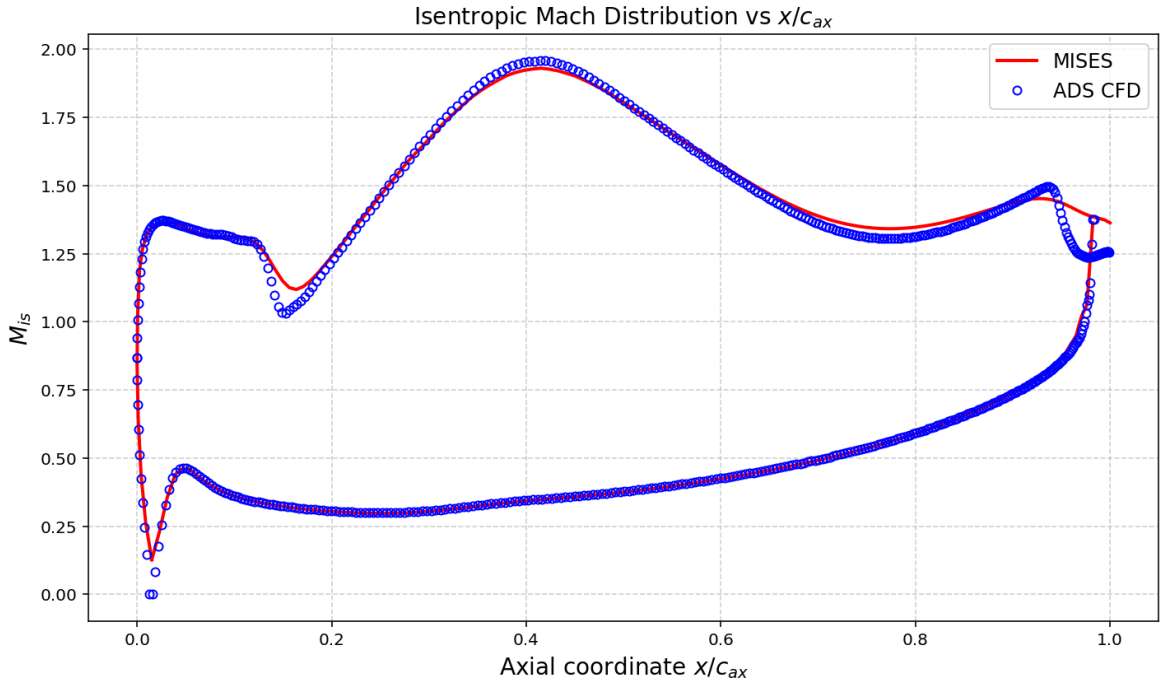


Figure 3.21: Comparison of blade surface isentropic Mach number distributions between `MISES` (`MUCON` $= -1$) and `ADS CFD` (RANS, k-ω SST)

Mach number contours

Figures 3.22 and 3.23 present the Mach number contour maps from both solvers. The `ADS CFD` solution reveals the full shock reflection pattern within the passage, which `MISES` cannot reproduce. Nevertheless, the global flow topology is qualitatively consistent between the two solutions.

The most important conclusion of this analysis concerns not the accuracy of `MISES`

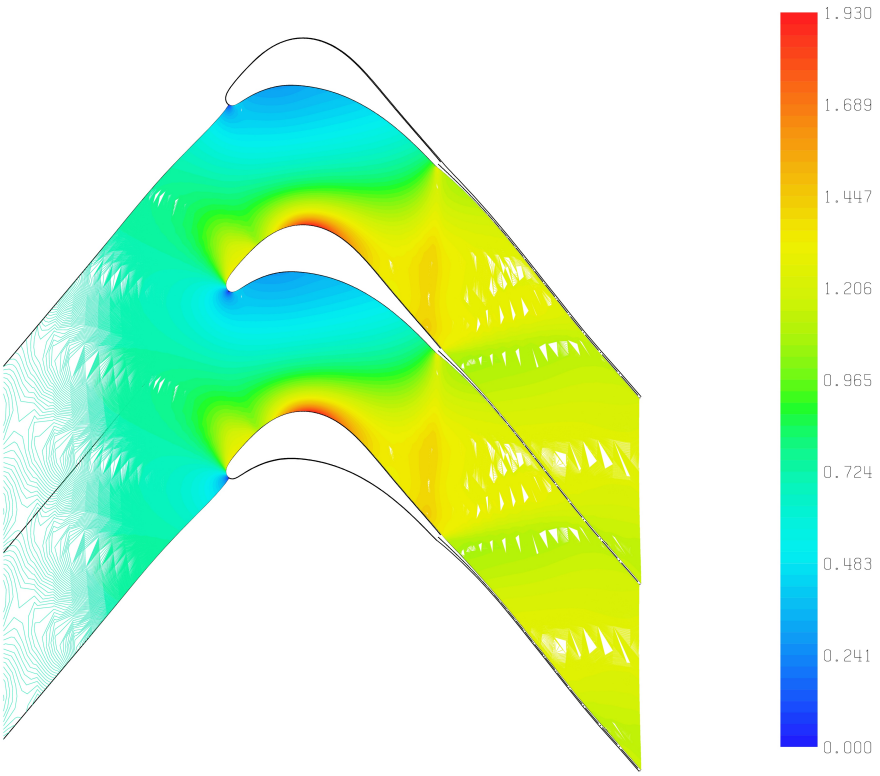


Figure 3.22: Mach number contours from `MISES` (`MUCON` $= -1$)

per se, but the physical legitimacy of the solutions obtained with `MUCON` $= -1$. The `ADS CFD` simulation confirms that the flow field at the C2 operating point is physically realizable: the blade performs as expected, with no spurious separation or unphysical behavior. This demonstrates that the failure of `MUCON` $= +1$ to converge is a numerical artifact of the second-order dissipation scheme on the `F-F` periodic grid—not a symptom of an inherently non-physical geometry. First-order dissipation does not manufacture convergence by masking real flow phenomena; it simply provides a more robust numerical path to the same physical solution.

For completeness, the exact `gridpar.xxx` and `ises.xxx` input files adopted for all database simulations are reported below.

```
F F
   30    30
   20
    1.500000
    0.500000
   150     1.000000     1.000000     0.010000
    1.000000     1.000000     0.000000     1.000000     1.000000
    0.000000
```

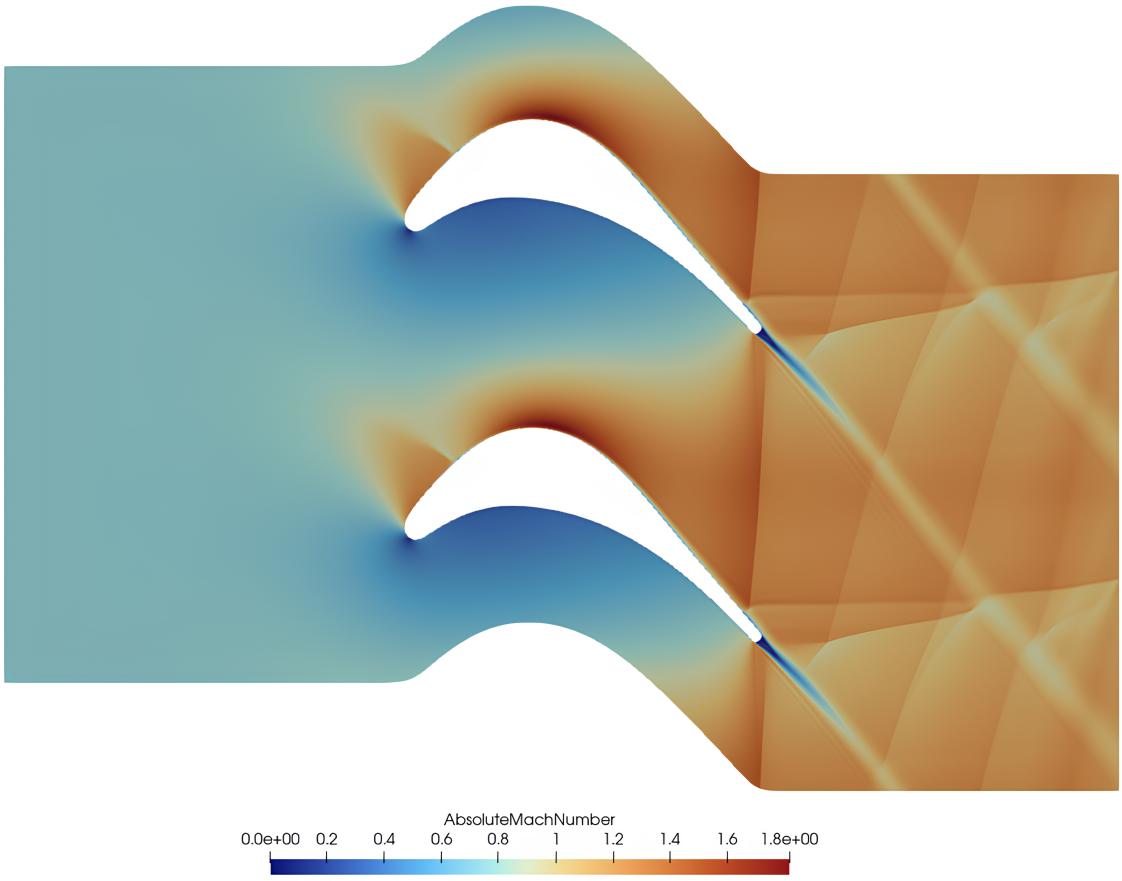


Figure 3.23: Mach number contours from `ADS CFD` (RANS, k-ω SST)

Listing 3.1: gridpar.xxx file adopted for all database simulations.

```
1   1              2                5                15
    6
2   1              3                4                17
    6
3   0.30           0.          sin(Alpha_1)       -0.250
4   M2             0.              0.              1.25
5   0.00           0.00        ! | MFRin     | HWRAT
6   Re_in         -3.50        ! | REYNOLDS  | NCRIT
7   0.030          0.030       ! | XTRANS1   | XTRANS2
8   4              0.95          -1.00     ! | ISMOM | MCRIT | MUCON
```

Listing 3.2: ises.xxx file adopted for all database simulations.

Chapter 4

Design of Experiments

The development of a **correlation-aware Design of Experiments (DoE)** strategy is essential to generate **geometrically feasible** and **aerodynamically realistic** turbine blade configurations. Unlike traditional space-filling methods that treat parameters as independent variables, this work recognizes that the **21 `ParaBlade` geometric parameters** exhibit **strong interdependencies** dictated by physical and aerodynamic constraints.

The fundamental challenge arises from **parameter non-independence**. A blade designed for high flow turning cannot be synthesized by arbitrarily combining individual parameter values drawn from independent distributions. Physically-consistent geometries require **coordinated adjustments** to camberline curvature, thickness distributions, leading and trailing edge radii, and metal angles. Random sampling of the 21-dimensional hypercube without accounting for these correlations inevitably produces **pathological geometries** that either fail to converge in CFD simulations or exhibit flow features incompatible with turbomachinery operation.

This chapter develops an intelligent DoE methodology based on three pillars:

- **Statistical foundation from reference database** - Extraction of parameter correlations from 30 physically-validated transonic turbine blade geometries.
- **Rank-preserving correlation enforcement** - Application of the Iman-Conover algorithm to impose realistic parameter dependencies while maintaining arbitrary marginal distributions.
- **Conditional sampling from multivariate Gaussian distributions** - A sampling approach that respects fixed design constraints (inlet/outlet angles) while propagating correlations to free parameters through conditional probability distributions.

The chapter is organized as follows. Section 4.1 describes the construction of the

30-blade reference database. Section 4.2 presents the mathematical framework for computing the global parameter correlation matrix. Section 4.3 discusses the choice of sampling distribution and introduces the conditional sampling approach based on multivariate Gaussian distributions. Section 4.4 introduces the Iman-Conover method for rank correlation preservation. Section 4.5 explains the empirical pitch selection model, and finally, Section 4.6 details the complete blade family generation algorithm including adaptive bounds and rejection sampling.

4.1 Reference Blade Database

To establish the correlation structure governing realistic transonic blade geometries, a reference database of **30 turbine cascades** was constructed. The database serves two essential purposes:

- **Statistical foundation** - Extracting parameter correlations to enable intelligent sampling strategies that preserve the physical relationships among design parameters, preventing the generation of infeasible geometries during design space exploration.
- **Design space coverage** - Providing anchor points across low, medium, and high turning configurations, ensuring adequate representation of the operating envelope.

The database comprises six well-documented real cascades selected from the open literature and 24 synthetic blades generated to augment coverage across the duty space.

4.1.1 Real Blade Geometries

Six transonic turbine cascades were selected from the open literature. The blades had to be **extensively documented** with published geometric coordinates and **experimentally validated** aerodynamic performance data. The selection prioritized cascades spanning both **subsonic** and **transonic** operating conditions with **maximum geometric diversity**—including stators and rotors, high-pressure and low-pressure turbine sections, and profiles with substantially different shapes.

The six cascades were organized into three turning regimes:

- **Low Turning**
  - *AACHEN Stator* - Low-pressure turbine stator section characterized in experimental campaigns at RWTH Aachen University.
  - *LS89* - High-pressure turbine nozzle guide vane from the von Kármán Institute, one of the most extensively validated transonic cascades in the literature.

- **Medium Turning**
  - *SPLEEN* - Mid-loaded transonic cascade from the von Kármán Institute, designed for operation at moderate deflection angles.
  - *LS59* - Moderately loaded high-pressure turbine profile from the von Kármán Institute well-documented.
- **High Turning**
  - *T106* - Low-pressure turbine profile with aggressive flow turning, widely used for transition and separation studies.
  - *HS1A* - Front-loaded high-pressure turbine cascade employed in the `MISES` validation study presented in Section 3.6.

Figure 4.1 shows the geometric profiles of the six cascades. Although these cascades belong to different turbine families, this is not a limitation. The database was constructed solely to **extract parameter correlations**—the fundamental relationships among the 21 `ParaBlade` parameters that govern **geometrically feasible** transonic blades, regardless of specific application.

4.1.2 Synthetic Blade Generation

The six real cascades provide anchor points but **insufficient sample size** for robust statistical correlation analysis. Computing a reliable Pearson correlation matrix for 21 parameters requires a sample substantially larger than the number of variables to ensure the matrix is **non-singular** and **adequately conditioned**.

To achieve this, **24 synthetic blades** were generated using the inverse design framework developed by Porta [6]. The Porta framework accepts prescribed **duty parameters** (inlet angle, outlet angle, exit Mach number, Reynolds number) as inputs and generates blade geometries through a **Gaussian Process surrogate model**.

The synthetic generation strategy allocated **eight blades** to each turning level (low, medium, high). Within each category, inlet and outlet flow angle pairs were distributed to ensure **uniform coverage** of the duty space. All synthetic blades were generated at $M_2 = 0.7$, consistent with the validated operating range of the framework.

Figure 4.2 shows the distribution of all 30 blades in the duty space defined by inlet angle θ_{in} and outlet angle θ_{out}. The real cascades anchor the extremes, while the synthetic blades fill intermediate regions. All 30 geometries were processed through the `MatchBlade` reverse-engineering procedure to extract consistent `ParaBlade` parametric representations.

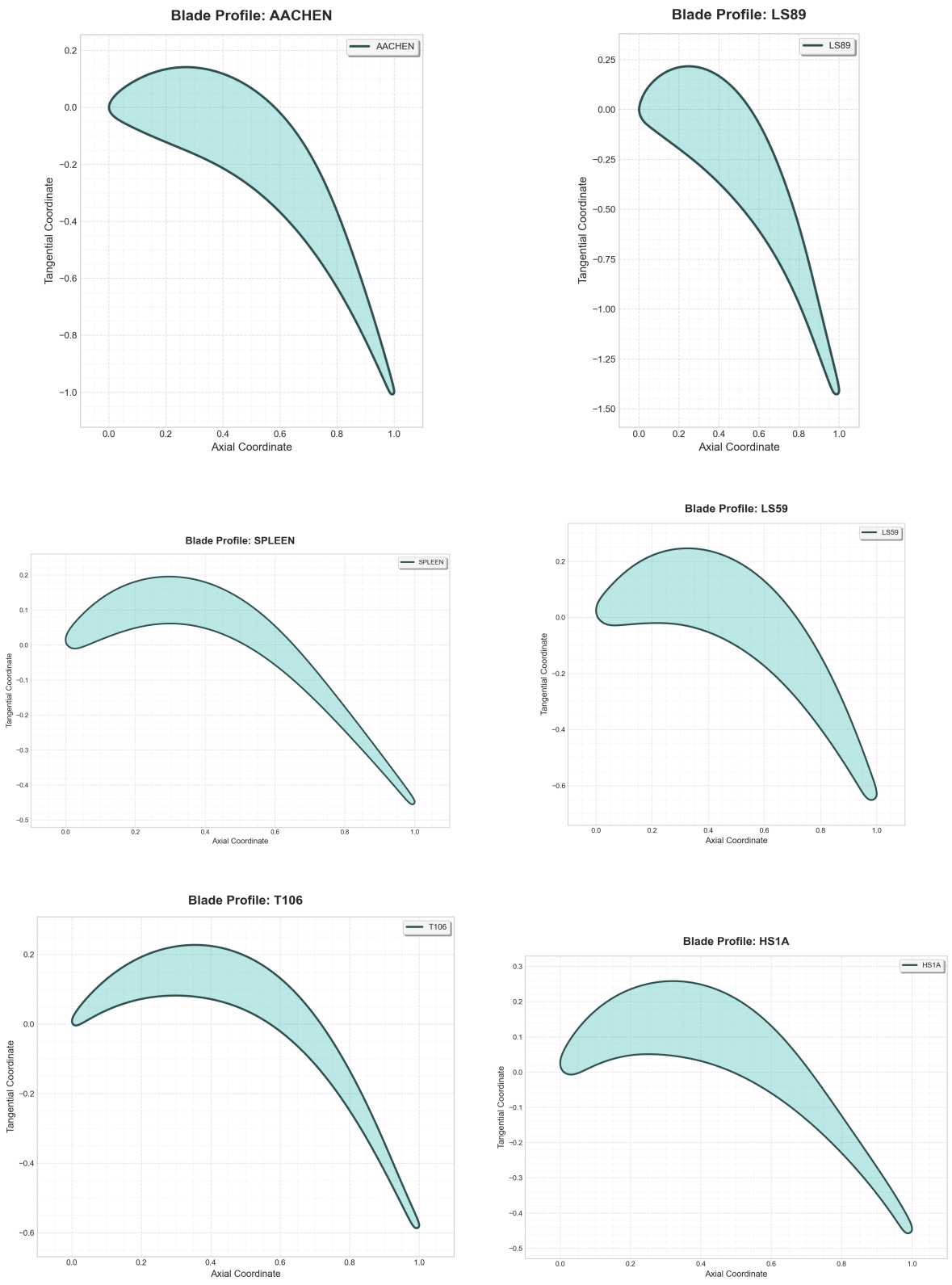


Figure 4.1: Geometric profiles of the six real transonic turbine cascades selected for the reference database. The blades are organized by turning regime: **Low Turning** (top row) - AACHEN stator and LS89; **Medium Turning** (middle row) - SPLEEN and LS59; **High Turning** (bottom row) - T106 and HS1A. All profiles are normalized by axial chord length to facilitate geometric comparison.

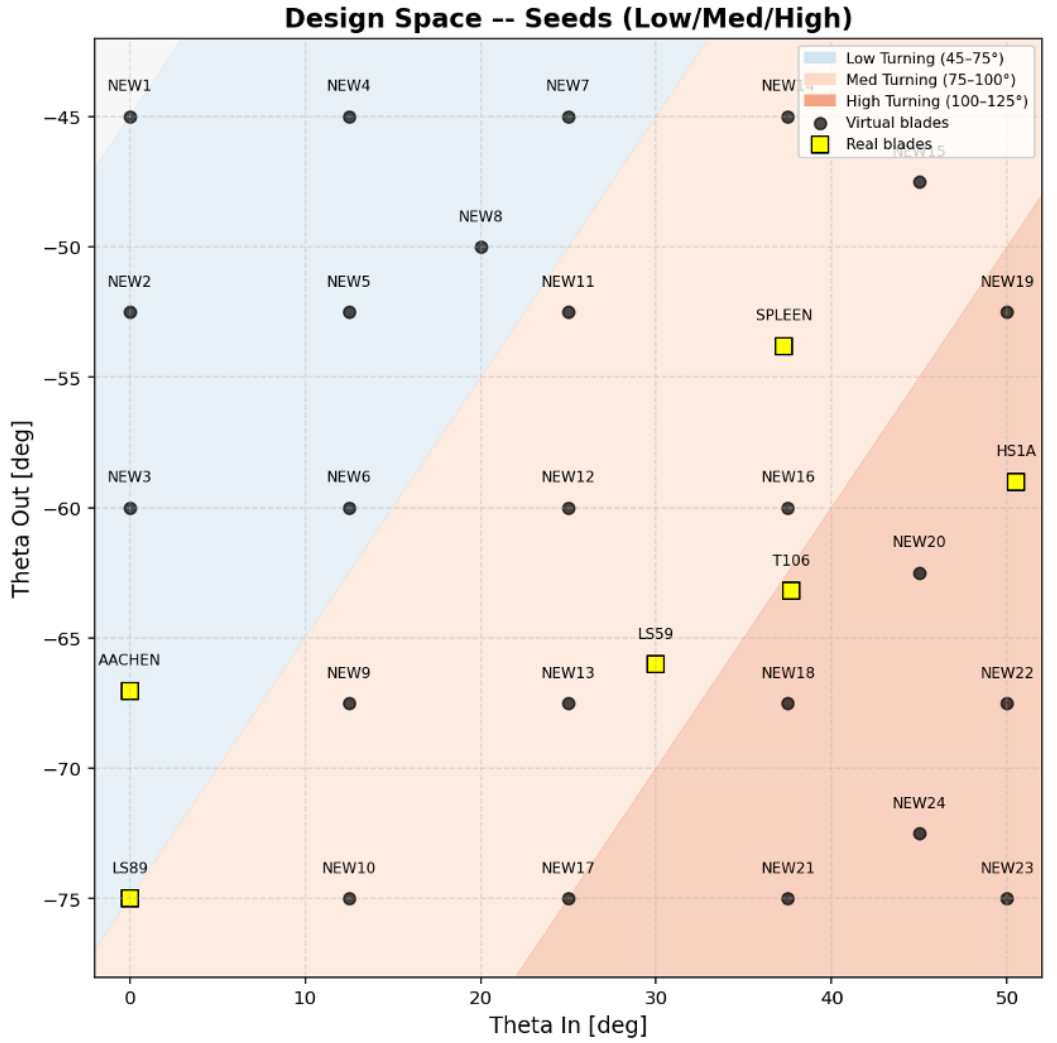


Figure 4.2: 30 Blades Reference Dataset.

4.2 Parameter Correlation Analysis

The extraction of parameter correlations from the 30-blade reference database is fundamental to the intelligent sampling strategy. This section presents the theoretical framework for correlation analysis, the computed global correlation matrix, and its statistical properties.

4.2.1 Correlation Theory

Pearson Product-Moment Correlation

The **Pearson correlation coefficient** quantifies the strength and direction of linear relationships between two variables. For parameters X and Y with n sample observations $(x_1, y_1), \ldots, (x_n, y_n)$, the sample Pearson correlation is:

$$r_{XY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4.1}$$

This can also be written as the normalized covariance:

$$r_{XY} = \frac{\mathrm{Cov}(X, Y)}{s_X s_Y} \tag{4.2}$$

The coefficient ranges from -1 (perfect negative linear correlation) to $+1$ (perfect positive linear correlation), with $r_{XY} = 0$ indicating no linear relationship. Pearson correlation measures linear dependence and works best when the data follows a bivariate normal distribution.

Spearman Rank Correlation

The Spearman rank correlation coefficient is computed by applying the Pearson formula to the ranks of the observations rather than their actual values. Let R_i and S_i denote the ranks of x_i and y_i. The Spearman correlation is:

$$\rho_{S,XY} = \frac{\sum_{i=1}^{n}(R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_{i=1}^{n}(R_i - \bar{R})^2}\sqrt{\sum_{i=1}^{n}(S_i - \bar{S})^2}} \tag{4.3}$$

When there are no tied ranks, this simplifies to:

$$\rho_{S,XY} = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \tag{4.4}$$

where $d_i = R_i - S_i$ is the rank difference. Spearman correlation has several advantages over Pearson:

- Works for any monotonic relationship, not just linear.
- Robust to outliers because it uses ranks instead of actual values.
- Makes no assumptions about the distribution of the data.
- Invariant to monotonic transformations of the variables.

Choice of Correlation Measure

For the 30-blade database, Spearman correlation was selected to extract the reference correlation structure. This choice is based on:

- **Small sample size** - With only 30 blades across 21 parameters, the ratio $n/p \approx 1.43$ is low. Individual outlier designs can disproportionately affect Pearson correlations. Spearman's rank-based approach is more robust with limited data.

- **Unknown marginal distributions** - Pearson correlation assumes bivariate normality. The actual distributions of blade parameters are unknown and may be skewed or multi-modal. Spearman makes no distributional assumptions.
- **Physical relationships** - Geometric relationships between blade parameters are often monotonic but not necessarily linear. Spearman captures these dependencies more accurately.
- **Consistency with Iman-Conover** - The Iman-Conover algorithm, which is the core of the sampling pipeline, operates on rank correlations and guarantees exact matching of a *Spearman* target. It is therefore natural to ensure that the most accurate estimate in the pipeline is precisely the Spearman correlation, rather than Pearson.

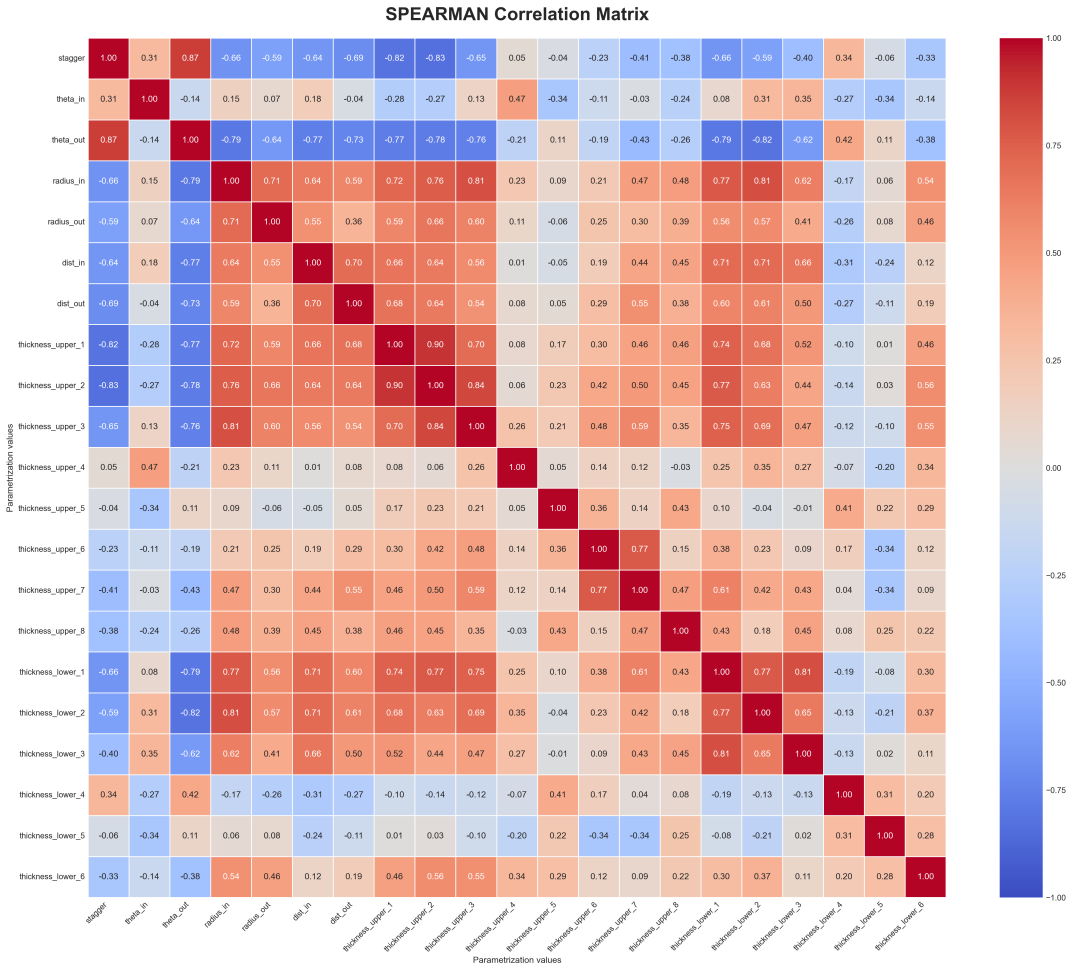


Figure 4.3: Spearman rank correlation matrix computed from the 30-blade reference database. The strong off-diagonal components highlight the severe interdependencies among the 21 `ParaBlade` geometric parameters.

4.3 Choice of Sampling Distribution

Generating a Design of Experiments for a high-dimensional blade parameter space requires choosing a sampling strategy that balances two competing objectives: covering the design space as uniformly as possible, to ensure that the surrogate model is trained on a representative dataset, and concentrating samples in regions that are physically plausible for the operating condition of interest. The most established approach in the DoE literature is **Latin Hypercube Sampling** (LHS) [25]. LHS divides each parameter dimension into n equal-probability intervals and draws exactly one sample per interval, ensuring that no two samples share the same interval in any dimension. Combined with the **maximin** criterion—which selects, among all valid LHS configurations, the one maximizing the minimum Euclidean distance between any two samples—LHS produces highly uniform space-filling designs that are widely considered optimal for surrogate model training [2].

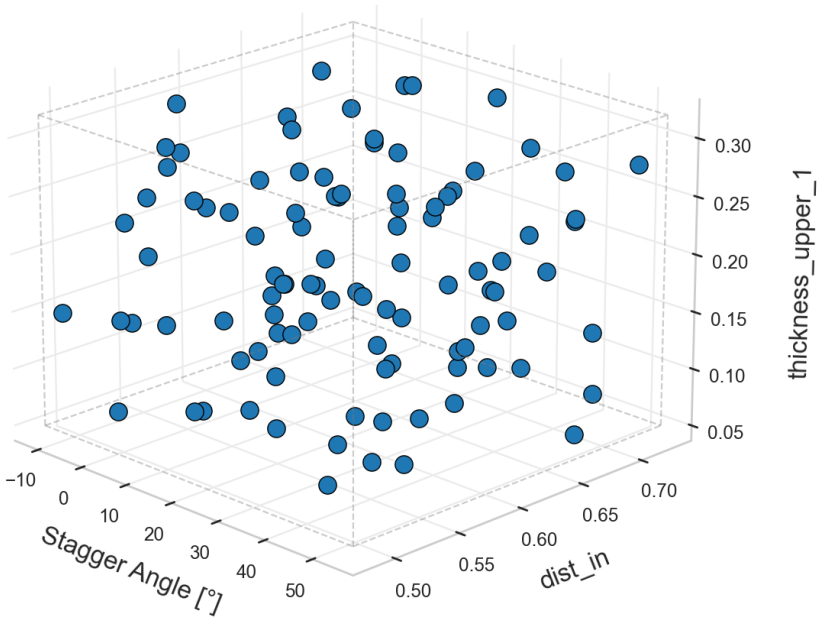


Figure 4.4: LHS *maximin* example.

LHS was evaluated as a candidate strategy and tested in combination with the Iman-Conover rank correlation enforcement described in Section 4.4. Notably, LHS combined with Iman-Conover produced rank correlations even closer to the target than the Gaussian approach ultimately adopted—precisely because it does not impose any copula hypothesis on the joint distribution of parameters.

Despite these favourable properties, LHS was discarded. The issue is not its correlation-matching capability, but its behaviour across the duty space. LHS fills the parameter hypercube uniformly by construction—and this is exactly its strength when the design space is homogeneous. In the present problem, however, the hypercube is not

homogeneous with respect to the operating condition: a parameter combination that produces a physically consistent blade at a moderate turning angle ($\alpha_2 \approx -60°$) may be entirely implausible at the extremes of the design space ($\alpha_2 \approx -45°$ or $\alpha_2 \approx -75°$), where the camberline geometry, metal angles, and thickness distribution must satisfy substantially different aerodynamic constraints.

One could apply a statistical rejection filter to discard samples that are geometrically incompatible with the target duty—but this would progressively deplete the uniform coverage that justifies using LHS in the first place. At the extremes of the duty space, the fraction of rejected samples becomes large enough to undermine the space-filling properties of the method entirely.

This observation motivates the adoption of a **conditional multivariate Gaussian sampling** strategy, described in the following sections. Rather than filling the hypercube uniformly and filtering out implausible regions, the Gaussian sampler anchors its mean to the local neighborhood of the target duty point in the blade database and concentrates samples naturally in the geometrically feasible region—without requiring any post-hoc rejection of incompatible configurations.

4.3.1 Multivariate Gaussian Conditional Sampling

Conditional multivariate Gaussian sampling is a **probabilistic approach** in which new parameter configurations are drawn from a multivariate Gaussian distribution that has been conditioned on a set of **prescribed values**—in this case, the duty parameters defining the target operating condition. Rather than sampling freely from the full joint distribution, the sampler restricts itself to the subspace consistent with the fixed constraints, naturally concentrating the generated geometries around **physically plausible configurations** for the given duty [26].

A multivariate Gaussian distribution over a vector of p continuous random variables $\mathbf{x} \in \mathbb{R}^p$ is fully characterised by its mean vector $\boldsymbol{\mu} \in \mathbb{R}^p$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$:

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{4.5}$$

For the blade parameter space, $\mathbf{x}$ is the vector of the 21 `ParaBlade` geometric parameters, $\boldsymbol{\mu}$ is estimated from the reference database, and $\boldsymbol{\Sigma}$ is constructed from the **Pearson correlation matrix** derived in Section 4.2.

The choice of the multivariate Gaussian is motivated by a property that no other common distribution possesses in closed form: **the conditional distribution of a subset of Gaussian variables, given the values of the remaining ones, is itself a Gaussian with analytically known mean and covariance.**

Formally, partition the parameter vector into two subsets—the **fixed** parameters

$\mathbf{x}_A$, whose values are prescribed by the target duty, and the **free** parameters $\mathbf{x}_O$, to be sampled:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_A \\ \mathbf{x}_O \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_O \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AO} \\ \boldsymbol{\Sigma}_{OA} & \boldsymbol{\Sigma}_{OO} \end{pmatrix}. \tag{4.6}$$

The conditional distribution of $\mathbf{x}_O$ given $\mathbf{x}_A = \mathbf{a}$ is then:

$$\mathbf{x}_O \mid \mathbf{x}_A = \mathbf{a} \;\sim\; \mathcal{N}\big(\boldsymbol{\mu}_{O|A},\, \boldsymbol{\Sigma}_{O|A}\big)\,, \tag{4.7}$$

where the conditional mean and covariance are given by:

$$\boldsymbol{\mu}_{O|A} = \boldsymbol{\mu}_O + \boldsymbol{\Sigma}_{OA}\, \boldsymbol{\Sigma}_{AA}^{-1}\, (\mathbf{a} - \boldsymbol{\mu}_A)\,, \tag{4.8}$$

$$\boldsymbol{\Sigma}_{O|A} = \boldsymbol{\Sigma}_{OO} - \boldsymbol{\Sigma}_{OA}\, \boldsymbol{\Sigma}_{AA}^{-1}\, \boldsymbol{\Sigma}_{AO}. \tag{4.9}$$

The interpretation of these equations is physically intuitive. The conditional mean $\boldsymbol{\mu}_{O|A}$ shifts away from the global database mean proportionally to how far the fixed parameters deviate from their average values and how strongly they are correlated with the free ones. The conditional covariance $\boldsymbol{\Sigma}_{O|A}$ is always smaller than the marginal covariance $\boldsymbol{\Sigma}_{OO}$: fixing parameters reduces the uncertainty on the remaining ones, with the reduction being proportional to the strength of the correlations. A strongly correlated pair of parameters—such as the stagger angle and the outlet metal angle—will see the variance of one collapse significantly when the other is fixed.

This propagation mechanism is precisely what makes conditional Gaussian sampling suitable for the present problem: fixing the inlet and outlet flow angles (α_1, α_2) as duty constraints automatically reshapes the entire distribution over the remaining geometric parameters, concentrating samples in the region of the design space that is consistent with the prescribed operating condition.

Figure 4.5 illustrates the difference between the two approaches for the parameter `dist_in`—which controls the axial position of the camberline curvature and has a strong influence on the blade loading distribution—at a fixed duty of $\alpha_1 = 50°$, $\alpha_2 = -65°$. The behaviour described above is clearly visible: the Gaussian conditional sampler concentrates samples in the statistically most probable region for the prescribed angles, naturally avoiding parameter combinations that would be geometrically inconsistent with that duty. The LHS distribution, by contrast, spreads uniformly across the full parameter range regardless of the operating condition.

It is worth noting that the Gaussian sampler was not constrained to the strict bounds of the 30-blade reference dataset. By trusting the mean and variance estimated from the database rather than imposing hard limits, the sampler is free to explore beyond the observed range when the conditional distribution suggests it—and in practice, all generated geometries remained physically feasible and geometrically diverse.

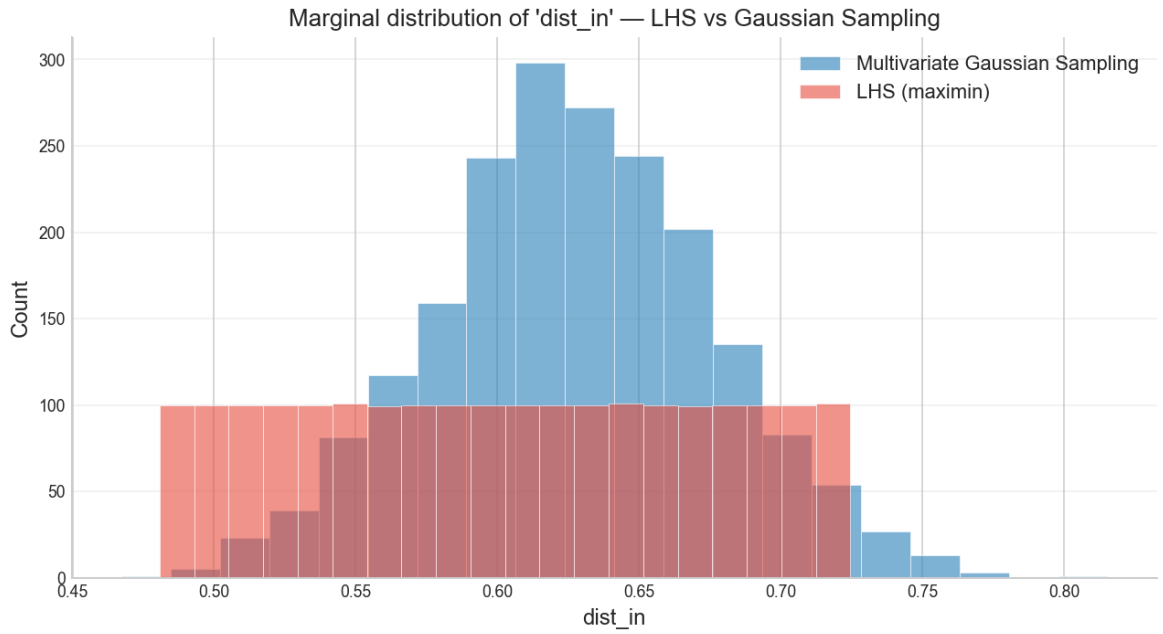


Figure 4.5: Comparison of marginal distributions between LHS (*maximin* criteria) and a Multivariate Gaussian Distribution ($\alpha_1 = 50°, \alpha_2 = -65°$).

4.3.2 Assumptions of the Gaussian Model

Gaussian marginals

The first assumption is that each of the 21 `ParaBlade` parameters follows, marginally, a Gaussian distribution. In practice, geometric parameters such as leading edge radius or thickness values are bounded and may exhibit skewed distributions. This assumption is therefore an approximation, accepted here in exchange for the analytical tractability that the Gaussian framework provides.

Gaussian copula

The second assumption concerns the *dependence structure* among the 21 parameters, how they relate to each other jointly. To model this dependence, the concept of a **copula** is introduced. A copula is a function that separates the marginal distributions of a set of random variables from their joint dependence structure. Choosing a **Gaussian copula** means assuming that, once the marginal information is removed from each parameter, the residual dependence structure is that of a multivariate normal distribution.

This assumption has a precise and testable consequence. Under a Gaussian copula, the Spearman rank correlation ρ_S and the Pearson correlation ρ_P of any parameter pair are related by the exact analytical formula derived by Kruskal [27]:

$$\rho_P = 2\sin\left(\frac{\pi\,\rho_S}{6}\right), \qquad \rho_S = \frac{6}{\pi}\arcsin\left(\frac{\rho_P}{2}\right). \tag{4.10}$$

Figure 4.6 shows the element-wise absolute difference between the Pearson matrix measured directly from the 30-blade dataset and the one derived from the Spearman matrix via Equation (4.10). If the Gaussian copula held exactly, this difference would be zero everywhere. The maximum observed deviation is $|\Delta\rho|_{\max} = 0.23$, confirming that certain parameter pairs exhibit monotonic but non-linear dependencies that the model cannot fully capture.

Despite this limitation, the approximation proves acceptable in practice. The conditioning step naturally restricts the generated samples to the physically plausible region of the parameter space for the prescribed duty, while the Iman-Conover algorithm enforces the target rank correlation structure on the final samples. The combined effect is that the generated blade families remain well-correlated and geometrically feasible throughout the design space, even if the correlation structure is not perfectly recovered. The quantitative assessment of this behavior is presented in Section 4.6.3.

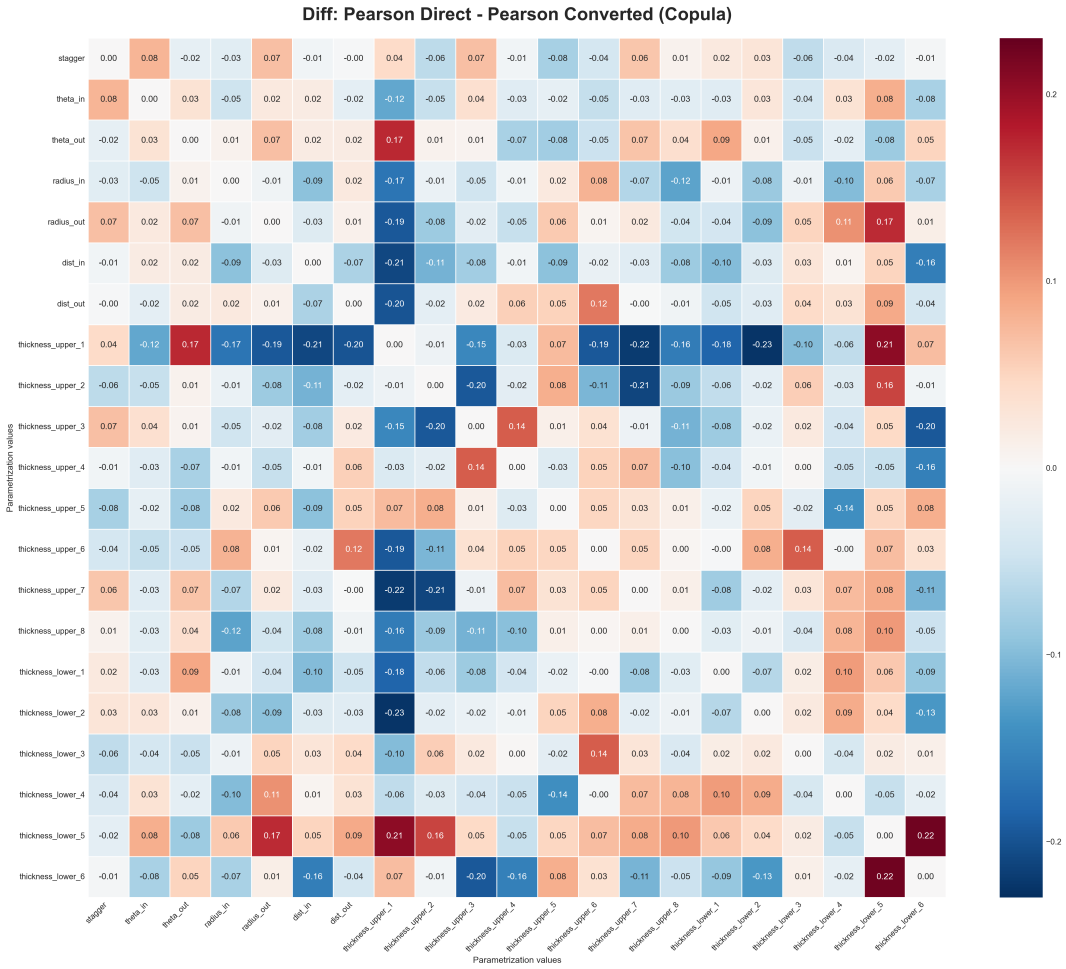


Figure 4.6: Element-wise absolute difference matrix between the direct Pearson correlations measured from the 30-blade dataset and the Pearson correlations derived from Spearman via the Kruskal formula.

4.4 The Iman-Conover Method

The conditional Gaussian sampling described in the previous section produces blade parameter sets that respect the prescribed duty constraints and are concentrated in the physically plausible region of the design space. However, as discussed in Section 4.3.2, the Gaussian copula hypothesis introduces an approximation in the correlation structure of the generated samples, and the final rank correlation of the output may deviate from the target.

A dedicated correction step is therefore needed to enforce the target rank correlation on the final sample set. This is achieved through the **Iman-Conover algorithm** [13], a distribution-free method that reorders the columns of a sample matrix to match a target Spearman rank correlation, without altering the marginal distribution of any parameter. The key property that makes it attractive here is precisely its distribution-free nature: unlike approaches that assume a specific joint distribution, Iman-Conover operates purely on ranks and can therefore be applied to any sampling strategy, regardless of the shape of the marginals.

Algorithm

The algorithm proceeds as follows. Given a matrix of generated samples $\mathbf{X} \in \mathbb{R}^{n \times p}$ and a target Spearman correlation matrix $\mathbf{R}_S$:

1. **Convert target Spearman to Pearson.** The target matrix $\mathbf{R}_S$ is converted to a Pearson matrix via Equation (4.10), obtaining $\mathbf{R}_P$. Its Cholesky factor $\mathbf{L}_T$ is computed as $\mathbf{R}_P = \mathbf{L}_T \mathbf{L}_T^T$.

2. **Compute ranks of input samples.** Each column of $\mathbf{X}$ is converted to integer ranks $1, \ldots, n$.

3. **Compute current rank correlation.** The standardised rank matrix $\mathbf{Z}_r$ is computed and the current rank correlation matrix $\tilde{\mathbf{C}}$ is obtained as:

$$\tilde{\mathbf{C}} = \frac{\mathbf{Z}_r^T \mathbf{Z}_r}{n-1}. \tag{4.11}$$

   Its Cholesky factor $\mathbf{Q}$ is computed as $\tilde{\mathbf{C}} = \mathbf{Q}\mathbf{Q}^T$.

4. **Compute transformation matrix.** The linear map that transforms the current rank structure into the target one is:

$$\mathbf{S} = \mathbf{L}_T \mathbf{Q}^{-1}. \tag{4.12}$$

5. **Apply transformation.** The input samples are mapped to the standard normal space via the quantile function Φ^{-1}, and the transformation is applied:

$$\mathbf{R} = \Phi^{-1}(\mathbf{X})\,\mathbf{S}^T. \tag{4.13}$$

6. **Reorder original samples.** Each column of $\mathbf{X}$ is reordered to match the ranking induced by the corresponding column of $\mathbf{R}$, preserving the original marginal distribution exactly.

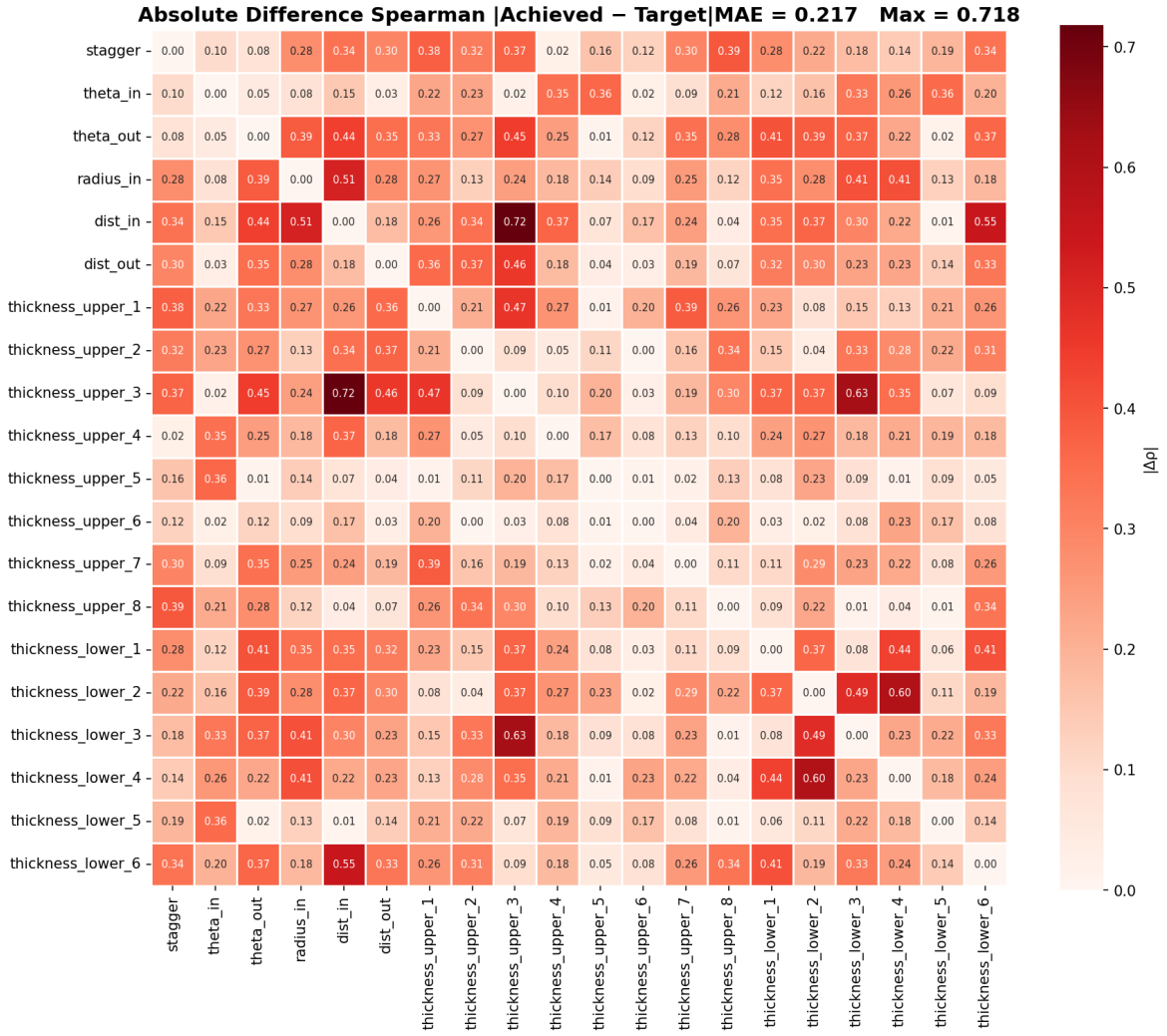


Figure 4.7: Element-wise absolute difference between the database Spearman correlation matrix and the one achieved after conditional Gaussian sampling alone, before Iman-Conover correction.

Implementation of the correlation enforcement

The effectiveness of the Iman-Conover correction is assessed by comparing the Spearman correlation matrix of the database with that of the generated samples at two stages of

the pipeline. Figure 4.7 shows the element-wise absolute difference between the database Spearman matrix and the one obtained after the conditional Gaussian sampling alone, before Iman-Conover is applied. The maximum deviation is $|\Delta\rho|_{\max} = 0.718$, confirming that the conditioning step already introduces a degree of correlation structure, but the approximation error remains non-negligible.

Figure 4.8 shows the same comparison after the Iman-Conover correction is applied. The maximum deviation reduces to $|\Delta\rho|_{\max} = 0.317$, with both the MAE and the maximum absolute error reduced by more than half with respect to the pre-correction values, demonstrating that the algorithm effectively brings the rank correlation structure of the generated samples into close agreement with the database target.

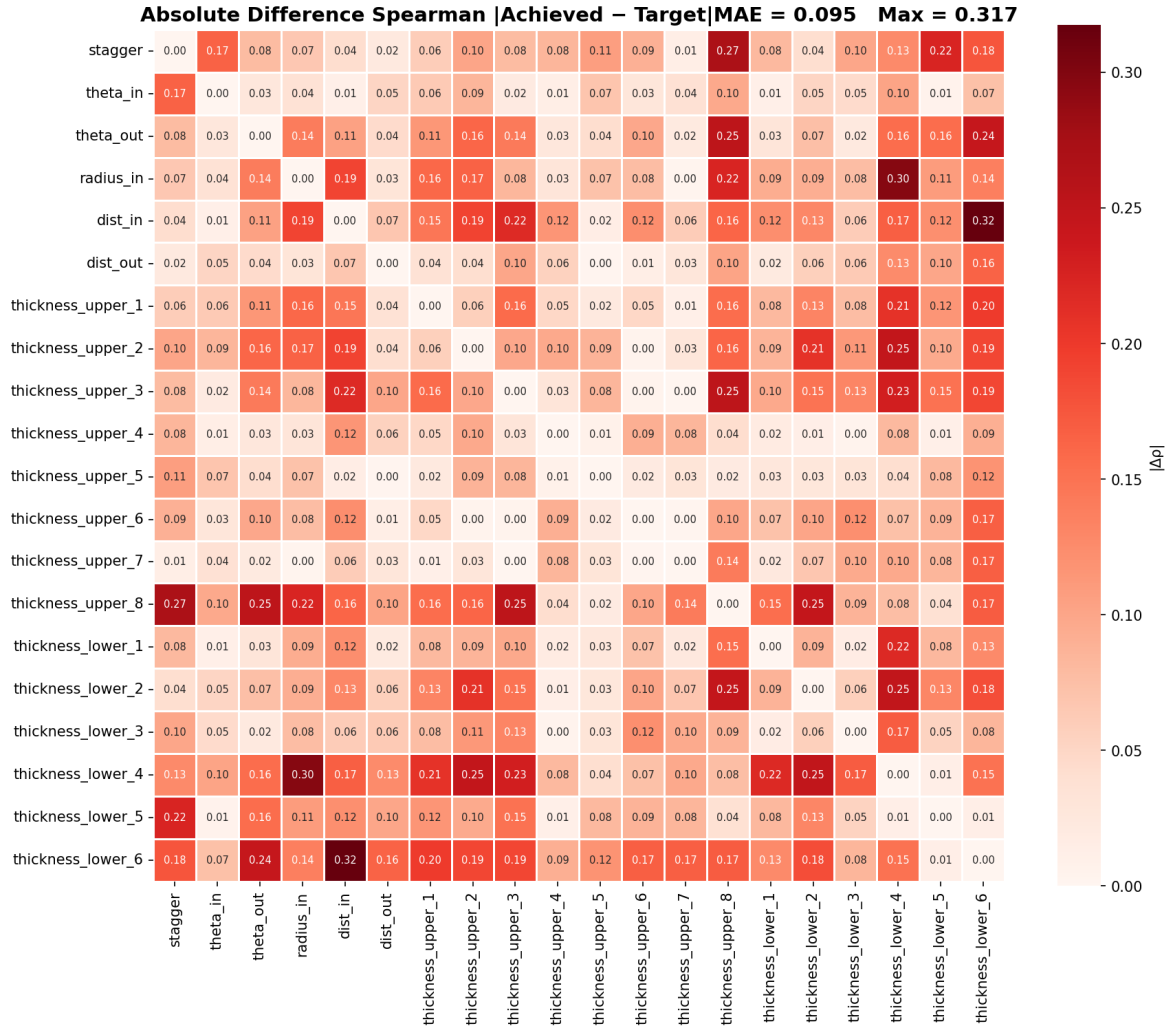


Figure 4.8: Element-wise absolute difference between the database Spearman correlation matrix and the one achieved after the full pipeline, including Iman-Conover correction.

4.5 Pitch-to-Axial-Chord Ratio Determination

The pipeline described in the previous sections produces a set of blade geometries defined by the free parameters of $\mathbf{x}_O$, with their correlation structure preserved by the Iman-Conover correction. However, one fundamental design variable has not yet been addressed: the **pitch-to-axial-chord ratio** t/c_{ax}.

Unlike the geometric parameters of $\mathbf{x}_O$, t/c_{ax} governs the aerodynamic loading of the cascade, influences the outlet flow angle through the throat-to-pitch ratio, and in the transonic regime determines whether a `MISES` simulation converges at all. For this reason, it is treated as a separate variable, decoupled from $\mathbf{x}_O$ and assigned independently through a uniform distribution over a prescribed range. NSGA-II will learn autonomously which pitch values are aerodynamically compatible with a given blade shape and duty.

Nevertheless, Deb [28] argues that embedding problem-specific knowledge into the initial bounds accelerates convergence and improves exploration quality—a poorly chosen pitch range results in a large fraction of infeasible individuals in Generation 0, wasting evaluation budget before the algorithm can self-correct. The goal of this section is therefore to determine a physically meaningful and duty-aware pitch range.

4.5.1 Limitations of Analytical Loading Criteria

A standard practice in turbomachinery design is to define the solidity implicitly through the **Zweifel lift coefficient** Z_w, a dimensionless parameter representing the ratio of the actual tangential force on the blade to an idealised reference force. Formally:

$$Z_w = \frac{\oint p \, \mathrm{d}x}{c_{ax}(P_{01} - P_2)}, \tag{4.14}$$

where c_{ax} is the axial chord. Since the detailed pressure distribution in Equation (4.14) is unavailable during preliminary design, control-volume analysis yields the incompressible algebraic approximation [**<empty citation>**]:

$$Z_{w,\mathrm{inc}} = \frac{2t}{c_{ax}} \sin^2\alpha_2 \left(\cot\alpha_1 + \cot\alpha_2\right). \tag{4.15}$$

Inverting Equation (4.15) for a target loading range (e.g., $Z_w \in [0.8, 1.0]$) provides a pitch estimate that scales automatically with the flow angles, and constituted the initial strategy considered in this work. However, in the transonic regime characterising the present design space, the incompressible formulation consistently overestimates the optimal pitch, producing blade rows with insufficient solidity that fail to turn the flow.

A compressibility-corrected formulation proposed by Ni et al. [8] was therefore

evaluated:

$$Z_{w,\text{comp}} = \frac{2t}{c_{ax}} \sin\alpha_2 \frac{M(M_1)\cos\alpha_1 + M(M_2)\cos\alpha_2}{M(M_2)}, \tag{4.16}$$

where $M(M)$ is the isentropic velocity function. While physically more rigorous, this corrected formulation exhibited the opposite limitation, systematically underestimating the pitch at high-turning and high-Mach conditions and producing unnecessarily dense cascades.

The extensive exploration of the design space confirmed that no fixed range of Zweifel coefficients—incompressible or corrected—yields a physically feasible pitch across all combinations of α_1, α_2, and M_2. The indirect determination of solidity via analytical loading criteria was therefore abandoned in favour of a fully data-driven approach, described in the following section.

4.5.2 Empirical Pitch Selection Model

The limitations of the analytical Zweifel-based approach discussed in Section 4.5.1 motivate the adoption of a fully data-driven strategy. Rather than deriving the pitch indirectly from a loading coefficient, the feasibility boundary of the cascade is mapped empirically, establishing a direct regression between the aerodynamic boundary conditions and the geometric stability limit:

$$\left(\frac{t}{c_{ax}}\right)_{\max} = f(\alpha_1,\, \alpha_2,\, M_2)\,. \tag{4.17}$$

This formulation bypasses the intermediate proxy of a loading coefficient, relying directly on the evidence collected by the optimiser to define the feasible design space boundaries.

Exploratory Optimisation Campaign

To map the stability boundaries of the cascade across the full duty space, a dedicated exploratory campaign of **41** independent NSGA-II optimisations was conducted prior to the main design of experiment. The duty points sampled follow the same grid described in Chapter 2:

- **Main design grid** ($M_2 = 1.2$): a 5×5 grid spanning $\alpha_1 \in [0°, 50°]$ and $\alpha_2 \in [-50°, -75°]$, for 25 runs at the target Mach number.
- **Compressibility sweep** ($M_2 \in \{0.8, 0.9, 1.0, 1.1\}$): the four corner points of the α_1-α_2 plane at each additional Mach number, for $4 \times 4 = 16$ further runs.

Each optimisation was run with 10,000 CFD evaluations and with the pitch bounded only by wide physical limits $t/c_{ax} \in [0.4,\, 2.5]$, allowing the optimiser to identify the maximum feasible pitch without any a priori bias.

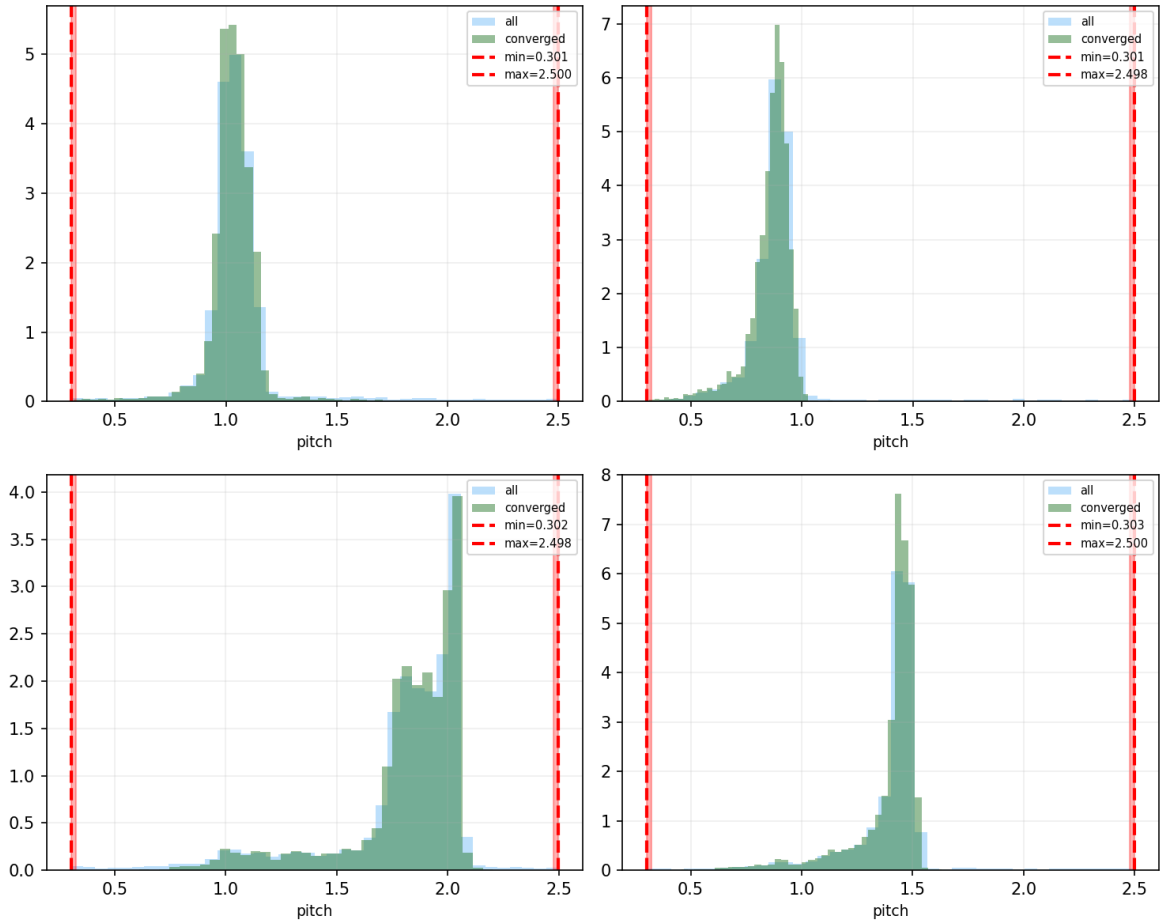


Figure 4.9: Pitch analysis of C1, C2, C3 and C4 corner points with $M_2 = 1.2$ and extended pitch bounds.

Quantile Regression Model

The 41 stability limits extracted by the filtering procedure were used to train a regression model. A **quantile regression** at the 98th percentile ($q = 0.98$) with a second-degree polynomial feature basis was selected. Unlike standard least-squares regression—which minimises the mean squared error and therefore estimates the conditional mean of the response—quantile regression minimises a weighted asymmetric loss function, known as the *pinball loss*:

$$\mathcal{L}_q(\epsilon) = \begin{cases} q\,|\epsilon| & \text{if } \epsilon \geq 0, \\ (1-q)\,|\epsilon| & \text{if } \epsilon < 0, \end{cases} \tag{4.18}$$

where $\epsilon = y - \hat{y}$ is the residual. For $q = 0.98$, underprediction errors are penalised 49 times more heavily than overprediction errors, so the fitted surface naturally hugs the upper envelope of the data rather than its centre. The resulting model directly estimates the 98th conditional percentile of the stability limit, which is conceptually consistent with the filtering step: both target the boundary of the feasible domain.

Model Performance and Application to the Design of Experiment

The parity plot confirms the high accuracy of the regression across all duty points and Mach numbers. In a limited number of cases the model underpredicts the true stability limit by up to 10%, a direct consequence of the pinball loss formulation which naturally tends towards the upper envelope of the training data. The upper sampling limit is therefore expanded by a factor of 1.10 to compensate for this known conservatism.

The primary motivation behind this model is **computational efficiency**. Without a duty-aware pitch range, the optimiser would continuously waste CFD evaluations on aerodynamically infeasible configurations throughout all generations, severely degrading the quality of the Pareto front. The pitch-to-axial-chord ratio is therefore bounded as:

$$\frac{t}{c_{ax}} \in \left[0.60 \cdot \hat{P}_{98\%},\ \ 1.10 \cdot \hat{P}_{98\%}\right], \tag{4.19}$$

where $\hat{P}_{98\%} = \hat{f}_{q=0.98}(\alpha_1, \alpha_2, M_2)$.

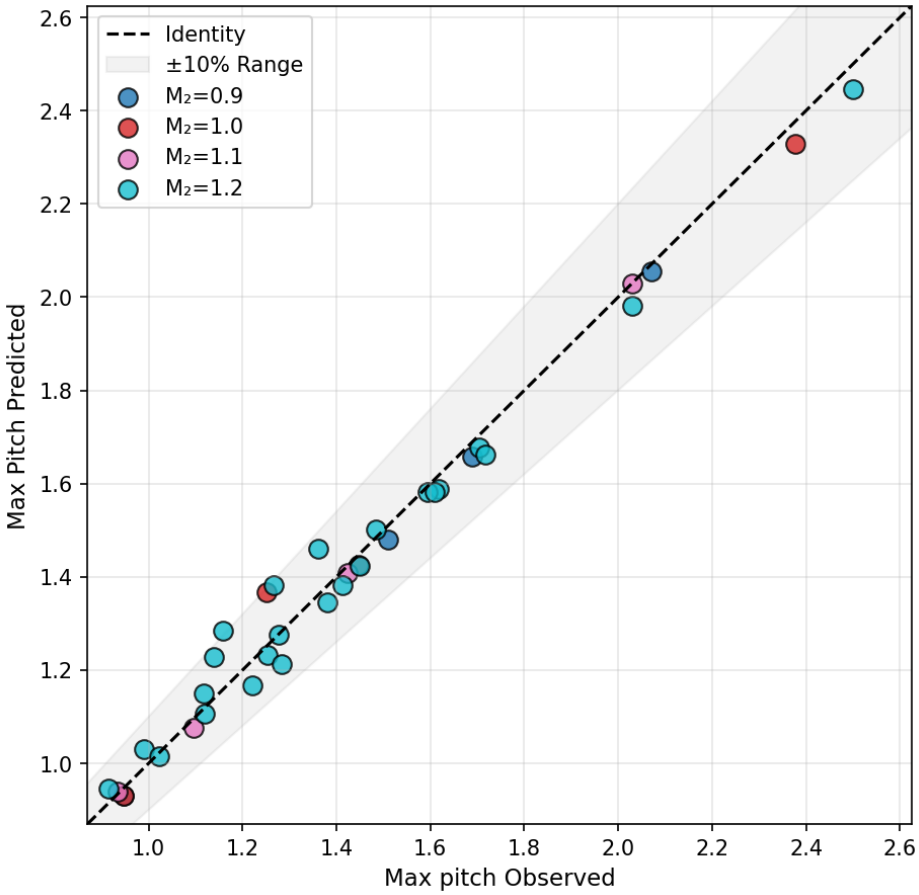


Figure 4.10: Parity plot of the pitch predictor model.

4.6 Blade Generation Algorithm

The theoretical components developed in the preceding sections are assembled into a single sequential pipeline. Given a target aerodynamic duty $(\alpha_1, \alpha_2, M_2)$, the algorithm generates a family of geometrically consistent blade sections defined by the 22 design variables of the parametrisation: the 21 geometric parameters of `ParaBlade` and the pitch-to-axial-chord ratio t/c_{ax}, which is treated separately for the reasons discussed in Section 4.5.2.

The resulting population constitutes Generation 0 of the NSGA-II optimisation. Figure 4.11 illustrates the complete workflow.

4.6.1 Parameter Partitioning and Boundary Conditions

Before sampling begins, the 21 `ParaBlade` geometric parameters are partitioned into two subsets according to their role in the prescribed duty. Two distinct constraint types are supported, summarised in Table 4.1.

Table 4.1: Parameter constraint types adopted in the blade generation pipeline.

Dictionary	**Effect**	**Typical use**
`FIXED_PARAMS_VALUES`	Variance set to zero; parameter clamped to a single value	Trailing edge radius
`FIXED_WITH_TOLERANCE`	Conditional mean anchored to the nominal value; uniform distribution enforced over the tolerance interval	Metal angles θ_{in}, θ_{out}

It is important to note that the angles prescribed in `FIXED_WITH_TOLERANCE` are the **metal angles** of the blade—θ_{in} at the leading edge and θ_{out} at the trailing edge—which in general differ from the actual inlet and outlet flow angles. At the inlet, the difference between the flow angle α_1 and the metal angle θ_{in} is the **incidence** i; at the outlet, the difference between θ_{out} and the actual exit flow angle α_2 is the **deviation** δ.

Both quantities depend on the blade geometry, the operating Mach number, and the loading condition, and neither can be determined a priori at this stage of the design process. For this reason, the metal angles are not fixed to the nominal flow direction but are instead allowed to vary within a **tolerance** of $\pm\mathbf{5}^\circ$ around it. This ensures that the generated family spans a range of camberline geometries wide enough to contain blades that, once evaluated by the CFD solver, produce the correct flow turning after accounting for the actual incidence and deviation. Without this tolerance, the family would be locked to a single camberline, and the probability of any individual blade achieving the target aerodynamic duty would be negligibly small.

A representative configuration for a high-turning duty point is:

```
FIXED_PARAMS_VALUES = {
    'radius_out': 0.015 # --- 3% of Trailing Edge Thickness ---
}

FIXED_WITH_TOLERANCE = {
    'theta_in':  {'value':  50, 'tolerance_lower': 5, 'tolerance_upper': 5},
    'theta_out': {'value': -75, 'tolerance_lower': 5, 'tolerance_upper': 5}
}
```

Listing 4.1: Example boundary condition configuration for a high-turning duty point ($\alpha_1 = 50°$, $\alpha_2 = -75°$).

The tighter tolerance on `theta_out` reflects the physical asymmetry of the design space near the high-turning boundary, where geometrically feasible configurations occupy a narrower angular range.

4.6.2 Pipeline Phases

Phase 1 — Data Loading and Correlation Estimation
The 30-blade reference database is loaded and the global Spearman rank correlation matrix $\mathbf{R}_S \in \mathbb{R}^{21\times21}$ is computed. The matrix serves as the statistical backbone of all subsequent phases.

Phase 2 — Adaptive Nearest-Neighbour Anchor
Rather than centering the Gaussian distribution on the global database mean—which would represent an average blade across all turning regimes—the algorithm constructs a duty-aware local anchor. The K nearest blades to the target in the $(\theta_{\text{in}}, \theta_{\text{out}})$ space are identified, and a locally weighted mean $\boldsymbol{\mu}_{\text{loc}}$ and covariance $\boldsymbol{\Sigma}_{\text{loc}}$ are computed. The neighbourhood size K is selected adaptively:

- **Central region** of the database ($\sigma < 0.5$): $K = 29$, every blade of the dataset influences $\boldsymbol{\mu}_{\text{loc}}$ and $\boldsymbol{\Sigma}_{\text{loc}}$.
- **Frontier region** ($\sigma > 1.8$): $K = 10$, to focus exclusively on the most geometrically similar blades and avoid contamination from distant design points.
- **Intermediate region**: K is linearly interpolated between the two extremes.

The neighbour weights follow a Gaussian kernel $w_i = \exp(-d_i^2/2\sigma_k^2)$, with the kernel width σ_k itself adapting to the local density of the database.

Phase 3 — Gaussian Conditional Sampling
The free parameter vector $\mathbf{x}_B$ is sampled from the conditional Gaussian distribution given the fixed parameters $\mathbf{x}_A$, as derived in Section 4.3.1:

$$\boldsymbol{\mu}_{B|A} = \boldsymbol{\mu}_B + \boldsymbol{\Sigma}_{BA}\,\boldsymbol{\Sigma}_{AA}^{-1}\Big(\mathbf{x}_A^{\text{target}} - \boldsymbol{\mu}_A\Big)\,. \tag{4.20}$$

The shift term propagates the duty constraint through the entire geometry: a prescribed outlet angle is automatically translated into statistically consistent adjustments of all correlated parameters. A total of 2.0× the target family size is generated at this stage to compensate for the rejections that follow.

Phase 4 — Rejection Sampling
As discussed in Section 4.3.1, the philosophy of the sampling strategy is to **trust the statistics**: the conditional Gaussian distribution is left free to explore the parameter space without clipping samples to the bounds of the database. The only exception is a minimum threshold of 0.01 imposed on all `ParaBlade` thickness control points, both on the upper and lower surface, to avoid generating blades with excessively thin sections that would produce geometrically invalid profiles. Rather than clipping the offending values to 0.01, which would distort the marginal distributions and corrupt the rank correlation structure built in Phase 3, any sample violating the constraint is discarded entirely and a new candidate is drawn from the Gaussian. The 2.0× oversampling factor introduced in Phase 3 compensates for the discarded candidates.

Phase 5 — Iman-Conover Rank Correction
The accepted samples are passed to the Iman-Conover algorithm (Section 4.4) to enforce the target conditional Spearman matrix $\mathbf{R}_{S,B|A}$. After the correction:

- Parameters in `FIXED_WITH_TOLERANCE` are mapped to **uniform** marginal distributions over their prescribed tolerance interval via Inverse Transform Sampling, ensuring flat coverage of the full angle range.
- All remaining free parameters retain their **Gaussian** marginals as shaped by Phase 3.
- The Spearman correlation structure of the family matches $\mathbf{R}_{S,B|A}$ by construction.

Phase 6 — Pitch Assignment
The pitch-to-axial-chord ratio t/c_{ax} is not part of the statistical sampling. As discussed in Section 4.5.2, it is assigned independently and paired randomly with the blade geometries, leaving the optimiser to identify compatible combinations autonomously.

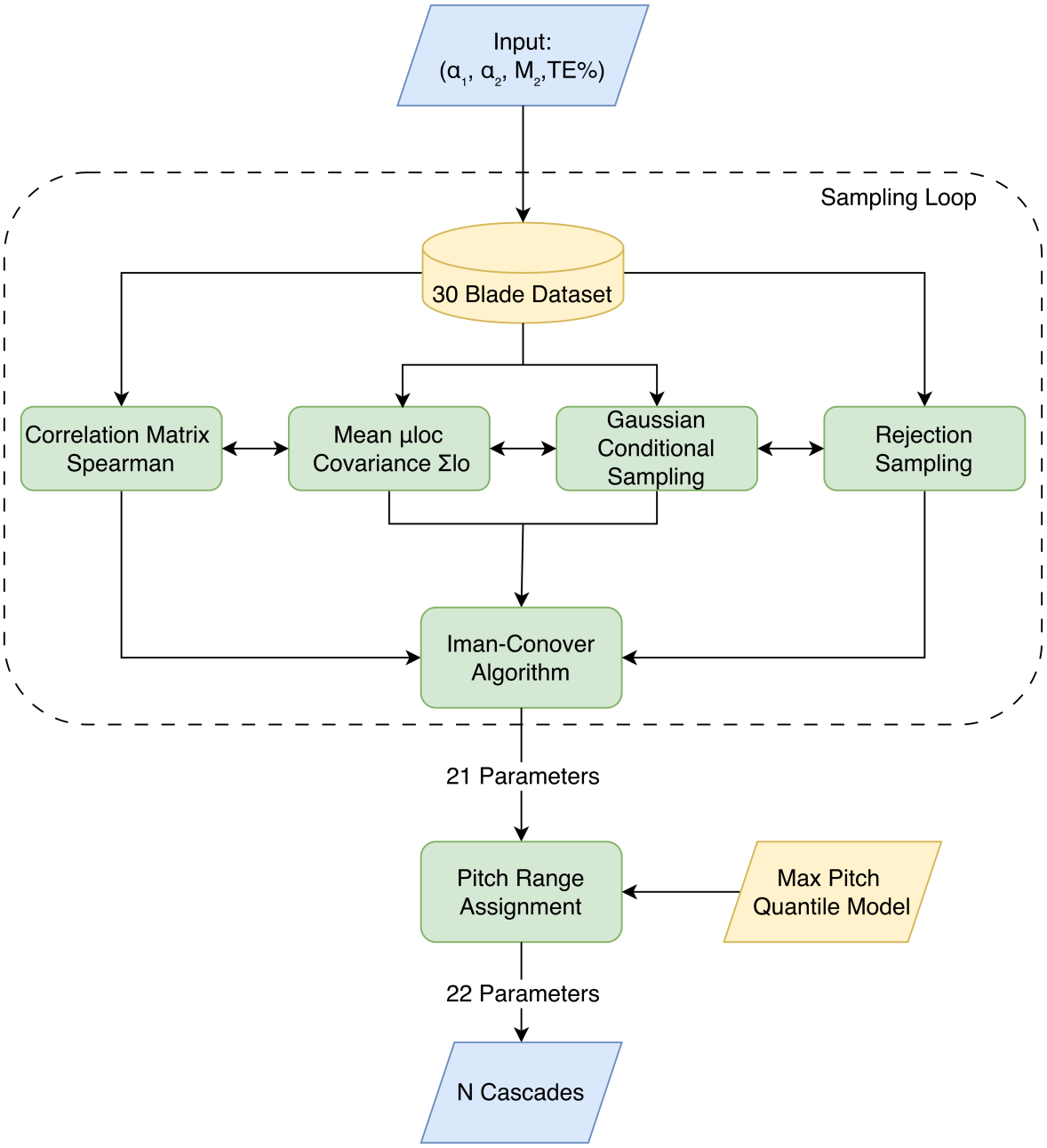


Figure 4.11: Complete pipeline of the DoE Blade Generation Algorithm.

4.6.3 DoE Results

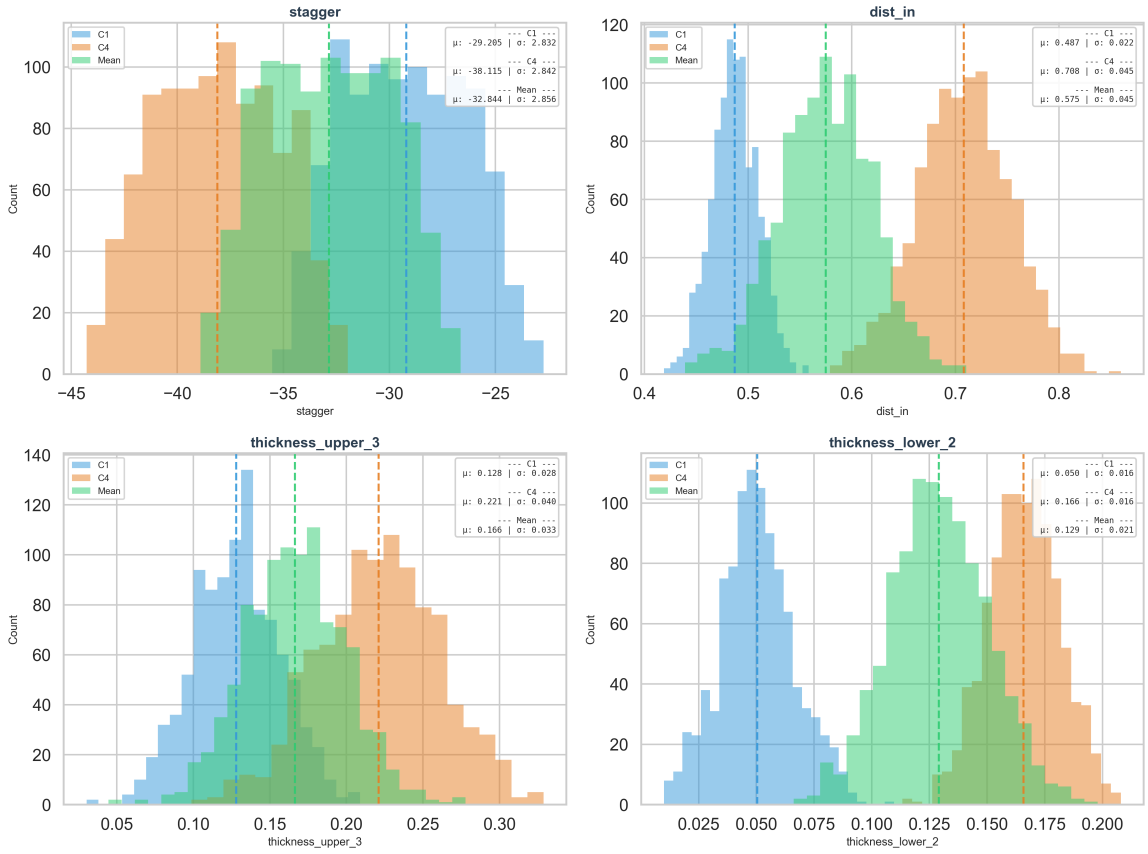
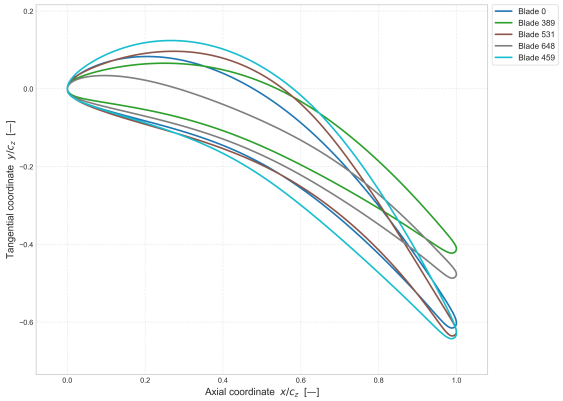


Figure 4.12: Marginals comparison of 4 important parameters (`stagger`, `dist_in`, `thickness_upper_3`, `thickness_lower_2`) between C1, C2 and the mean blade.

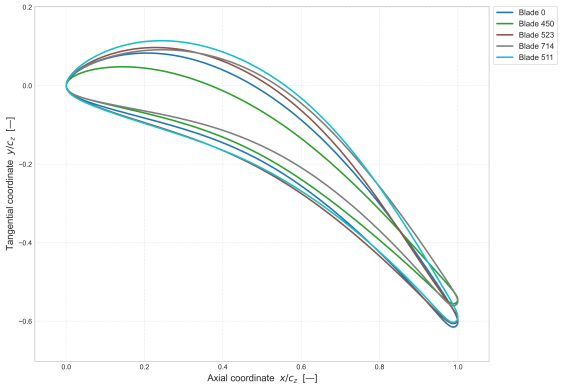
The complete pipeline described in the preceding sections was applied to generate blade families across the full duty space. In order to show the results of the Blade Generator Algorithm, for each operating condition a sample of **1000 blades** was produced. The results presented here focus on three representative duty points: the corner points C1 and C4, which probe the extremes of the design space, and a mid-range condition ($\alpha_1 = 26°$, $\alpha_2 = -62°$), selected to illustrate the behaviour of the adaptive nearest-neighbour anchor in a well-populated region of the database.

The marginal distributions (Figure 4.12) of the most significant geometric parameters are reported for C1 and C4, compared against the mean blade of the reference database, to highlight how the conditional sampler shifts and reshapes the parameter distributions in response to the prescribed duty.

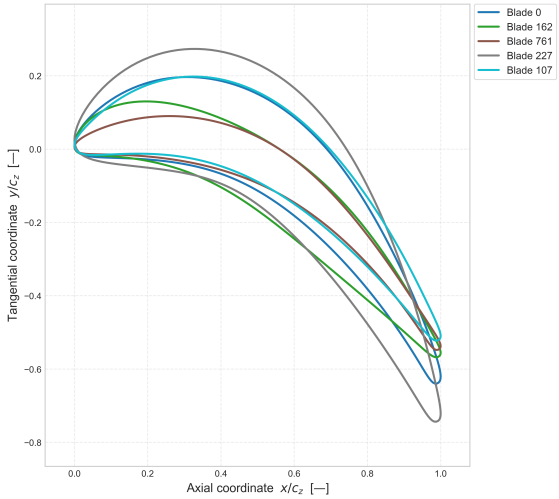
Subsequently, for each of the three duty points, two sets of **5 representative blades** are shown in Figure 4.13, extracted from the 1000-blade sample by maximising the normalised Euclidean distance across all 21 `ParaBlade` parameters (left column), and by maximising the distance across the thickness control points only at fixed stagger (right column).



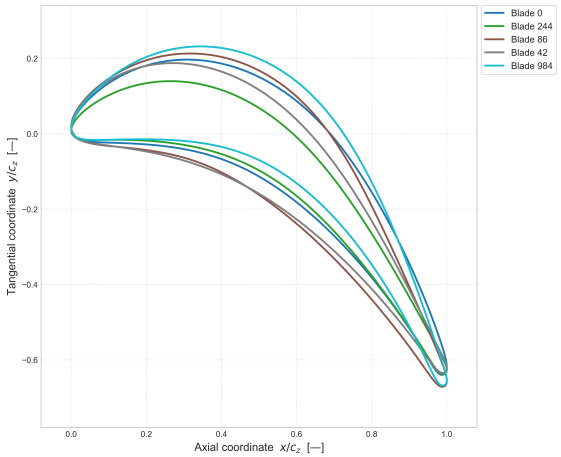
C1 — Global diversity



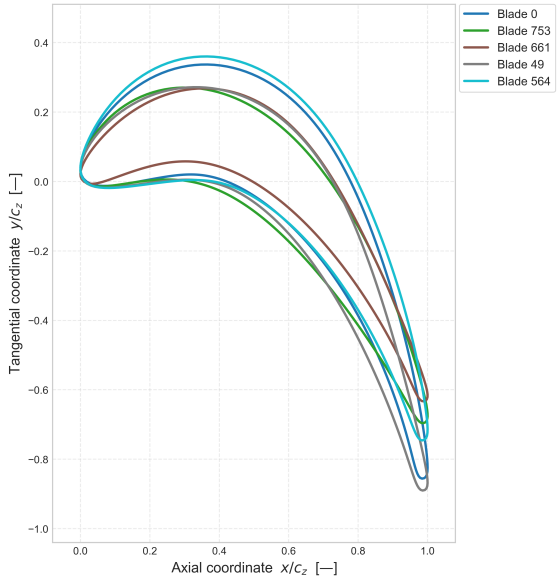
C1 — Thickness diversity at fixed stagger



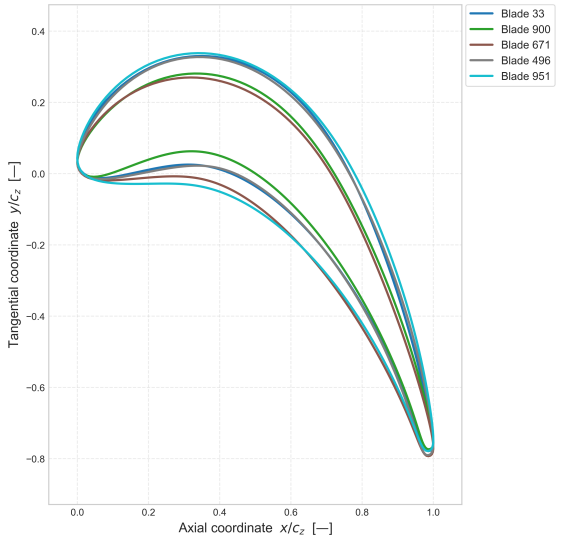
Mid — Global diversity



Mid — Thickness diversity at fixed stagger



C4 — Global diversity



C4 — Thickness diversity at fixed stagger

Figure 4.13: Representative blade profiles generated for three duty points.

Chapter 5

The Optimizer

The original ambition of the design tool developed in this work was to function as a complete inverse design system: given a target aerodynamic duty and a desired aerodynamic style, return the corresponding blade geometry. The difficulty of establishing a reliable mapping between geometric parameters and aerodynamic style forced a restriction of the problem to the duty alone. A prescribed duty $(\alpha_1, \alpha_2, M_2)$, however, does not uniquely define a cascade: an infinite family of geometrically distinct blades can satisfy the same flow conditions, each with different loss levels and loading distributions.

To close this ambiguity, an optimization layer was introduced into the tool. Starting from a single-objective formulation and progressively evolving towards a full multi-objective approach, the optimizer returns—for each duty point—a **Pareto front** of non-dominated solutions, giving the designer explicit control over the trade-off between aerodynamic losses and blade loading. The algorithm selected for this purpose is **NSGA-II** [29], a gradient-free genetic evolutionary algorithm widely adopted in turbomachinery optimization.

The chapter is organized as follows. Section 5.1 describes the preliminary single-objective optimization carried out with Differential Evolution, which validated the parametrization and the CFD pipeline before the full multi-objective campaign. Section 5.2 introduces the NSGA-II algorithm and its theoretical foundations. Finally, Section 5.3 presents the sensitivity study used to tune the objectives, constraints, and hyperparameters of the algorithm.

5.1 Preliminary Single-Objective Test

Before committing to a full multi-objective campaign, a preliminary single-objective optimization was carried out. This first test served a dual purpose: it represented the natural first step in the progressive development of the tool, and it provided a concrete validation of the entire pipeline—parametrization, DoE, and CFD evaluation—under a

controlled and interpretable setting.

The test case was fixed at a single duty point ($\alpha_1 = 30°$, $\alpha_2 = -70°$, $M_2 = 1.2$), with the trailing edge radius `radius_out`= 0.015 to hold TE thickness constant at 3% of the axial chord throughout. The objective was to minimize the kinetic energy loss coefficient ξ, subject to a constraint on the outlet flow angle α_2.

The kinetic energy loss coefficient ξ is defined as:

$$\xi = 1 - \frac{1 - \left(\frac{p_2}{p_{02}}\right)^{\frac{\gamma-1}{\gamma}}}{1 - \left(\frac{p_2}{p_{01}}\right)^{\frac{\gamma-1}{\gamma}}}, \tag{5.1}$$

where p_{01} is the total pressure at the inlet, p_{02} is the total pressure at the outlet, p_2 is the static pressure at the outlet, and γ is the ratio of specific heats. A value of $\xi = 0$ corresponds to an isentropic cascade, while $\xi > 0$ quantifies the fraction of kinetic energy lost to irreversibilities.

5.1.1 Objective Function and Angle Constraint

The outlet flow angle produced by a blade is not an input to the `MISES` solver but an output: it depends on the full geometry and cannot be prescribed directly. To steer the optimizer towards blades that achieve the target $\alpha_{2,\text{target}}$, a **soft penalty** was introduced, following the approach adopted by Porta [6]. The cost function is:

$$J = \xi \cdot \left[1 + k \cdot \max(0,\, \Delta\alpha_2 - \epsilon)^2\right], \tag{5.2}$$

where $\Delta\alpha_2 = |\alpha_{2,\text{target}} - \alpha_{2,\text{eff}}|$ is the absolute deviation of the effective outlet angle from the target, ϵ is a tolerance threshold below which no penalty is applied, and k is a scaling factor balancing the relative magnitude of the loss and the angular deviation.

The penalty is **unilateral**: it activates only when the blade underturns the flow, i.e., when the effective exit angle falls short of the target in magnitude. No penalty is applied to blades that overturn slightly. This asymmetry exploits the physics of the problem: a blade that underturns produces lower losses precisely because it deflects the flow less, and would therefore be artificially favoured by the optimizer in the absence of a constraint. Blades that overturn, by contrast, naturally tend back towards the target as the optimizer reduces their losses, since higher turning is aerodynamically more demanding.

The parameters were set to $k = 0.1$ and $\epsilon = 0.1°$. The choice of $k = 0.1$ was motivated by the need to keep the penalty term of the same order of magnitude as the loss, avoiding a regime where the optimizer ignores the angular constraint entirely or,

conversely, one where it sacrifices aerodynamic performance solely to match the exit angle. With these settings, it was observed that after approximately 5,000 function evaluations the optimizer consistently converged within the tolerance ϵ, confirming that the penalty formulation was sufficient to enforce the angular constraint without requiring a hard wall.

5.1.2 Optimizer Setup

The optimization was performed using the **Differential Evolution** (DE) algorithm, implemented via the `scipy.optimize.differential_evolution` module. DE is a population-based, gradient-free evolutionary strategy that evolves candidate solutions through mutation, crossover, and selection operations, making it well suited to non-convex, high-dimensional design spaces [30].

The algorithm was configured as follows:

Table 5.1: Differential Evolution hyperparameters for the single-objective preliminary test.

Parameter	**Value**	**Description**
Population size	50	Individuals per generation
Generations	100	Maximum number of generations
Total evaluations	5,000	Population × Generations
Crossover rate Cr	0.9	Probability of crossover per parameter
Mutation factor F	$\mathcal{U}[0.5, 1.0]$	Randomized to promote diversity
Strategy	`best/1/bin`	Mutation from best individual

The mutation factor F was sampled uniformly in $[0.5, 1.0]$ at each generation rather than fixed, following the dithering strategy recommended by Storn and Price [30]. This prevents premature convergence by maintaining a variable perturbation amplitude throughout the optimization.

The Generation 0 population of 50 individuals was produced by the blade generation pipeline described in Chapter 4, centered on the target duty point ($\alpha_1 = 30°$, $\alpha_2 = -70°$, $M_2 = 1.2$).

5.1.3 Results

Figure 5.1 shows the optimal blade geometry returned by the optimizer at the end of the 100 generations.

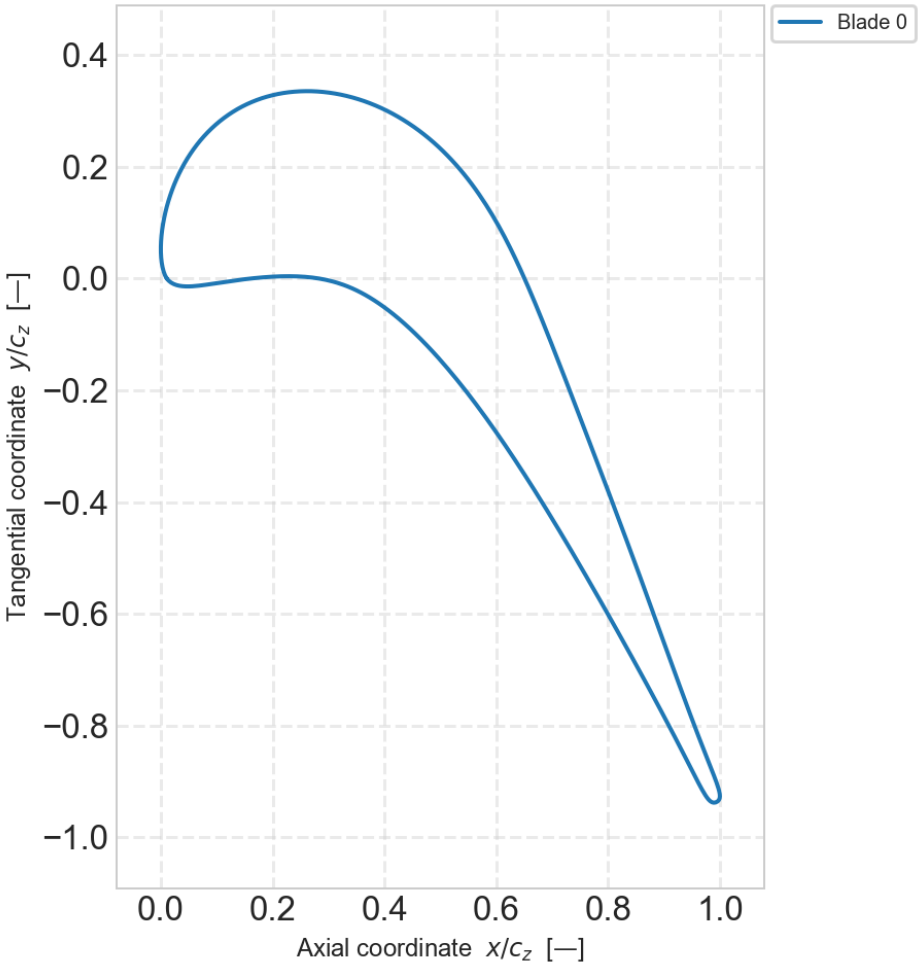


Figure 5.1: Optimal blade geometry configuration obtained via single-objective optimization.

Figure 5.2 shows the evolution of both the loss coefficient ξ and the fitness function J (Equation 5.2) over the 100 generations. The two quantities diverge whenever the angle penalty is active, while they converge to the same value once the optimizer has found blades that satisfy the angular constraint within the threshold ϵ.

A loss reduction of more than **30%** is achieved with respect to the Generation 0 population: the best individual (in term of cost function) at Generation 0 exhibits $\xi = 5.07\%$, while the best individual at the final generation reaches $\xi = 3.51\%$, as visible in Figure 5.3. The fitness function J shows good convergence towards the end of the campaign, with no significant improvement observed beyond the final generations.

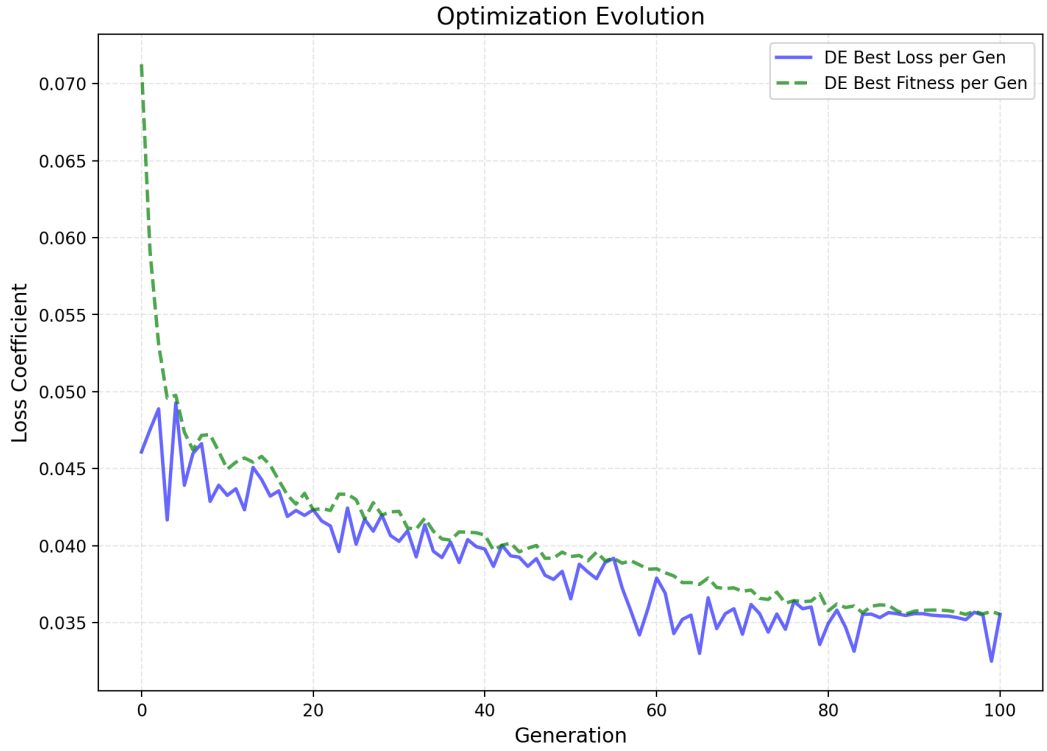


Figure 5.2: Convergence history of the single-objective optimization.

Figure 5.3 presents the full evolution of ξ, t/c_{ax}, and the stagger angle across all 100 generations. Each grey marker represents a single individual, while the red line connects the best fitness individual at each generation. The simultaneous convergence of all three quantities—loss, pitch, and stagger—towards stable values provided the evidence that 5,000 function evaluations were sufficient for this preliminary single-objective test: beyond a certain generation, no significant improvement is observed in any of the monitored parameters.

The aerodynamic performance of the optimal blade is further examined in Figures 5.4 and 5.5, which show the Mach number contour and the blade surface pressure distribution. The low loss achieved is primarily attributable to the near-zero deviation at the trailing edge: the flow exits the cascade following the metal angle of the blade almost exactly, with no separation or over-turning at the suction side. This confirms that the penalty formulation of Equation 5.2 was effective in driving the optimizer towards geometries that satisfy both the aerodynamic performance and the outlet angle requirements.

The optimal blade achieves a kinetic energy loss coefficient of $\xi = 3.51\%$, with an effective outlet flow angle of $\alpha_{2,\text{eff}} = -69.97°$, confirming convergence within the prescribed threshold $\epsilon = 0.1°$. The Mach number gradients visible on the suction side are primarily caused by the interaction between the boundary layer and the oblique

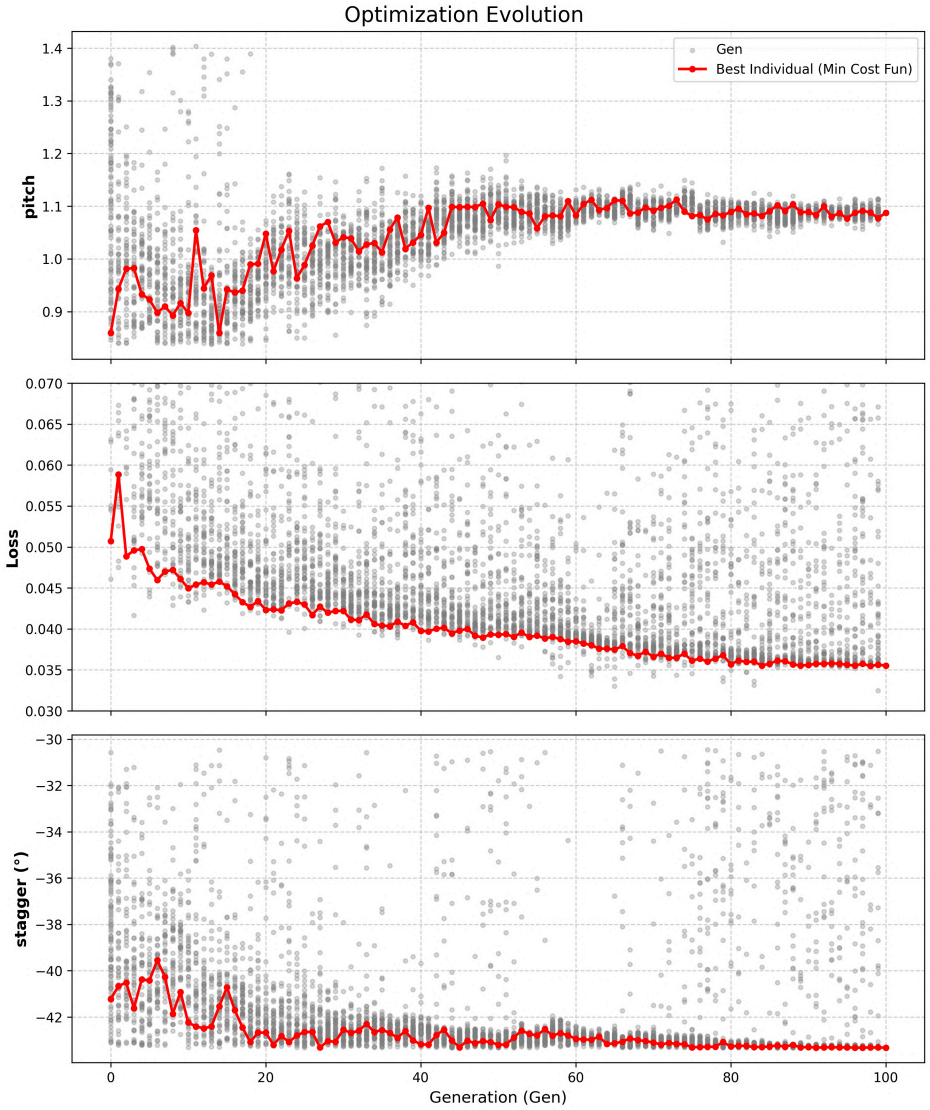


Figure 5.3: Evolution of ξ, t/c_{ax}, and stagger angle over 100 generations.

shock detaching from the trailing edge of the adjacent blade. The oscillations observed downstream of the first shock reflection are a direct consequence of the `MUCON` $= -1$ setting discussed in the `MISES` validation chapter. With a more refined formulation or a RANS solver, these oscillations reduce significantly. The chosen setting, however, offers a deliberate trade-off: it greatly improves numerical stability and computational speed—this blade converged in 30 iterations in just under 2 seconds—while maintaining a level of aerodynamic accuracy sufficient for the purposes of the optimization campaign.

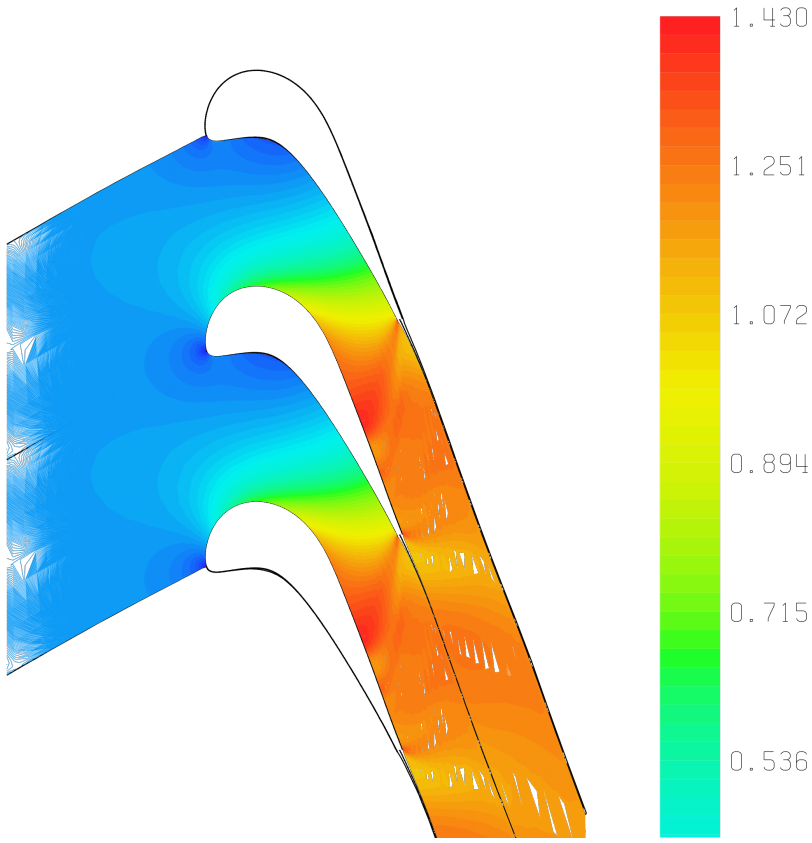


Figure 5.4: Mach number contours for the optimal blade configuration.

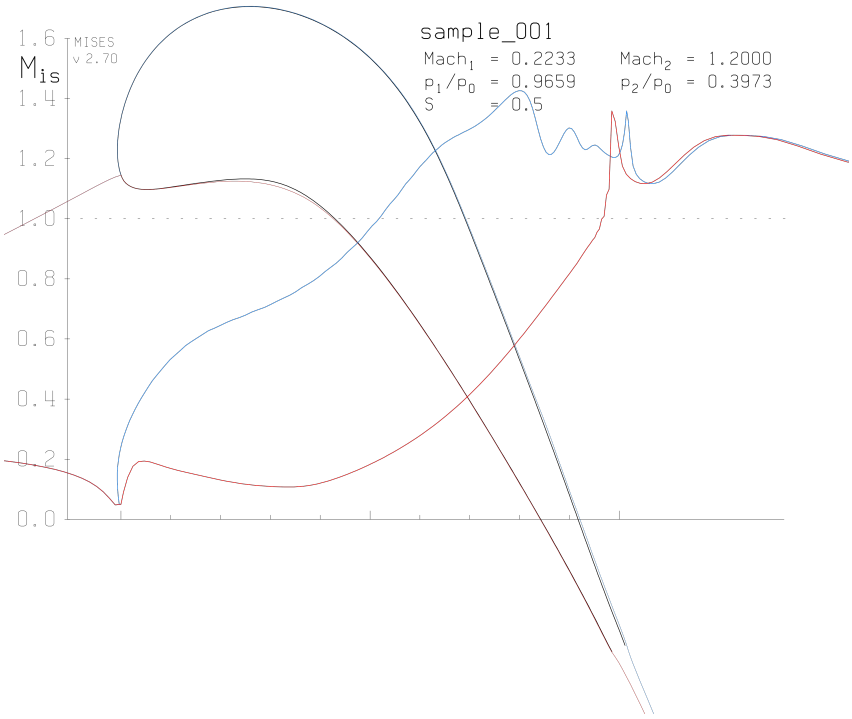


Figure 5.5: Blade surface loading distribution for the optimal blade configuration.

5.2 NSGA-II

5.2.1 Genetic Algorithms

A **genetic algorithm** (GA) is a population-based, gradient-free optimization method inspired by the mechanisms of biological evolution. Rather than iterating on a single candidate solution, a GA maintains a **population** of N individuals—in this context, N blade geometries—and evolves it over successive generations through three operators:

- **Selection**: individuals are chosen to reproduce with a probability proportional to their fitness, so that better solutions contribute more offspring to the next generation.
- **Crossover**: two parent individuals exchange portions of their parameter vectors to produce offspring that inherit characteristics from both.
- **Mutation**: random perturbations are applied to individual parameters with a small probability, preventing premature convergence and maintaining genetic diversity in the population.

At each generation, the offspring population is evaluated—here, through a full CFD simulation for each individual—and the cycle repeats. The absence of gradient information makes GAs naturally robust to discontinuities, local optima, and non-convex design spaces, which are all characteristic of transonic turbomachinery optimization.

5.2.2 The Pareto Front

In a multi-objective optimization problem with objectives $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$, no single solution minimizes all objectives simultaneously. A solution $\mathbf{x}_i$ is said to **dominate** $\mathbf{x}_j$ if it is at least as good in all objectives and strictly better in at least one:

$$\mathbf{x}_i \prec \mathbf{x}_j \iff \begin{cases} f_k(\mathbf{x}_i) \leq f_k(\mathbf{x}_j) & \forall\, k \\ f_k(\mathbf{x}_i) < f_k(\mathbf{x}_j) & \exists\, k \end{cases} \tag{5.3}$$

The **Pareto front** $\mathcal{F}^*$ is the set of all non-dominated solutions—those for which no other feasible solution exists that improves at least one objective without worsening another:

$$\mathcal{F}^* = \{\mathbf{x} \in \mathcal{X} \mid \nexists\, \mathbf{x}' \in \mathcal{X} : \mathbf{x}' \prec \mathbf{x}\}\,. \tag{5.4}$$

Every point on the Pareto front represents a different trade-off between the competing objectives. In the context of this work, the front connects the blade with minimum

aerodynamic losses at one extreme to the blade with maximum pitch—and therefore maximum loading—at the other, with a continuous family of compromise solutions in between.

5.2.3 Non-Dominated Sorting

NSGA-II (Non-dominated Sorting Genetic Algorithm II), proposed by Deb et al. [29], replaces the scalar fitness of classical GAs with a ranking mechanism based on Pareto dominance. At each generation, the population is sorted into a hierarchy of fronts $\mathcal{F}_1, \mathcal{F}_2, \ldots$ The first front $\mathcal{F}_1$ coincides with the current Pareto front of the population. Once its members are removed, the process is repeated on the remaining individuals to identify $\mathcal{F}_2$, and so on. Each solution is assigned a **rank** equal to the index of the front it belongs to, with lower rank always preferred during selection.

5.2.4 Crowding Distance

Within the same front, all individuals share the same rank. A secondary criterion is therefore needed to distinguish between them and maintain a well-distributed set of solutions along the front. NSGA-II uses the **crowding distance** d_i, which estimates how isolated a solution is in the objective space with respect to its nearest neighbours on the same front. For each objective f_k, the contribution to d_i is:

$$d_i^k = \frac{f_k(\mathbf{x}_{i+1}) - f_k(\mathbf{x}_{i-1})}{f_k^{\max} - f_k^{\min}}, \tag{5.5}$$

where $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i+1}$ are the neighbouring solutions along objective k, and $f_k^{\max}$, $f_k^{\min}$ are the extremes of that objective in the current front. The total crowding distance is $d_i = \sum_k d_i^k$, with boundary solutions assigned $d_i = \infty$ to ensure they are always preserved. Solutions in less crowded regions are preferred, promoting a uniform distribution of solutions along the Pareto front.

5.2.5 Selection, Elitism, and the Generation Cycle

At each generation t, offspring are produced from the current parent population P_t through **binary tournament selection**: two individuals are drawn at random and compared according to the **crowded comparison operator**—lower rank is preferred, and ties are broken by larger crowding distance. The selected parents undergo crossover and mutation to generate an offspring population Q_t of the same size N.

The two populations are then merged into a combined pool $R_t = P_t \cup Q_t$ of $2N$ individuals, which is sorted by non-dominated ranking into fronts $\mathcal{F}_1, \mathcal{F}_2, \ldots$ The next parent population P_{t+1} is assembled by accepting fronts in order from $\mathcal{F}_1$ onwards. If a

front cannot be accepted in its entirety without exceeding N, individuals within it are selected by decreasing crowding distance, and the remaining individuals are discarded. This **elitist** strategy ensures that the best solutions found at any generation are never lost, and that the Pareto front can only improve from one generation to the next.

The complete generation cycle is illustrated in Figure 5.6.

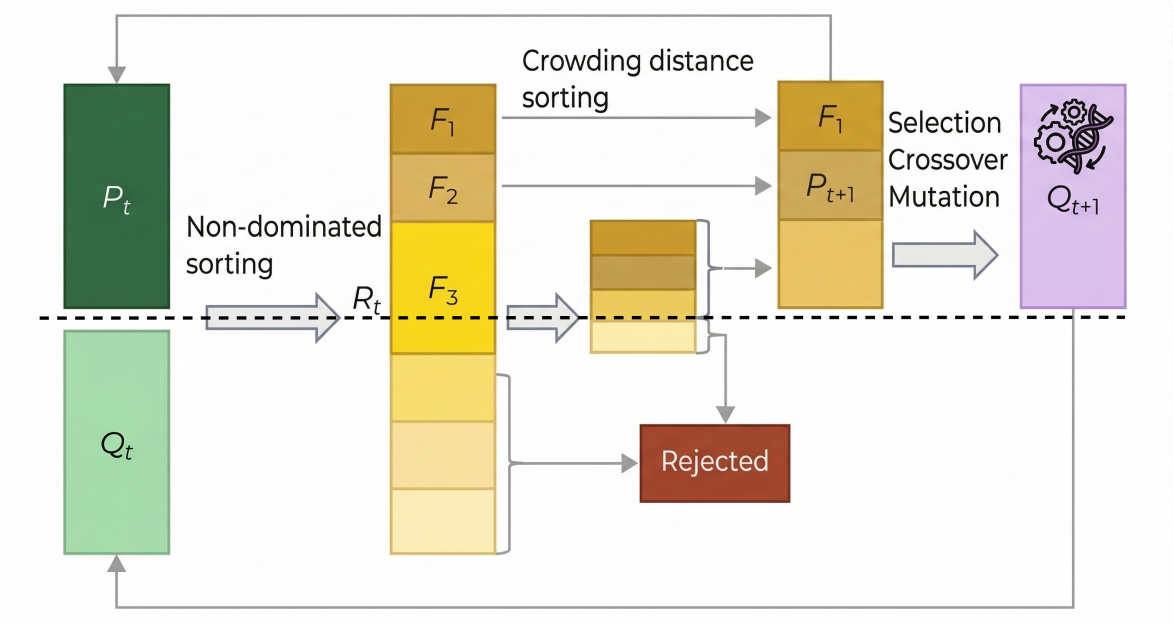


Figure 5.6: Typical NSGA-II generation cycle.

5.3 Sensitivity Study

The configuration of the NSGA-II optimizer involves a large number of choices: the objective functions, the constraint handling strategy, the crossover operator, and the hyperparameters of the algorithm. Rather than fixing these a priori, a dedicated **sensitivity study** was carried out to identify the configuration that best balances convergence speed, Pareto front quality, and robustness across the design space. All runs in this study were performed with a fixed budget of **10,000 CFD evaluations**, corresponding to 200 generations of 50 individuals each.

5.3.1 Choice of the Test Case

The sensitivity study was conducted on a single representative duty point, chosen to be the **most challenging in the design space**. The corner point **C2** ($\alpha_1 = 50°$, $\alpha_2 = -50°$, $M_2 = 1.2$, TE $= 3\%$) was selected for this purpose.

Preliminary runs across the design space had shown that C2 consistently produced the highest number of non-converged `MISES` simulations, the least well-defined Pareto

fronts, and the longest wall-clock time per optimization run. Its high inlet angle combined with a relatively low deflection makes it **aerodynamically demanding**: the blade must turn the flow significantly at the leading edge while avoiding separation on the suction side in a transonic regime. Any configuration of the optimizer that performs well on C2 can therefore be expected to perform adequately across the rest of the design space.

Notably, even with the full budget of 10,000 function evaluations, C2 remains the duty point with the **highest total computation time** in the entire campaign, owing to the large fraction of non-converged individuals that must still be processed and logged before being discarded.

Optuna and the Ask-and-Tell Interface

The optimization was implemented using the `Optuna` framework, an open-source hyperparameter optimization library that provides a flexible and modular interface for evolutionary algorithms. Specifically, the `NSGAIISampler` was used as the sampling backend, which implements the NSGA-II logic described in Section 5.2—non-dominated sorting, crowding distance, binary tournament selection, and elitist replacement—directly within `Optuna`'s trial management system.

The optimizer operates through an **ask-and-tell** pattern: at each step, the sampler *asks* for a new set of candidate parameters, the CFD pipeline evaluates the corresponding blade, and the objective values are *told* back to the sampler. This decoupled interface allowed the full optimization loop—from `ParaBlade` geometry generation to `MISES` simulation and post-processing—to be integrated without any modification to the core NSGA-II logic, and made it straightforward to log every evaluated individual, including non-converged simulations, into the output file.

5.3.2 Choice of Crossover Operator

The first design choice investigated in the sensitivity study was the **crossover operator**, which determines how genetic information is recombined between two parent individuals to produce offspring. Three operators were compared:

- **SBX** (Simulated Binary Crossover) [29]: the standard crossover operator for real-valued genetic algorithms. For each parameter, offspring values are sampled from a probability distribution centred between the two parents, with a spread controlled by the distribution index η_c.
- **VSBX** (Variable Simulated Binary Crossover): a modification of SBX that generates offspring without excluding any region of the parameter space, while retaining the core properties of the original operator. VSBX maintains a more uniform

coverage of the feasible domain, improving global exploration particularly in the early generations.

- **UNDX** (Unimodal Normal Distribution Crossover) [31]: offspring are generated by sampling from a **multivariate normal distribution** built from **three** parent individuals. The distribution is centred on the midpoint of two primary parents, with the principal axis oriented along the direction connecting them and the spread in the orthogonal directions controlled by the position of the third parent. This multi-parent scheme promotes a more informed exploration of the parameter space compared to standard two-parent crossover operators.

The three operators were compared starting from an identical Generation 0 population, with the goal of assessing whether all three converge to comparable ranges of ξ and t/c_{ax} on the final Pareto front, and of evaluating the **quality and regularity** of both the front and the generated blade profiles.

Table 5.2: Crossover operator hyperparameters used in the sensitivity study.

Parameter	**SBX**	**VSBX**	**UNDX**
Distribution index η_c	20	20	—
Mutation probability p_m	$1/N_{\text{var}} \approx 5\%$	$1/N_{\text{var}} \approx 5\%$	—
σ_ξ	—	—	0.9
σ_η	—	—	0.1
Number of parents	2	2	3

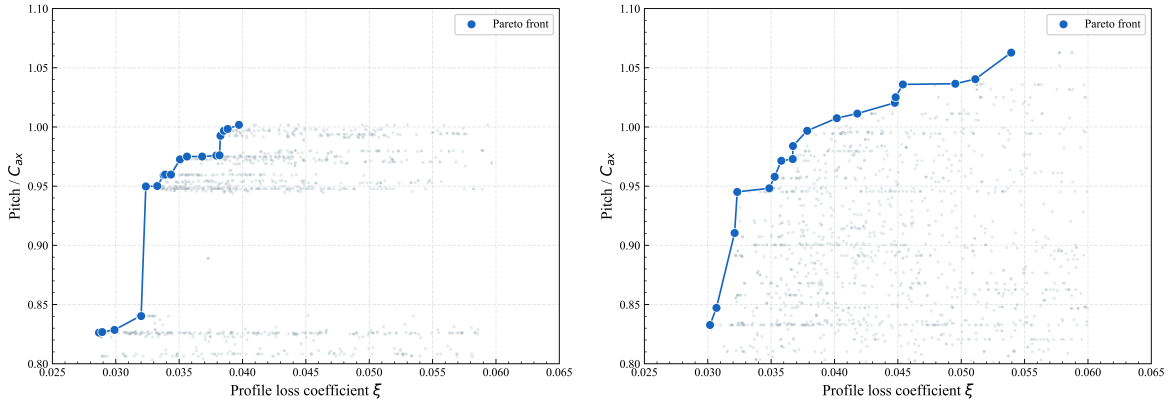
Results

Figure 5.7 shows the final Pareto fronts obtained with SBX and VSBX on the C2 duty point. SBX produces the least regular front, characterized by visible discontinuities and a reduced number of non-dominated solutions. VSBX yields a significantly more regular front, reflecting the broader parameter space coverage introduced by its modified formulation.

Figure 5.8 shows the corresponding result for UNDX. UNDX produces the **most regular and well-distributed front** among the three operators, with a higher number of non-dominated solutions and no visible discontinuities.

The direct comparison of all three operators in the objective space is shown in Figure 5.9.

The discriminating factor between VSBX and UNDX emerges from the inspection of representative blade profiles on the final front, shown in Figure 5.10, where two blades



SBX VSBX

Figure 5.7: Final Pareto fronts obtained with SBX (left) and VSBX (right) on the C2 duty point.

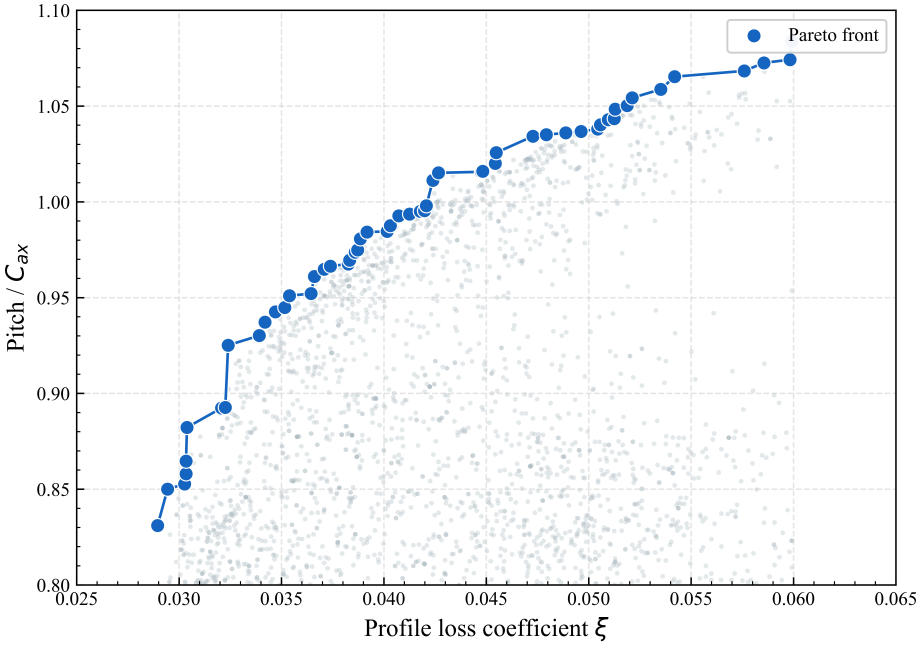


Figure 5.8: Final Pareto front obtained with UNDX on the C2 duty point.

at comparable loss ($\xi \approx 4.1\%$) are overlaid.

While the two profiles appear similar at first glance, the curvature distributions along the pressure side, shown in Figure 5.11, reveal a clear difference. The blade generated by VSBX exhibits a higher absolute curvature and a larger curvature gradient along the pressure side, a signature of the localized geometric artifacts introduced by the crossover mechanism. The UNDX blade, by contrast, displays a **smooth and monotonic curvature distribution**, consistent with a physically sound pressure side geometry.

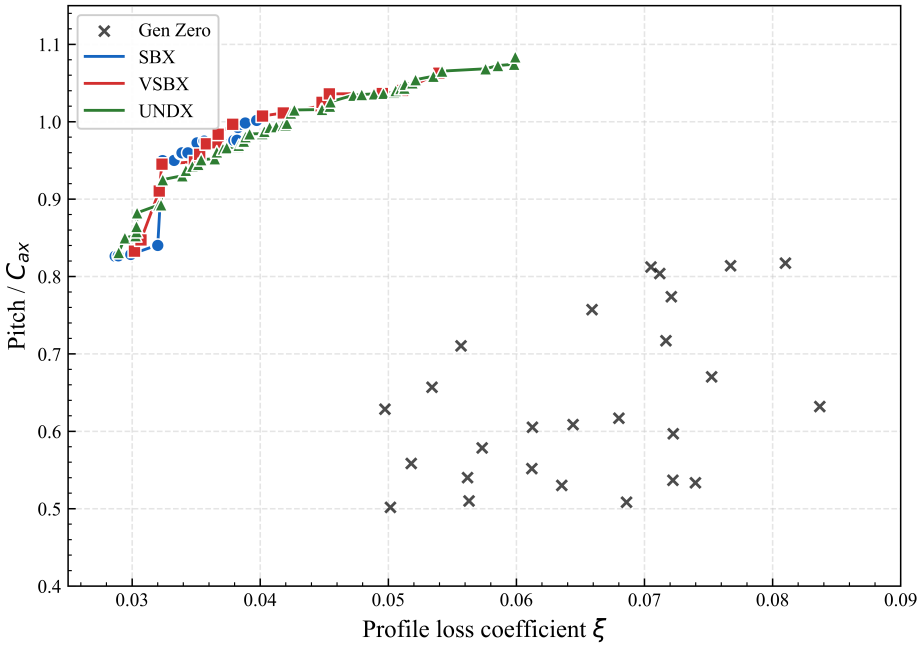


Figure 5.9: Comparison of the final Pareto fronts obtained with SBX, VSBX, and UNDX on the C2 duty point, starting from the same Generation 0 population.

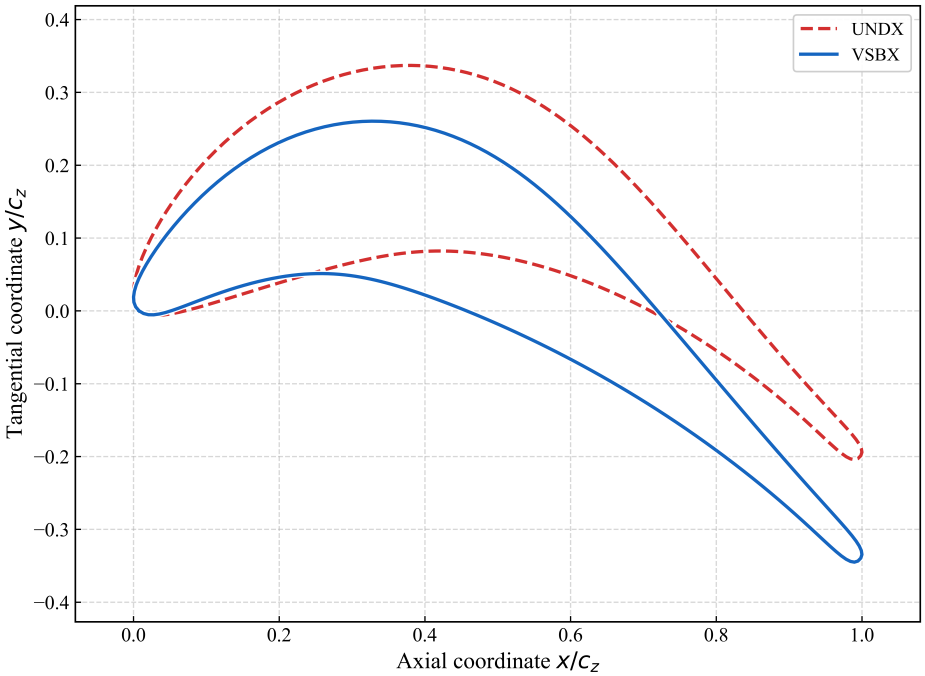


Figure 5.10: Overlay of two blade profiles at comparable loss ($\xi \approx 4.1\%$) obtained with VSBX and UNDX on the C2 duty point.

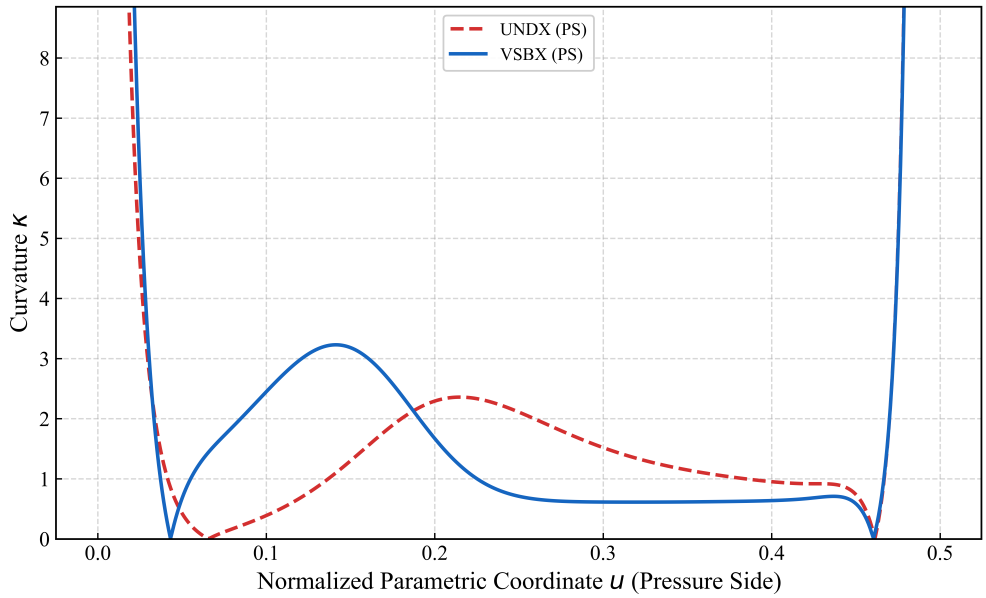


Figure 5.11: Pressure side curvature distributions for the two blades shown in Figure 5.10.

UNDX was therefore selected as the crossover operator for all subsequent optimizations. Beyond the superior blade quality, this choice is also motivated by a consistency argument with the DoE methodology developed in Chapter 4: the multivariate normal distribution that governs offspring generation in UNDX is the **same distributional framework** used in the conditional Gaussian sampler to generate the Generation 0 population. The optimization therefore inherits and continues the same statistical structure established at the sampling stage, rather than introducing a discontinuous recombination mechanism.

5.3.3 Choice of the Loading Objective

The second objective of the optimization—representing the blade loading or lifting capacity—is not uniquely defined in turbomachinery literature. Three candidates were evaluated.

Zweifel Coefficient

The Zweifel coefficient Z_w is the most widely adopted non-dimensional measure of blade loading in turbine cascades [32], defined as the ratio of the actual blade circulation to a reference dynamic pressure force:

$$Z_w = \frac{\oint p \, \mathrm{d}x}{c_{ax} \left(p_{01} - p_2 \right)}, \tag{5.6}$$

where $\oint p\,\mathrm{d}x$ is the integral of the static pressure around the blade profile in the axial direction, c_{ax} is the axial chord, p_{01} the inlet total pressure, and p_2 the outlet static pressure.

Coull Circulation Coefficient

Coull and Hodson [33] introduced an alternative loading coefficient C_o, defined as the integral of the square root of the local pressure coefficient along the blade surface, normalised by the suction surface arc length S_0:

$$C_o = \oint \sqrt{\frac{p_{01} - p}{p_{01} - p_2}}\,\mathrm{d}\Big(\frac{S}{S_0}\Big) = \oint \sqrt{C_p}\,\mathrm{d}\Big(\frac{S}{S_0}\Big). \tag{5.7}$$

By integrating along the curvilinear suction surface arc rather than along the axial direction, C_o becomes **inherently sensitive to the curvature distribution** of the blade and to its orientation within the cascade channel, capturing aspects of the loading distribution that Z_w cannot represent. Furthermore, since the normalization is performed with respect to the local suction surface geometry rather than to pitch and chord, C_o remains comparable across blades operating at different duties, making it a more general metric for assessing and ranking loading across the full design space.

Pitch

The third candidate is the pitch-to-axial-chord ratio t/c_{ax} itself. At a fixed duty $(\alpha_1, \alpha_2, M_2)$, the Euler work per unit mass W is proportional to the change in tangential velocity ΔV_u, which is fixed by the velocity triangles:

$$W \propto \dot{m}\,\Delta V_u = \rho\, V_{ax}\, t \cdot \Delta V_u, \tag{5.8}$$

where at fixed duty ΔV_u and V_{ax} are approximately constant, so the total work output **scales directly with the pitch** t. Maximising the pitch therefore maximises the work output per blade passage and, as a direct consequence, minimises the number of blades required for a given stage duty—reducing the weight and manufacturing cost of the machine.

Results

Figure 5.12 shows the three Pareto fronts obtained with the three loading objectives, plotted with Z_w on the vertical axis for a direct comparison. The loss ranges achieved by the three runs are practically identical, confirming that the choice of loading objective does not affect the optimizer's ability to minimize ξ.

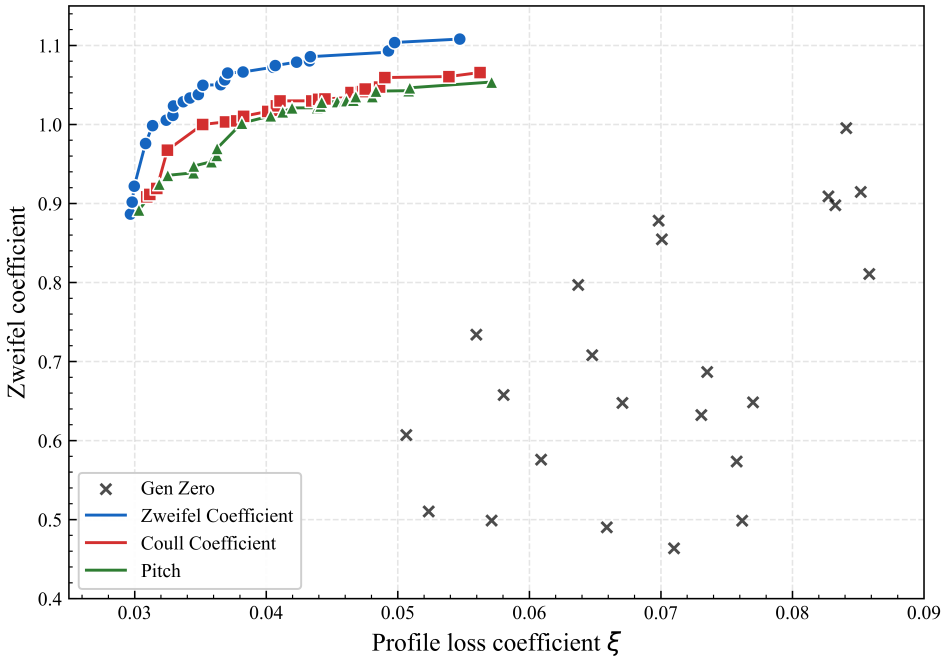


Figure 5.12: Pareto fronts obtained with Zweifel, Coull, and Pitch as the loading objective, plotted in the ξ–Z_w plane.

As expected, the run that directly maximizes Z_w reaches slightly higher values of the Zweifel coefficient. However, when the same fronts are plotted with the pitch t/c_{ax} on the vertical axis (Figure 5.13), the three runs achieve **comparable pitch ranges**.

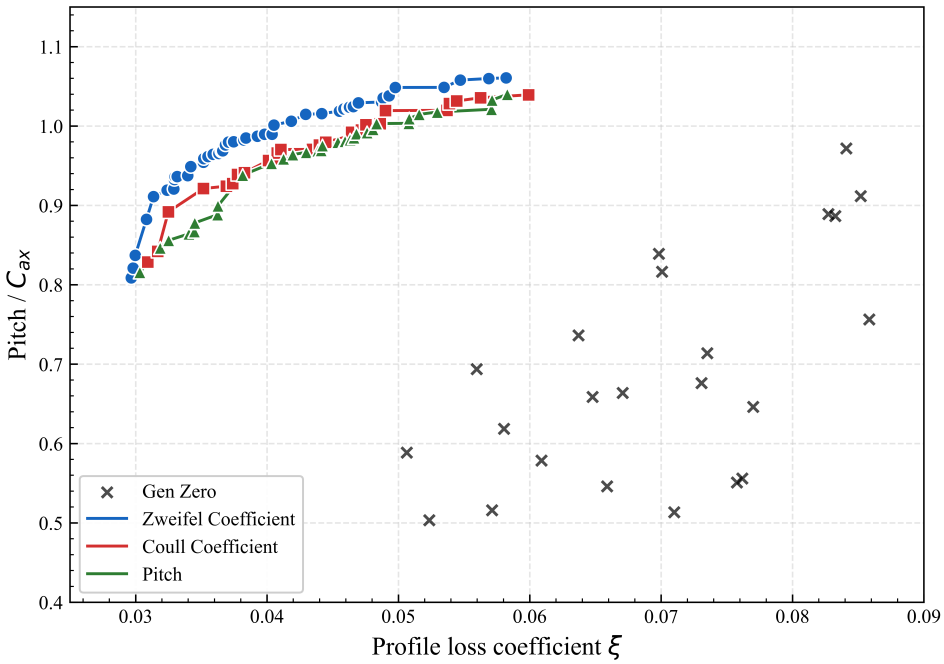


Figure 5.13: Pareto fronts obtained with Zweifel, Coull, and Pitch as the loading objective, plotted in the ξ–t/c_{ax} plane.

This observation suggests that the higher Z_w values reached by the Zweifel run do not necessarily correspond to blades that ingest more mass flow or extract more work. Instead, they may be an **artefact of the pressure integration**: numerical oscillations along the blade surface—such as those induced by the `MUCON` $= -1$ setting discussed in the `MISES` validation chapter, or localised pressure spikes near the trailing edge—can artificially inflate the value of Z_w without any physical counterpart. Since Z_w is computed as a surface integral of the pressure distribution, any non-physical perturbation of that distribution is directly reflected in the objective value, potentially biasing the optimizer towards geometries that exploit these numerical features rather than genuine aerodynamic loading. Furthermore, the Zweifel front remains the least uniform of the three, with more visible discontinuities and fewer non-dominated individuals.

For the Coull coefficient, the front plotted in the ξ–C_o plane (Figure 5.14) is well-defined and more uniform than the Zweifel front, with a higher number of non-dominated solutions and fewer discontinuities. Its quality is comparable to that of the pitch front.

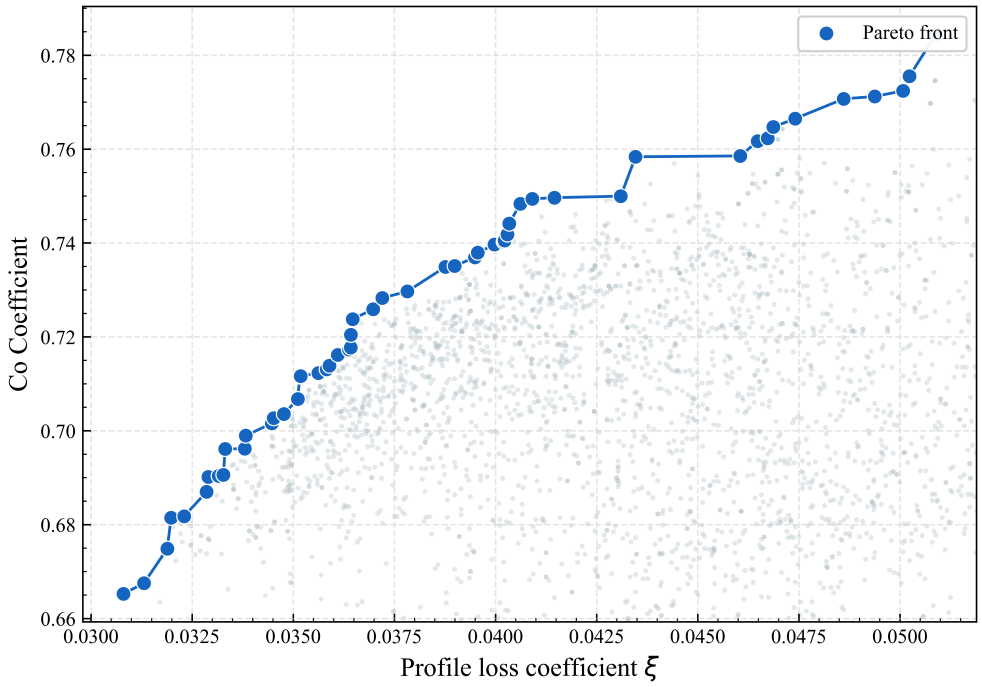
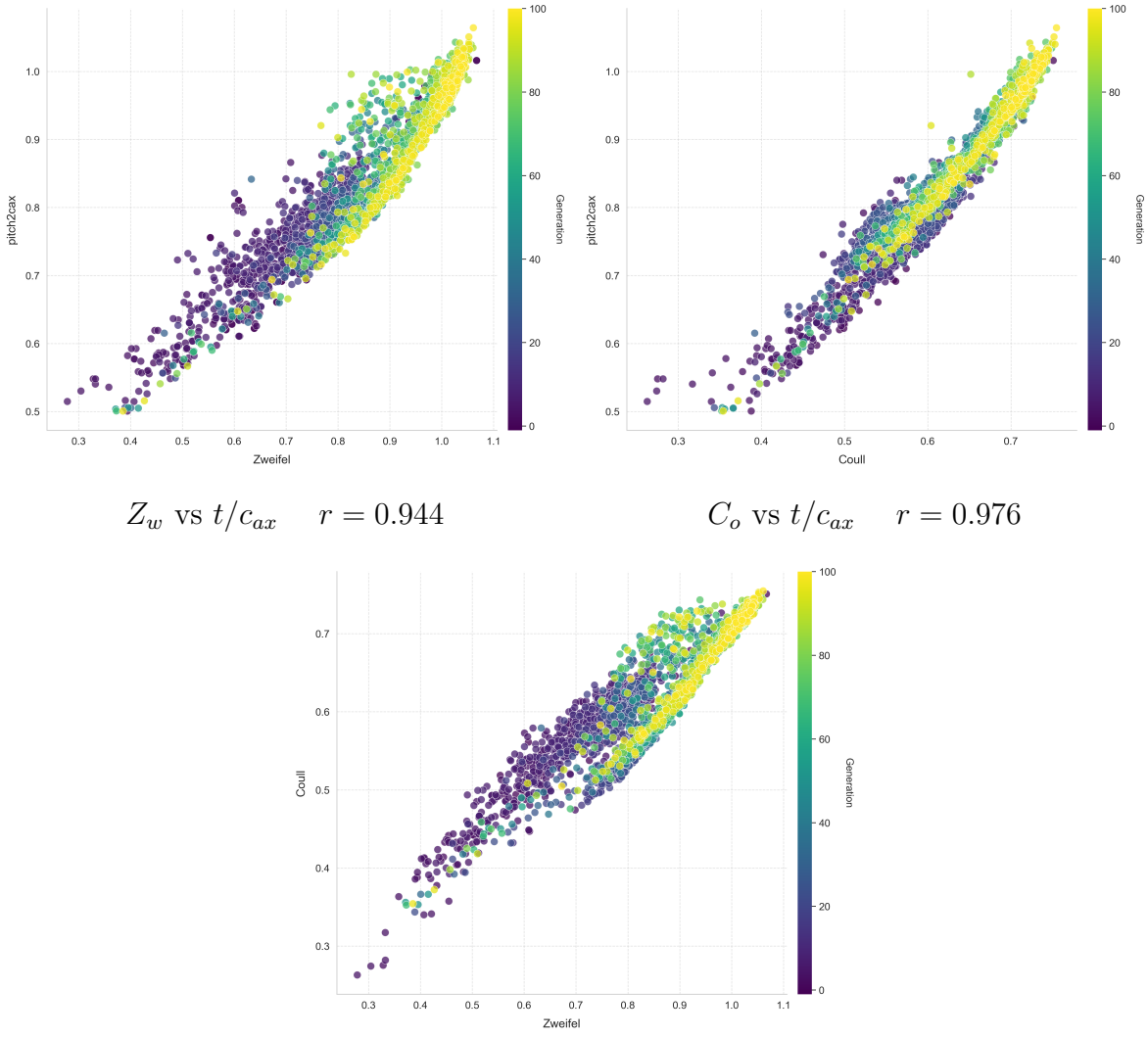


Figure 5.14: Pareto front obtained with Coull as the loading objective, plotted in the ξ–C_o plane.

The relationship between the three loading metrics across all evaluated individuals is shown in Figure 5.15. The Pearson correlation coefficient r—a measure of the strength of the linear relationship between two variables, ranging from -1 (perfect negative correlation) to $+1$ (perfect positive correlation)—was computed for each pair. The high values obtained confirm that, at fixed duty, all three quantities are effectively **equivalent proxies** for blade loading.



Z_w vs t/c_{ax} $r = 0.944$ C_o vs t/c_{ax} $r = 0.976$

Z_w vs C_o $r = 0.918$

Figure 5.15: Scatter plots of the three loading metrics across all evaluated individuals in the C2 sensitivity run.

Pitch was selected as the loading objective for all subsequent optimizations, for three complementary reasons. First, together with Coull, it produces the most uniform and well-distributed Pareto front. Second, and most importantly, it is **entirely decoupled** from the surface pressure integration: maximising t/c_{ax} directly increases the mass flow through the passage and therefore the work output, without depending on local pressure values that may be affected by numerical artefacts of the `MISES` solver. This eliminates any risk of the optimizer exploiting integration-based biases to inflate the objective value without a genuine aerodynamic benefit—a risk that affects both Z_w and C_o to different degrees. Third, it was observed that blades on the Coull Pareto front tend to be a bit thinner, with lower $t/c_{\max}$ values compared to the Zweifel and Pitch runs. Since adequate blade thickness is a prerequisite for internal cooling channels

and structural integrity in real transonic turbine applications, the pitch objective was preferred to avoid systematically driving the optimizer towards geometrically marginal profiles.

The difference in blade geometry between the three objectives is illustrated in Figure 5.16, where the blade with the best loss achieved on each Pareto front is shown ($\xi \approx 3\%$ in all three cases).

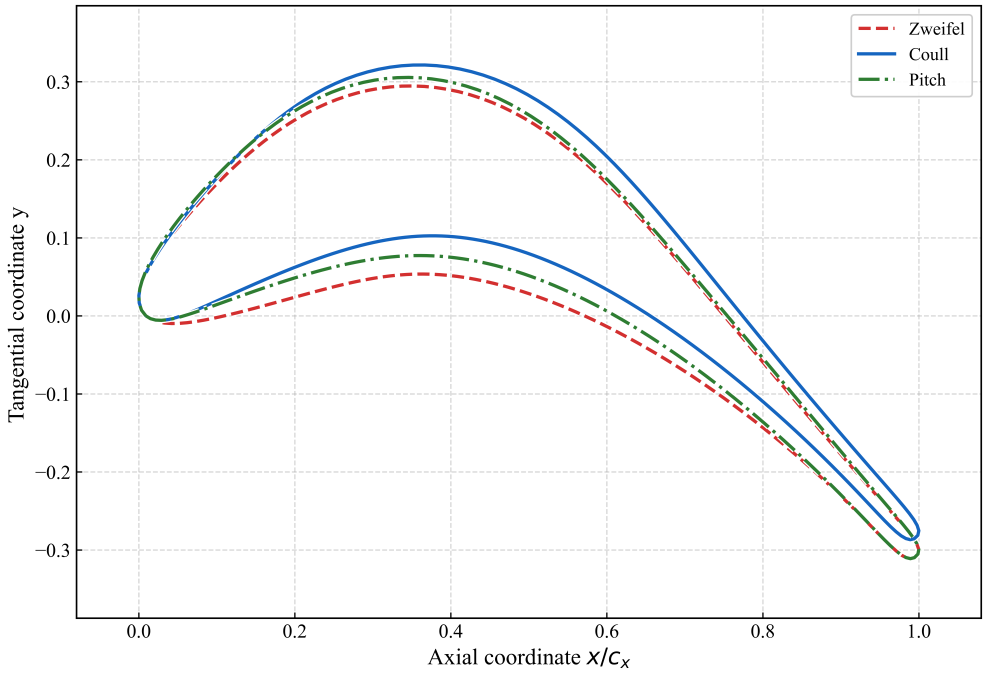


Figure 5.16: Blade profiles with the best loss ($\xi \approx 3\%$) from the Pareto front of each loading objective run.

5.3.4 Choice of the Angle Constraint Strategy

Constraint Handling in NSGA-II

Constraint handling in genetic algorithms requires particular care, since the concept of feasibility interacts directly with the selection and ranking mechanisms. In NSGA-II, infeasible individuals are not discarded but ranked according to the following preference rule:

- A feasible individual **always dominates** an infeasible one.
- Between two infeasible individuals, the one with the **smaller constraint violation** is preferred.
- Between two feasible individuals, the standard crowded comparison operator applies.

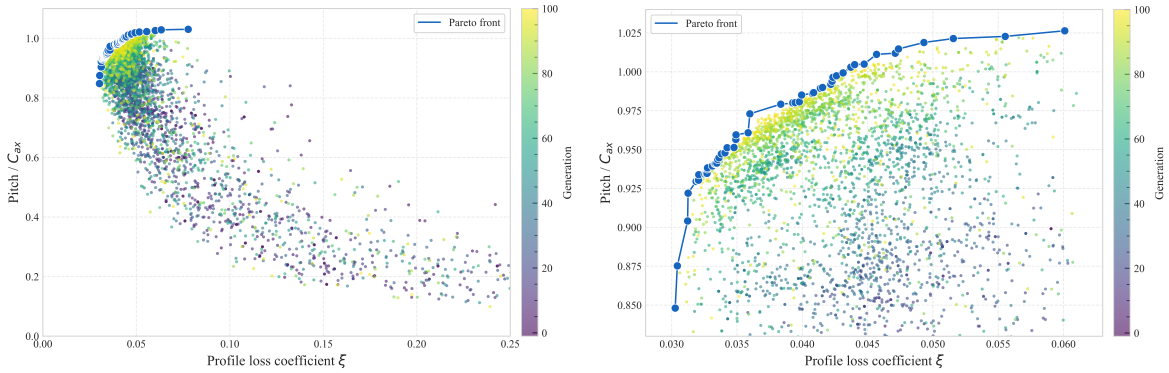
This mechanism preserves genetic diversity: infeasible solutions remain in the population and can still contribute genetic material through crossover, while never entering the Pareto front as long as feasible individuals exist. In the literature, hard constraints are generally preferred in multi-objective optimization [34]. Both strategies were nonetheless tested and compared here. The two approaches are:

- **Hard constraint**: any individual with $\Delta\alpha_2 = |\alpha_{2,\text{target}} - \alpha_{2,\text{eff}}| > \epsilon$ is marked as infeasible and ranked below all feasible solutions, with $\epsilon = 0.5°$.
- **Penalty**: the angular deviation is incorporated into both objective functions, penalizing loss upwards and loading downwards:

$$J_\xi = \xi \cdot \left[1 + k \cdot \max(0,\, \Delta\alpha_2 - \epsilon)^2\right], \tag{5.9}$$

$$J_{t/c_{ax}} = \frac{t/c_{ax}}{1 + k \cdot \max(0,\, \Delta\alpha_2 - \epsilon)^2}. \tag{5.10}$$

  Blades far from the angular target are pushed towards high loss and low loading, producing a characteristic half C-shaped distribution in the ξ–t/c_{ax} plane that converges towards the Pareto front as the constraint is satisfied.



Full objective space Zoom on the Pareto front

Figure 5.17: Penalty run on the C2 duty point.

Results

The penalty formulation produces a particularly instructive result, shown in Figure 5.17. The C-shaped distribution of evaluated individuals in the ξ–t/c_{ax} plane is a direct visualization of the angular violation gradient: at one extreme, blades that are far from the target angle are penalized towards high loss and near-zero loading; as the optimizer learns the correct camberline geometry and the angular deviation decreases, the individuals progressively migrate towards the true Pareto front. This guided exploration

demonstrates that the penalty formulation is not merely a constraint enforcement tool, but acts as an effective navigation mechanism that steers the population towards the feasible region in a smooth and continuous manner.

The direct comparison between the two strategies is shown in Figure 5.18.

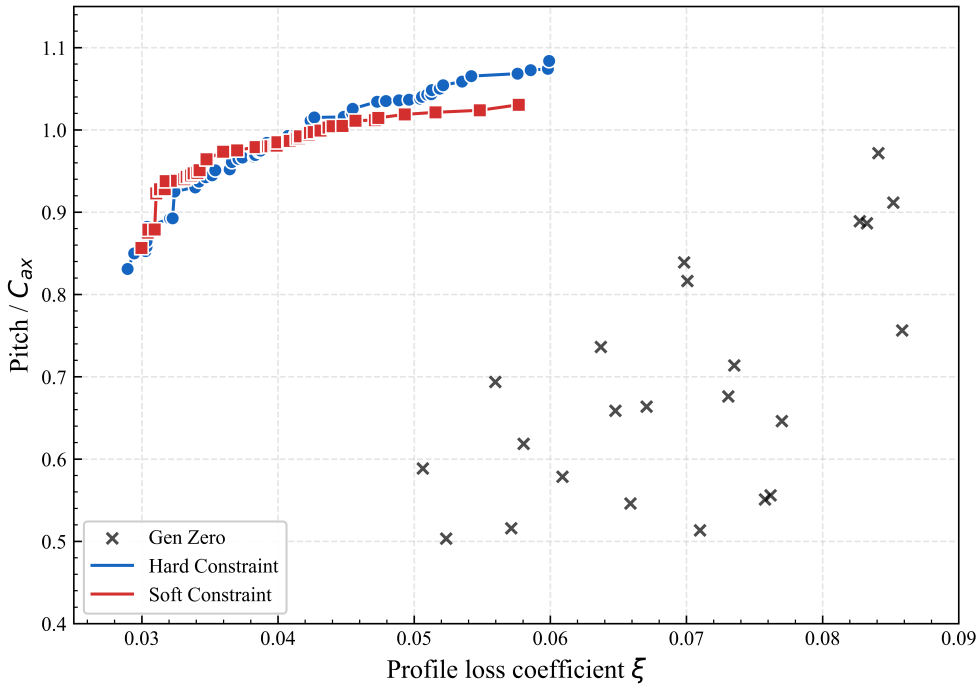


Figure 5.18: Comparison of the Pareto fronts obtained with hard constraint and penalty on the C2 duty point.

Both strategies reach comparable loss and pitch ranges. The hard constraint achieves slightly lower minimum loss, slightly higher maximum pitch, and a marginally larger number of non-dominated solutions. Blade profiles are similar, though penalty blades occasionally exhibit slightly more curvature irregularities on the pressure side.

The hard constraint was selected for all subsequent optimizations, as it exploits the native feasibility ranking mechanism of NSGA-II directly, without requiring the tuning of a penalty constant k whose optimal value may vary across duty points.

5.3.5 Hard Constraint on Loss

To prevent the optimizer from wasting CFD evaluations on aerodynamically trivial configurations—such as separated blades that produce very high losses—a **hard upper bound on** ξ was imposed in addition to the angular constraint. Any individual exceeding this bound is marked as infeasible regardless of its angular compliance.

The bound was estimated a priori using the same **dataset of 41 exploratory NSGA-II runs** described in Section 4.5.2. A second-degree polynomial regression model was trained on these data to predict the minimum achievable loss as a function

of the duty point $(\alpha_1, \alpha_2, M_2)$, yielding $R^2 = 0.9203$ and MAE $= 0.0023$ on the training set, as shown in the parity plot of Figure 5.19. Precision was not the primary goal: the model provides an indicative lower bound on the loss, which is then **multiplied by a factor of 2** to set a conservative feasibility threshold $\xi_{\max}$. This ensures that the constraint eliminates genuinely poor solutions without excluding any aerodynamically meaningful region of the design space.

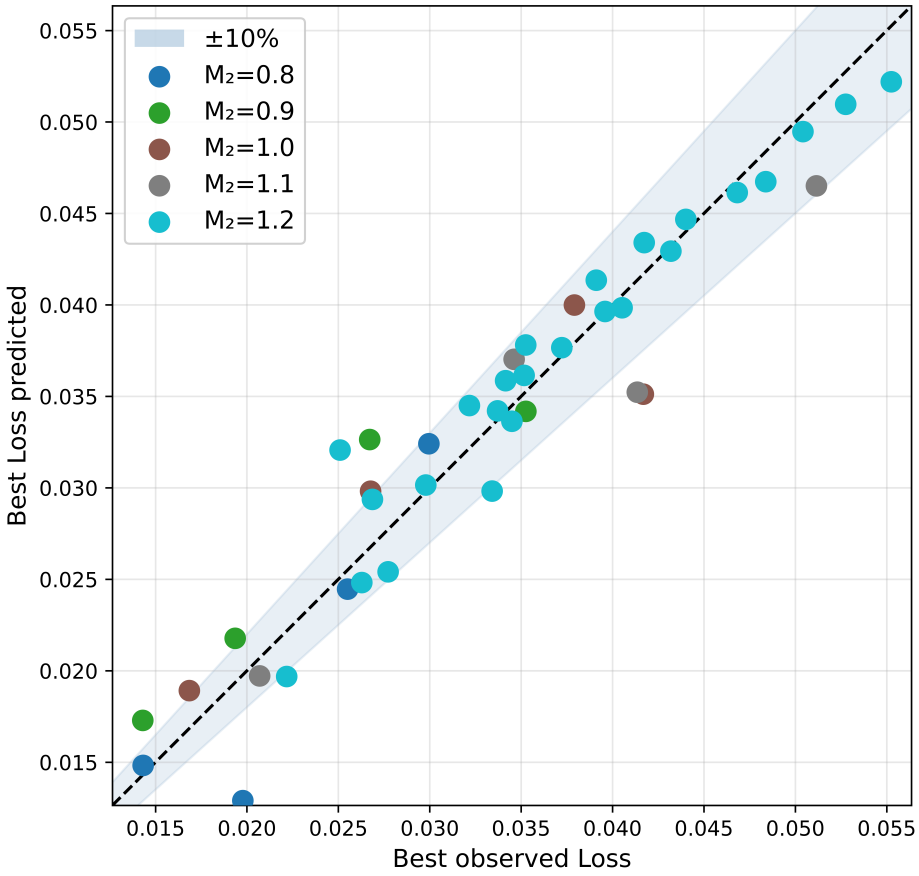


Figure 5.19: Parity plot of the polynomial loss model ($R^2 = 0.9203$, MAE $= 0.0023$).

5.3.6 Hyperparameter Sensitivity Study

UNDX Parameters: σ_ξ and σ_η

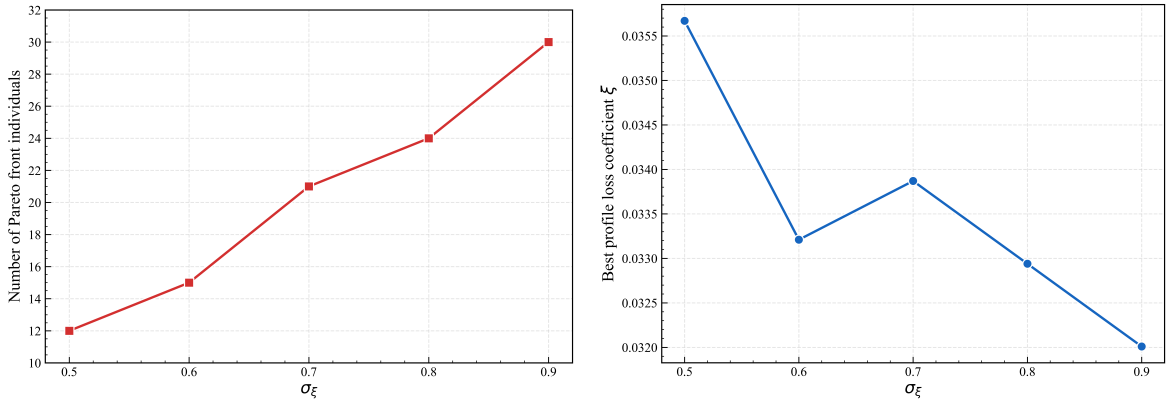
The UNDX crossover operator is governed by two parameters:

- σ_ξ: parametrizes the normal distribution along the principal axis connecting the two primary parents. A larger value produces offspring that explore more widely along the direction of the parent pair, promoting bolder moves in the parameter space.
- σ_η: parametrizes the distribution in the orthogonal directions. A smaller value

concentrates offspring close to the subspace spanned by the parents, while a larger value allows broader exploration in the transverse directions.

Sensitivity to σ_ξ

Five values were tested: $\sigma_\xi \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, with $\sigma_\eta = 0.30$ fixed, a population of 100 individuals, and 100 generations (10,000 CFD evaluations), starting from the same Generation 0 family. The results are shown in Figure 5.20.



Number of Pareto front individuals | Best ξ on the Pareto front

Figure 5.20: Sensitivity to σ_ξ. All runs share the same Generation 0 population, 100 individuals, 100 generations.

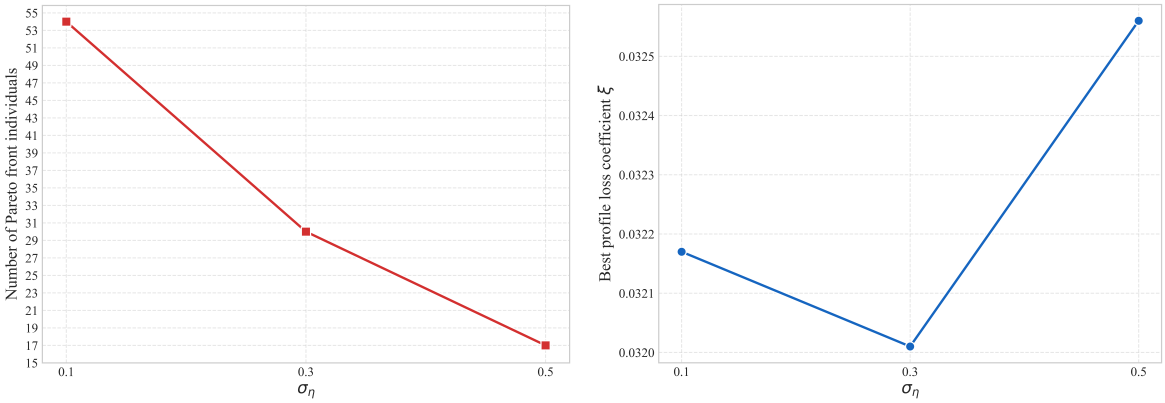
Both the number of non-dominated solutions and the best loss tend to improve with increasing σ_ξ, with $\sigma_\xi = 0.9$ yielding the largest and best-performing front. A larger σ_ξ encourages bolder exploration along the parent direction, allowing the optimizer to escape local clusters and populate a wider region of the Pareto front. The choice $\sigma_\xi = \mathbf{0.9}$ was therefore natural.

Sensitivity to σ_η

Three values were tested: $\sigma_\eta \in \{0.1, 0.30, 0.5\}$, with $\sigma_\xi = 0.9$ fixed and the same setup as above. The results are shown in Figure 5.21.

Here the trend is reversed: a **smaller** σ_η yields a significantly larger number of non-dominated solutions, while the loss ranges remain comparable across the three values. A small σ_η concentrates the offspring perturbations along the principal parent direction, limiting noise in the transverse directions and producing smoother, more coherent geometric variations between parent and offspring. Larger values of σ_η introduce broader transverse perturbations that tend to scatter the population without a corresponding improvement in objective space coverage.

The **final UNDX configuration** is therefore $\sigma_\xi = \mathbf{0.9}$ and $\sigma_\eta = \mathbf{0.1}$.



Number of Pareto front individuals | Best ξ on the Pareto front

Figure 5.21: Sensitivity to σ_η. All runs share the same Generation 0 population, 100 individuals, 100 generations.

Population Size and Number of Generations

Given a fixed budget of 10,000 CFD evaluations, the allocation between population size and number of generations represents a fundamental trade-off. A larger population explores the design space more broadly at each generation but allows fewer generational cycles for the genetic operators to refine the solutions; a smaller population undergoes more generations of selection and crossover, at the cost of reduced diversity in the initial pool. Three configurations were compared:

Table 5.3: Population and generation configurations tested at fixed 10,000 CFD evaluations.

Population size	Generations	Total evaluations
200	50	10,000
100	100	10,000
50	200	10,000

Hypervolume Indicator

To compare the convergence behaviour of the three configurations, the **hypervolume indicator** $\mathcal{HV}$ was adopted. The hypervolume is the volume of the objective space dominated by the current Pareto front and bounded by a reference point $\mathbf{r} = (\xi_{\text{ref}},\ (t/c_{ax})_{\text{ref}})$:

$$\mathcal{HV}(\mathcal{F}, \mathbf{r}) = \lambda\left(\bigcup_{\mathbf{f}\in\mathcal{F}} [\mathbf{f},\ \mathbf{r}]\right), \tag{5.11}$$

where λ denotes the **Lebesgue measure** (area in 2D) and $[\mathbf{f}, \mathbf{r}]$ is the hyperrectangle bounded by solution $\mathbf{f}$ and the reference point. A larger hypervolume indicates a Pareto front that is simultaneously closer to the ideal point and better distributed.

The reference point was set to $\mathbf{r} = (\mathbf{1.0}, \mathbf{0.0})$ in the normalized objective space, corresponding to the **worst admissible loss** and **zero loading**. The hypervolume was normalized by the area of the full reference rectangle to allow comparison across runs with different generation counts. Figure 5.22 shows the normalized hypervolume as a function of the number of generations for the three configurations, and Table 5.4 summarises the final Pareto front quality.

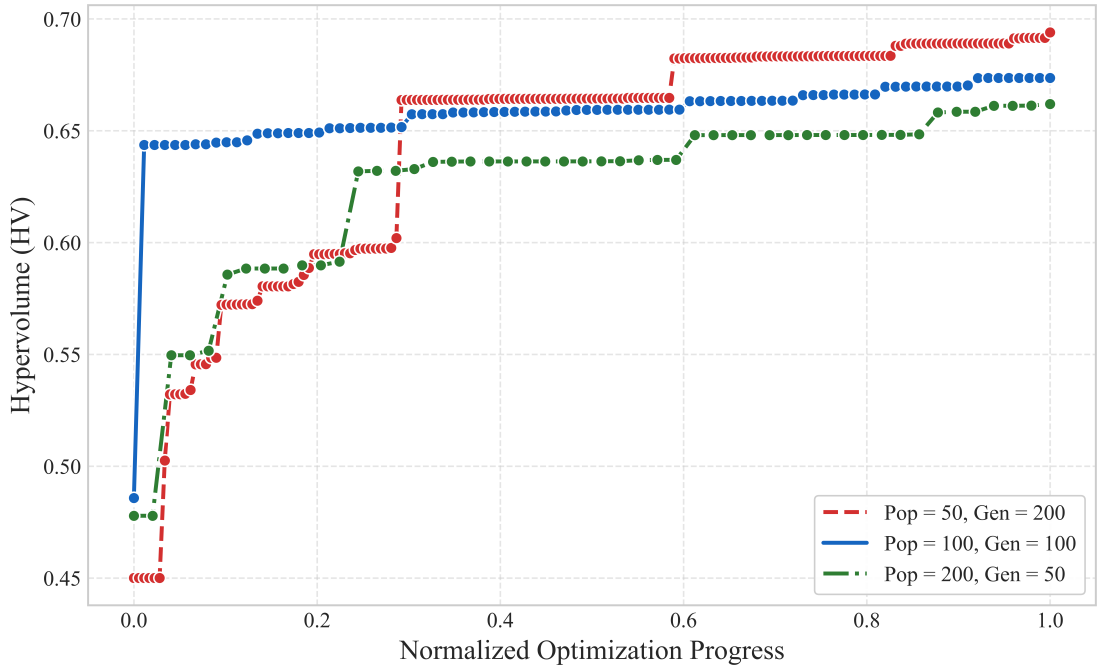


Figure 5.22: Hypervolume as a function of generation number for the three population/generation configurations at fixed 10,000 CFD evaluations.

Table 5.4: Summary of Pareto front quality for the three configurations.

Pop. size	Generations	Best ξ	Pareto front size
200	50	0.03189	10
100	100	0.03217	54
50	200	0.03221	73

The configuration with 200 individuals performs worst across all metrics: despite its broader initial exploration, the limited number of generational cycles is insufficient for the genetic operators to refine the solutions, resulting in a lower hypervolume, a less regular Pareto front, and fewer non-dominated individuals. The configurations

with 100 and 50 individuals achieve comparable results, but the **50-individual run** reaches a slightly higher final hypervolume with a more regular and better-populated front. With more generations available, the UNDX crossover has more opportunities to exploit the search directions established by the parent pairs, progressively refining the front beyond what a larger but less evolved population can achieve.

At fixed CFD budget, fewer individuals and more generations are therefore preferred. The question of whether 10,000 evaluations constitute a sufficient total budget is addressed in the following section.

5.3.7 Total CFD Budget

A classical rule of thumb for evolutionary algorithms suggests a minimum budget of approximately $1000 \times N_{\text{var}}$ function evaluations [28], which would indicate roughly 20,000 evaluations for this problem. This guideline, however, was established for general-purpose optimization with random initialization and is now considered conservative. Given the optimized Generation 0 initialization of Chapter 4, computational constraints, and the progressive tuning carried out in the preceding sections, **10,000 evaluations** were adopted as the working budget.

To verify whether this is sufficient, all four corner points C1–C4 at $M_2 = 1.2$ and TE = 3% were extended by 100 additional generations, reaching **15,000 CFD evaluations**. The hypervolume histories are shown in Figure 5.23 and the results are summarised in Table 5.5.

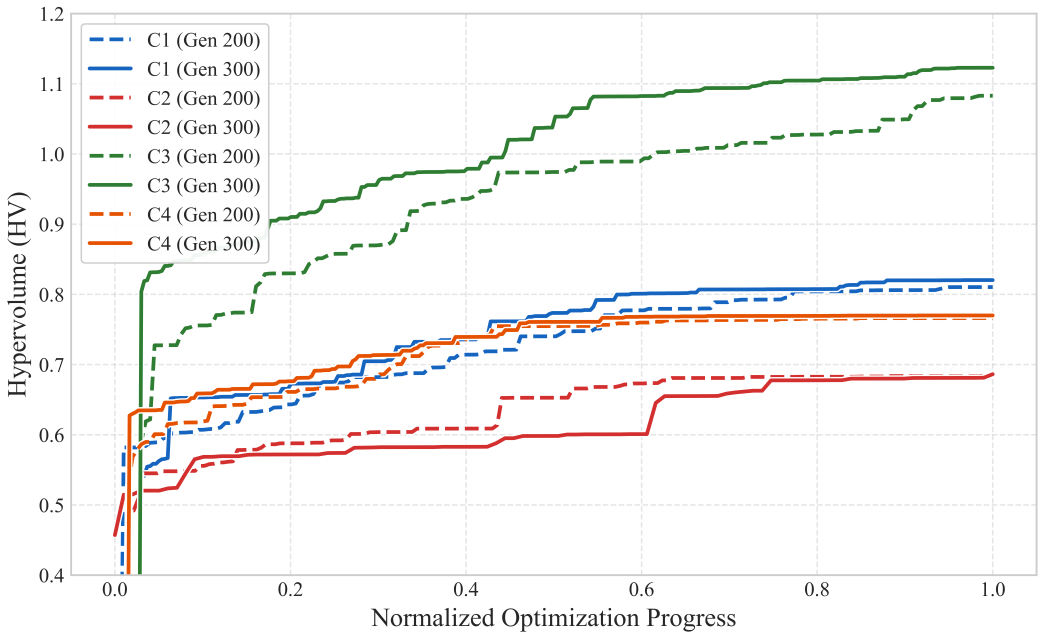


Figure 5.23: Hypervolume as a function of generation number for C1, C2, C3, and C4 at $M_2 = 1.2$.

Table 5.5: Pareto front quality at 200 and 300 generations for the four corner points ($M_2 = 1.2$, TE = 3%).

	Best ξ [%]		**Pareto front size**	
Case	200 gen	300 gen	200 gen	300 gen
C1	2.22	2.15	86	90
C2	3.18	3.24	73	79
C3	4.17	4.14	59	71
C4	5.52	5.44	110	176

The additional 100 generations yield a negligible improvement in hypervolume for three out of four cases, with only C3 approaching a 5% gain. Loss values are virtually identical between the two budgets. The Pareto front size does increase—most notably for C4, where a 60% growth is observed—but without a corresponding hypervolume gain, meaning the additional solutions populate already-covered regions of the front rather than extending its boundaries. The comparison of the Pareto fronts is shown in Figure 5.24.

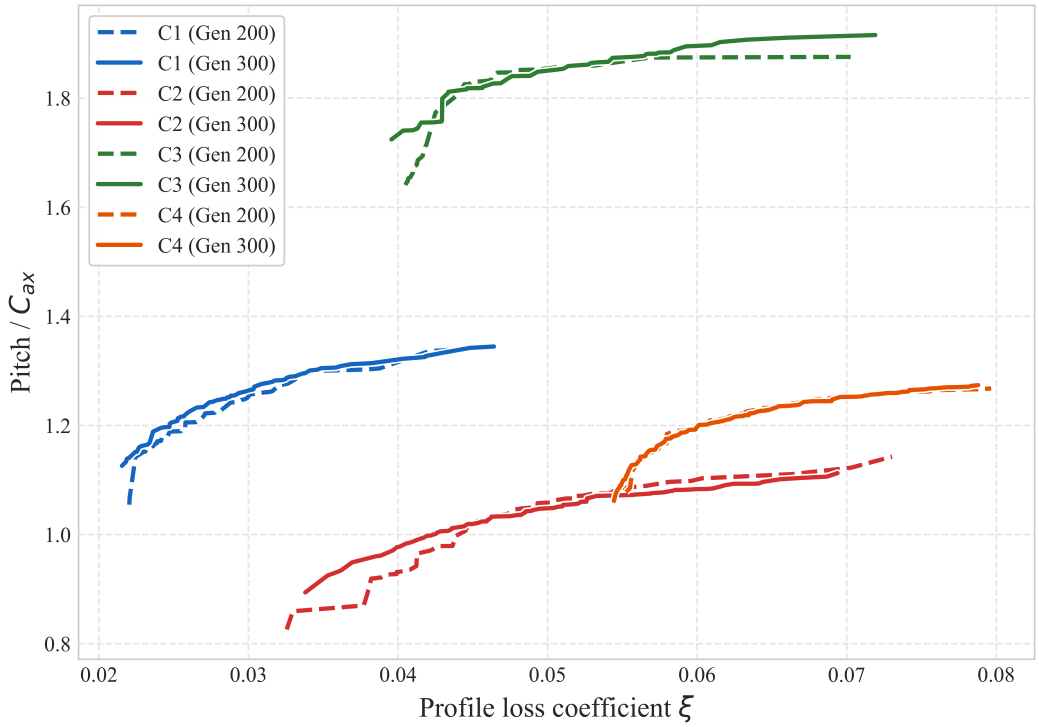


Figure 5.24: Comparison of the Pareto fronts at 200 and 300 generations for C1, C2, C3, and C4 at $M_2 = 1.2$ and TE thickness = 3%.

The marginal quality gain does not justify the additional computational cost. The full CFD evaluation pipeline—from `ParaBlade` geometry reconstruction to `MISES` simulation and post-processing—runs on a single CPU core per duty point and **requires on average 6–7 hours to complete 10,000 function evaluations** for standard cases. Duty points with a high ratio of non-converged individuals, such as **C2**, can require up to 11 hours. Extending the budget to 15,000 evaluations would add between 3 and 5 hours per duty point—a 30–50% increase in wall-clock time—for a gain that is essentially limited to a denser but not wider Pareto front.

A budget of 10,000 CFD evaluations (50 individuals, 200 generations) was therefore confirmed as the final configuration for the full optimization campaign across all 125 duty points of the design space.

Table 5.6: Final NSGA-II configuration adopted for the full optimization campaign.

Category	**Parameter**	**Value**
Crossover operator	Operator	UNDX
	σ_ξ	0.9
	σ_η	0.1
Population	Population size	50
	Generations	200
	Total CFD evaluations	10,000
Objectives	Minimize	ξ
	Maximize	t/c_{ax}
Constraints	Angle strategy	Hard constraint
	Angle tolerance ϵ	0.5°
	Loss constraint	$\xi_{\max}$ from regression

Chapter 6

The Surrogate Model

The optimization campaign described in Chapter 5 produced, for each of the 125 duty points in the design space, a Pareto front of non-dominated blade geometries representing the achievable trade-off between aerodynamic efficiency and blade loading. The collection of these fronts constitutes the raw output of the entire computational pipeline: **125 multi-objective optimization runs**, each consuming 10,000 CFD evaluations, for a total of **1.25 million `MISES` simulations**. The question that naturally follows is how to exploit this information to build a surrogate model capable of generating near-optimal blade geometries for any prescribed operating condition, without requiring a new optimization.

This chapter addresses that question in full. Section 6.1 discusses a fundamental design choice that precedes any machine learning consideration: how to transform the discrete, multi-valued Pareto fronts into a structured training dataset. Section 6.2 describes the automated pipeline used to build the dataset from the raw optimization logs. Section 6.3 states the regression problem formally and defines the input–output structure of the model. Sections 6.4 and 6.5 present the network architecture and the hyperparameter optimisation procedure. Finally, Section 6.6 reports the training results and discusses the prediction accuracy across all geometric parameters.

6.1 From Optimisation Results to a Training Dataset

The output of the NSGA-II optimizer is, for each duty point, a discrete collection of **Pareto-optimal blade geometries**. Converting this collection into a training dataset is not a trivial step. In a **single-objective** setting, each duty point yields one optimal solution, and the surrogate faces a well-posed, single-valued regression problem. In the **multi-objective** setting, a single duty point is associated with an entire front of solutions, each representing a different and equally valid compromise between loss and loading. A surrogate trained naively on this data would be presented with multiple,

geometrically distinct outputs for the same input and would respond by averaging conflicting geometries—a physically meaningless artifact.

Several strategies are possible to resolve this ambiguity. One could select a **single representative solution** per front—the minimum-loss point, the maximum-loading point, or the knee—thereby reducing the problem to a single-valued mapping at the cost of discarding most of the generated data. Alternatively, one could define a **near-Pareto band** by introducing a tolerance d_{Pareto} on the distance to the front, and pair it with additional descriptors of geometric diversity, recovering a richer but more complex training structure in the spirit of the ε-dominance framework of Laumanns et al. [35].

6.1.1 Parametrisation of the Pareto Front via Arc-Length

The strategy adopted in this work is to parametrize each Pareto front by a **scalar coordinate** $p \in [0, 1]$, defined as the **normalized arc length** along the front in the objective space. Given a front of N non-dominated solutions sorted in order of increasing pitch ($L_1 \leq L_2 \leq \cdots \leq L_N$), the arc-length up to solution i is:

$$s_i = \sum_{k=2}^{i} \sqrt{\left(\frac{\xi_k - \xi_{k-1}}{\xi_{\max} - \xi_{\min}}\right)^2 + \left(\frac{L_k - L_{k-1}}{L_{\max} - L_{\min}}\right)^2}, \tag{6.1}$$

where $L_k = t/c_{ax}$ is the loading objective of solution k, and both differences are normalized by the respective objective ranges to give equal weight to the two axes. The coordinate $p_i = s_i/s_N$ maps the front onto $[0, 1]$, with:

- $\mathbf{p = 0}$ at the **minimum-loss** extreme;
- $\mathbf{p = 1}$ at the **maximum-loading** extreme.

Since the second objective is the pitch directly, p is in practice **equivalent to a normalized pitch**: increasing p corresponds to selecting a blade with larger pitch. This parametrization transforms the multi-objective dataset into a **single-valued regression problem**, with p promoted to an **additional input** of the model alongside the aerodynamic duty:

$$(\alpha_1,\, \alpha_2,\, M_2,\, p) \;\longrightarrow\; \mathbf{x}_{\text{blade}} \in \mathbb{R}^{21}. \tag{6.2}$$

At inference time, the designer prescribes not only the aerodynamic duty but also the **desired position along the Pareto front**, retaining explicit and continuous control over the loss–loading trade-off without any additional CFD evaluation. The 125 optimized fronts constitute, however, a general-purpose dataset: the training strategy described here is one specific choice among several, and designers with different priorities

may adopt alternative formulations—including the extensions outlined above—without modifying the underlying optimization data.

6.2 Dataset Construction Pipeline

With the arc-length coordinate p defined, the next step is to extract a training dataset from the 125 Pareto fronts that is both representative of the full $p \in [0, 1]$ interval and free from the sampling bias inherent in the raw optimization output. The NSGA-II algorithm evaluates thousands of individuals per duty point, but their distribution in objective space is far from uniform: by design, the search concentrates progressively around the non-dominated region, so that later generations are dominated by individuals clustered near the Pareto front, with only sparse coverage elsewhere. Training directly on this unfiltered collection would therefore bias the surrogate towards the most populated regions of the front, at the expense of accuracy in sparser areas.

Uniform binning

To correct this imbalance, the training dataset is assembled by **uniform binning** on p. The interval $[0, 1]$ is divided into **100 equal-width bins**, and a single representative blade is selected from each bin. The selection follows a strict priority:

- **Non-empty bin**: if one or more Pareto-optimal solutions fall within the bin, the one whose p value is closest to the bin center is selected.
- **Empty bin**: if no Pareto solution falls in the interval, the exact point on the Pareto curve at the bin center is obtained by linear interpolation of the front in the normalized objective space, and the nearest near-Pareto design among all evaluated individuals is selected as a representative.

A global uniqueness constraint prevents the same individual from being assigned to multiple bins.

This strategy guarantees that the 100 samples extracted per duty point are **uniformly distributed along the front**, with each bin covered by either a true Pareto solution or a near-Pareto design whose geometric parameters are representative of that region of the trade-off. Since p corresponds to normalized pitch, the selected blades are ordered by **increasing pitch** by construction. For true Pareto solutions, this ordering is also consistent with monotonically increasing loss. For near-Pareto solutions, however, this strict consistency is not guaranteed: small **oscillations in loss along** p are accepted, as the selection criterion prioritizes proximity in pitch space rather than loss ordering. The theoretical justification for treating near-Pareto solutions as valid training samples is provided by the ε-dominance framework [35]: a design at small distance from the

front can be considered ε-non-dominated, and its geometry carries valid information about the mapping from duty and loading position to near-optimal blade shapes.

6.2.1 Dataset composition

With 100 bins per duty point and 125 duty points, the final training dataset contains **12,500 samples**. Each sample carries the input variables $(\alpha_1, \alpha_2, M_2, p)$, the 21 `ParaBlade` geometric parameters, and the distance d_{Pareto} to the reconstructed front. The quality of the binning depends directly on how many individuals are available on the Pareto front for each duty point. Table 6.1 summarizes the key statistics across the 125 runs, and the full distribution of Pareto-filled bins is shown in Figure 6.1.

Table 6.1: Binning statistics across the 125 duty points.

Quantity	Mean ± Std	Min / Max
Pareto front size (individuals)	56 ± 22	20 / 129
Bins filled by true Pareto solutions (out of 100)	45.2 ± 13	22 / 79

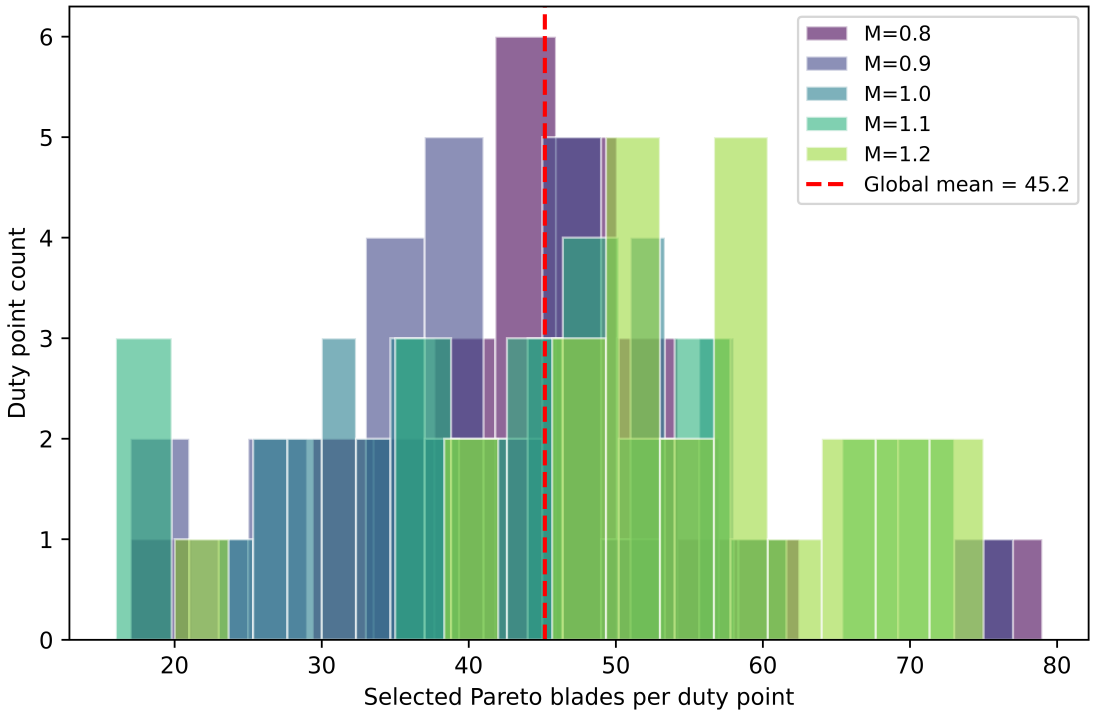


Figure 6.1: Distribution of the number of bins filled by a true Pareto solution across the 125 duty points.

The reliability of the near-Pareto selection depends on how far from the front the algorithm must reach to fill the empty bins. The distance is measured in the **normalized**

objective space, where both the loss and the loading objective are linearly scaled to $[0, 1]$ over the range observed across the full campaign, so that neither axis dominates the metric. For each duty point, the **maximum distance** reached to complete all 100 bins is reported in Figure 6.2.

In the vast majority of cases this distance remains well below 5%, confirming that the near-Pareto representatives are genuinely close to the front and not arbitrary designs pulled from the interior of the feasible space. Only **two duty points** require reaching beyond the 5% threshold, corresponding to the sparsest and most irregular fronts observed in the campaign.

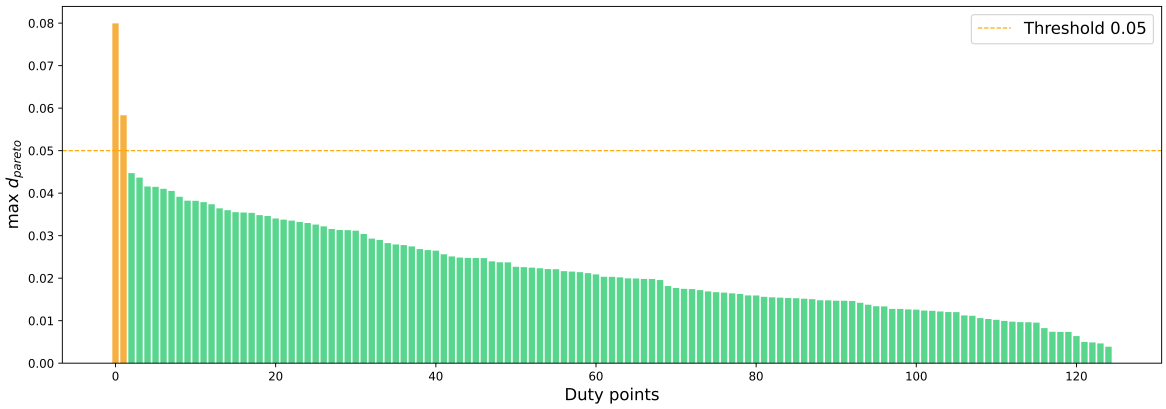


Figure 6.2: Maximum normalized distance from the Pareto front reached to complete all 100 bins, for each of the 125 duty points.

To give a concrete sense of what the binning produces in practice, Figures 6.3 and 6.4 show the result for the:

- **best case**: the duty point with the smallest maximum distance covered;
- **worst case**: the duty point requiring the largest excursion from the front to complete the bins.

In the best case, all selected blades lie essentially on the Pareto curve, and the 100 samples provide a dense and faithful representation of the full trade-off. In the worst case, a larger fraction of near-Pareto designs is recruited, yet the selected individuals remain clustered in a narrow band around the front.

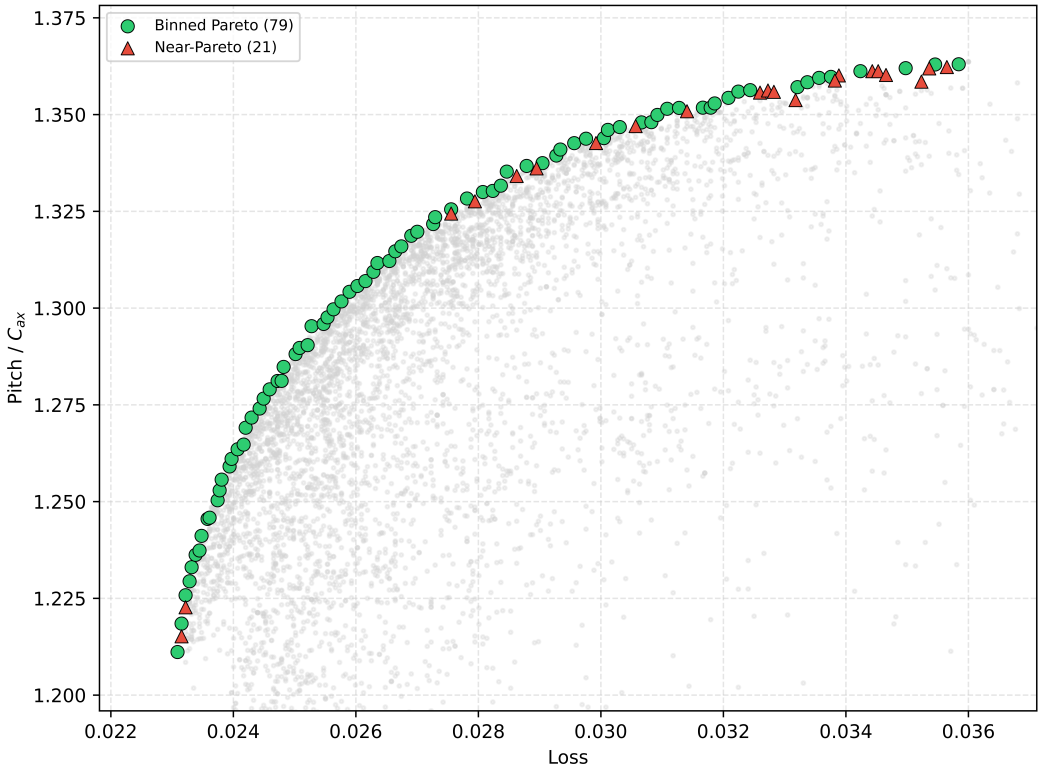


Figure 6.3: Binning result for the best case duty point ($A8$ at $M_2 = 0.8$).

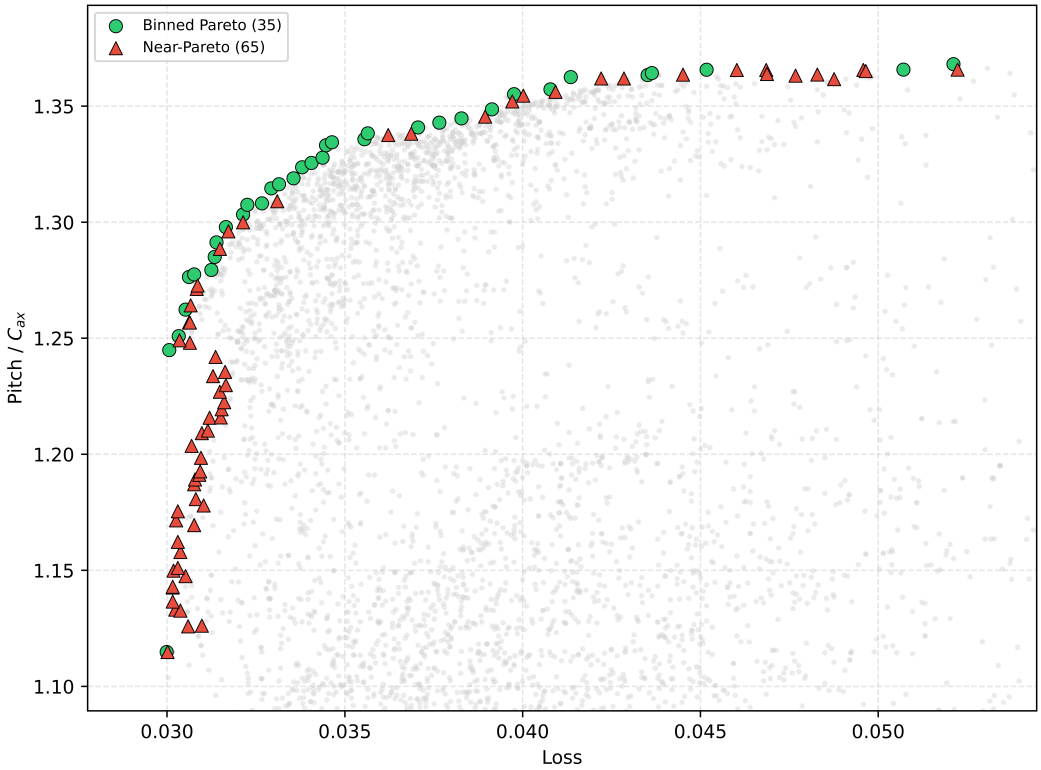


Figure 6.4: Binning result for the worst case duty point ($A5$ at $M_2 = 1.1$).

6.3 Problem Formulation

The surrogate model is formulated as a **multi-output regression problem**. The input vector is the four-dimensional duty-loading descriptor:

$$\mathbf{u} = (\alpha_1,\ \alpha_2,\ M_2,\ p) \in \mathbb{R}^4, \tag{6.3}$$

where α_1 and α_2 are the inlet and target outlet flow angles, M_2 is the isentropic outlet Mach number, and $p \in [0,1]$ is the Pareto arc-length coordinate defined in Section 6.1.1.

The output vector contains **21 geometric parameters**: the 20 `ParaBlade` shape parameters plus the pitch-to-axial-chord ratio t/c_{ax},

$$\mathbf{x} = \Big(\text{stagger},\ \theta_{\text{in}},\ \theta_{\text{out}},\ r_{\text{in}},\ d_{\text{in}},\ d_{\text{out}},\ t/c_{ax},\ t_{u,1}, \ldots, t_{u,8},\ t_{l,1}, \ldots, t_{l,6}\Big) \in \mathbb{R}^{21}, \tag{6.4}$$

where $t_{u,i}$ and $t_{l,j}$ are the upper and lower surface thickness control points respectively.

The trailing-edge radius `radius_out` and the Reynolds number $\text{Re}_2 = 10^6$ are excluded from both the input and the output of the model, as both are held constant throughout the entire campaign. Their values are fixed directly at inference time before geometry reconstruction and CFD evaluation.

6.4 Network Architecture

The surrogate model is implemented as a **fully connected multi-layer perceptron** (MLP) using the `PyTorch` framework [36]. The architecture follows a standard feed-forward structure in which the input vector $\mathbf{u} \in \mathbb{R}^4$ is passed through a sequence of L hidden layers, each consisting of a linear transformation followed by a nonlinear activation and a dropout unit, before reaching the linear output layer that produces the 21 geometric parameters:

$$\hat{\mathbf{x}} = W_L \circ \phi \circ W_{L-1} \circ \cdots \circ \phi \circ W_1 \circ \mathbf{u}, \tag{6.5}$$

where W_k denotes the affine transformation of the k-th layer and ϕ is the element-wise activation function, whose specific form is determined by the hyperparameter optimization described in Section 6.5. The final layer carries no activation, as no non-negativity or boundedness constraint is imposed on the geometric parameters at the model level. A schematic of the architecture is shown in Figure 6.5.

Training objective

The model is trained by minimizing the **mean squared error** (MSE) between the predicted and true output vectors, computed in the standardized space:

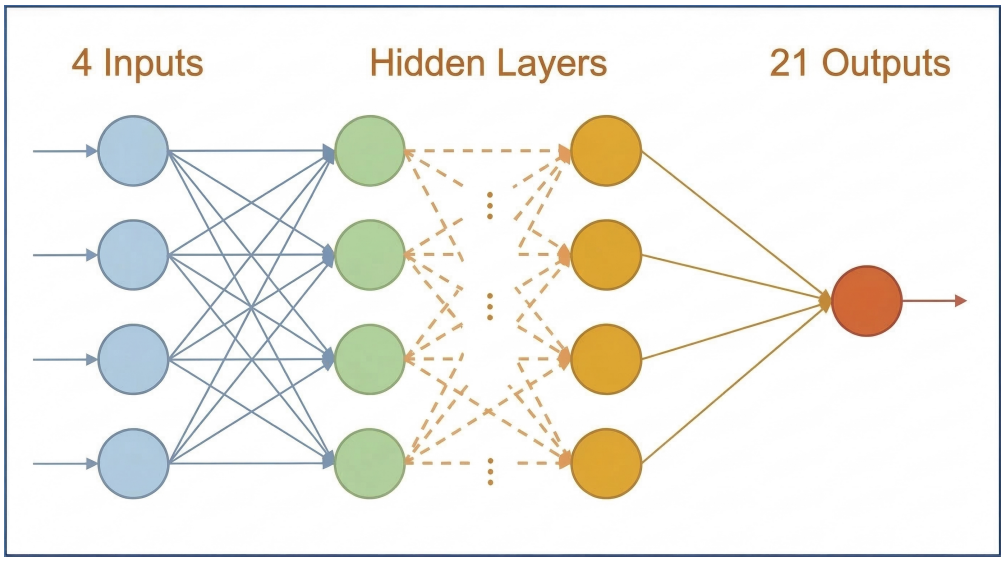


Figure 6.5: Schematic of the MLP surrogate model.

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left\| \tilde{\mathbf{x}}_i - \hat{\tilde{\mathbf{x}}}_i \right\|^2, \tag{6.6}$$

where $\tilde{\mathbf{x}}_i$ and $\hat{\tilde{\mathbf{x}}}_i$ are the standardized true and predicted output vectors for sample i, and N is the number of training samples. Operating in the standardized space ensures that all 21 output dimensions contribute equally to the loss, regardless of their physical scale.

Preprocessing

All input and output variables are standardized to **zero mean and unit variance** using a `StandardScaler` fitted exclusively on the training set. The transformation is applied independently to each dimension, and the inverse transformation is applied at inference time to recover physical units. This preprocessing is particularly important given the heterogeneous physical scales of the 21 output parameters: camberline angles span tens of degrees, while thickness control points are of order 10^{-2} in normalized chord units.

6.5 Hyperparameter Optimization

The architecture of the MLP—number of hidden layers, hidden layer width, dropout rate, learning rate, and activation function—was optimized using the `Optuna` framework [37], a hyperparameter optimization library that implements the **Tree-structured Parzen Estimator (TPE)** sampler. Unlike grid search or random search, TPE builds a probabilistic model of the objective function and uses it to propose trials that are more

likely to improve upon the current best, making it significantly more sample-efficient for high-dimensional search spaces.

6.5.1 Search Space

A total of **100 trials** were executed, each evaluating a full training run on an 85%/15% training–validation split. The search space is summarized in Table 6.2.

Table 6.2: Hyperparameter search space for the `Optuna` optimization.

Hyperparameter	Type	Range / Choices
Number of hidden layers	Integer	2 – 5
Hidden layer width	Categorical	{128, 256, 512}
Dropout rate	Continuous	[0.0, 0.3]
Learning rate	Log-continuous	$[10^{-4}, 10^{-2}]$
Activation function	Categorical	{ReLU, Tanh, SiLU}

To reduce the computational cost of the search, a `MedianPruner` was employed: trials whose intermediate validation loss exceeded the median of all completed trials at the same epoch were terminated early, avoiding the waste of compute on configurations that were clearly underperforming.

Training loop

Each trial was trained using the `Adam` optimizer [38] with the MSE loss defined in Equation 6.6. `Adam` was selected for its adaptive learning rate mechanism, which scales the update for each parameter individually based on the history of its gradients, making it robust to the heterogeneous scales of the 21 output dimensions. The learning rate was further reduced on plateau via `ReduceLROnPlateau` with a reduction factor of 0.5 and patience of 10 epochs: whenever the validation loss stops improving, the learning rate is halved, allowing the optimizer to make finer adjustments as it approaches convergence. **Early stopping** with patience of 50 epochs was applied, monitoring the validation loss, to prevent overfitting. The batch size was fixed at 256 throughout all trials.

Final training

Once the best trial was identified, a **final training run** was performed on the full training set using the optimal configuration, with the epoch budget and early stopping patience **doubled** to 1000 and 100 epochs respectively. This allows the model to converge more completely than during the search phase, where the tighter stopping criterion was necessary to keep each trial computationally affordable. The final run

reached early stopping at **epoch 257**, confirming that the doubled budget was sufficient for full convergence.

6.5.2 Optimal Configuration

The hyperparameter configuration that minimized the validation MSE across all 100 trials is reported in Table 6.3.

Table 6.3: Optimal hyperparameter configuration identified by `Optuna`.

Hyperparameter	Value
Number of hidden layers	4
Hidden layer width	256
Dropout rate	0.013
Learning rate	0.002
Activation function	SiLU

The resulting model is a four-hidden-layer MLP with 256 units per layer. The **SiLU** activation function, defined as:

$$\mathrm{SiLU}(z) = z \cdot \sigma(z) = \frac{z}{1 + e^{-z}}, \tag{6.7}$$

emerged as the best-performing activation, outperforming both ReLU and Tanh across the search. Its smoothness and non-monotonicity make it well suited to the present regression task, where the output surface varies continuously across the duty-loading space. The **near-zero dropout rate** ($\delta = 0.013$) confirms that the training set of approximately 12,500 samples is sufficiently large relative to the model capacity that strong regularization is not required.

6.6 Training Results

Learning curve

During training, the dataset is split into a **training set** (85%), used at each epoch to update the network weights via backpropagation, and a **validation set** (15%), held out and never used for weight updates. The validation loss is monitored at every epoch to assess generalization to unseen data and to trigger early stopping.

Figure 6.6 shows the evolution of both losses over the **257 epochs** of the final training run. Both drop sharply in the **first 50 epochs**, then continue to decrease more gradually until early stopping is triggered. The best validation loss is reached at **epoch 157**, with final values of **0.0587** (training) and **0.0703** (validation) in standardized

space, corresponding to a **gap of 16.5%**. The validation curve never rises, confirming that **no overfitting** occurs and that the model generalizes correctly to unseen data.

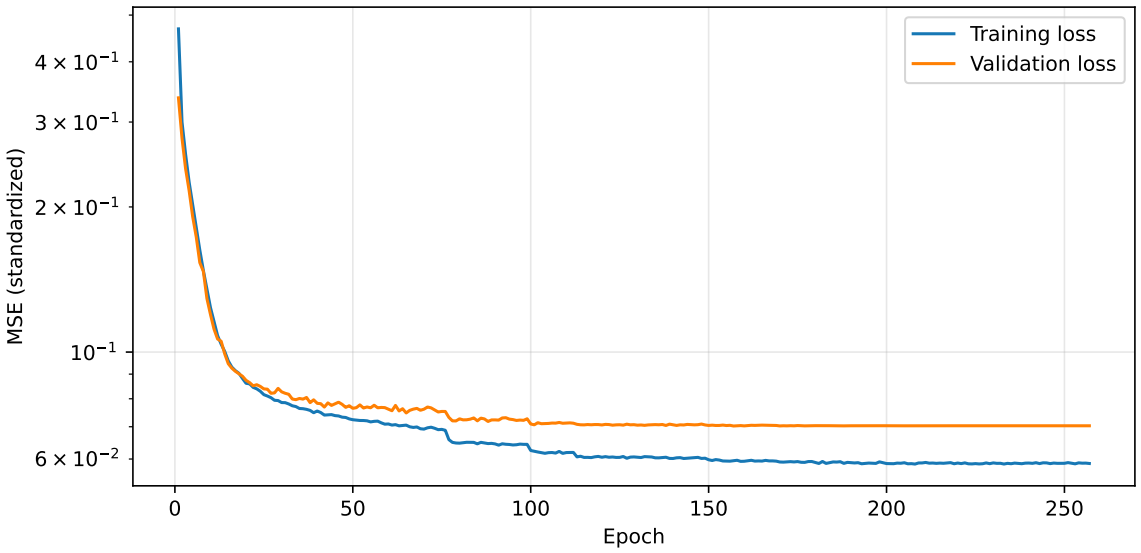


Figure 6.6: Training and validation MSE as a function of epoch, in logarithmic scale. Early stopping is triggered at epoch 257; the best validation loss is achieved at epoch 157.

6.6.1 Prediction Accuracy

The final model was evaluated on a held-out test set comprising duty points not used during training or validation, achieving a **mean $\mathbf{R^2}$ of 0.9428** across all 21 outputs. The results are organized by parameter group in Tables 6.4–6.7.

The **pitch-to-axial-chord ratio** t/c_{ax}, which constitutes the second optimization objective alongside the aerodynamic loss, is reported first in Table 6.4. Its prediction accuracy is very high, confirming that the surrogate has correctly learned the relationship between the duty-loading position and the blade spacing.

Table 6.4: Prediction accuracy — pitch-to-axial-chord ratio.

Parameter	RMSE	MAE	R^2	RMSE/mean
t/c_{ax}	7.97×10^{-3}	6.11×10^{-3}	0.9993	0.57%

The **camberline parameters**—stagger, θ_{in}, θ_{out}, d_{in}, d_{out}—are reported in Table 6.5. All achieve $R^2 > 0.98$, with angular RMSE well below 1°. The slightly higher normalized error on θ_{in} (2.04%) reflects the wider range that this parameter spans across the design space, rather than a genuine loss of accuracy.

Table 6.5: Prediction accuracy — camberline parameters.

Parameter	RMSE	MAE	R^2	RMSE/mean
stagger	0.288°	0.214°	0.9993	0.72%
θ_{in}	0.620°	0.438°	0.9989	2.04%
θ_{out}	0.344°	0.241°	0.9985	0.55%
d_{in}	5.77×10^{-3}	3.93×10^{-3}	0.9976	0.95%
d_{out}	1.07×10^{-2}	5.57×10^{-3}	0.9841	1.64%

The **suction-side thickness control points** $t_{u,1}$–$t_{u,8}$, together with the **leading-edge radius** r_{in} which governs the curvature shared between the two surfaces, are reported in Table 6.6. A clear trend is visible: R^2 **decreases progressively** moving from the leading edge towards the trailing edge, from **0.987** at $t_{u,1}$ down to **0.923** at $t_{u,8}$. This reflects the increasing geometric freedom that the optimizer exercises in the aft region of the suction side, where the aerodynamic constraints are less tight than near the leading edge.

The **leading-edge radius** $\mathbf{r}_{\text{in}}$ achieves $R^2 = 0.885$, the lowest among this group. Unlike the thickness control points, which belong exclusively to one surface, r_{in} controls the curvature of the **leading edge shared between the suction and pressure sides**: its variability therefore reflects not only the suction-side aerodynamic constraints but also the higher scatter of the pressure-side parameters, which are driven by weaker aerodynamic selection as discussed below. This dual influence explains why r_{in} deviates from the otherwise monotone trend of the suction-side group.

Table 6.6: Prediction accuracy — suction-side parameters.

Parameter	RMSE	MAE	R^2	RMSE/mean
r_{in}	3.57×10^{-3}	2.57×10^{-3}	0.8849	10.53%
$t_{u,1}$	8.84×10^{-3}	5.36×10^{-3}	0.9875	6.31%
$t_{u,2}$	5.87×10^{-3}	3.70×10^{-3}	0.9963	2.27%
$t_{u,3}$	4.36×10^{-3}	2.80×10^{-3}	0.9958	1.82%
$t_{u,4}$	5.43×10^{-3}	3.67×10^{-3}	0.9964	3.45%
$t_{u,5}$	3.64×10^{-3}	2.41×10^{-3}	0.9839	8.05%
$t_{u,6}$	3.91×10^{-3}	2.48×10^{-3}	0.9736	10.51%
$t_{u,7}$	3.81×10^{-3}	2.58×10^{-3}	0.9570	7.53%
$t_{u,8}$	2.93×10^{-3}	1.99×10^{-3}	0.9230	11.06%

The **pressure-side thickness parameters** $t_{l,1}$–$t_{l,6}$ are reported in Table 6.7 and show the **lowest prediction accuracy** of the entire output set ($R^2 = 0.78$–0.89,

normalized RMSE up to **23.7%**). The absolute RMSE values remain small (of order 10^{-3}–10^{-2}), but the higher normalized errors reflect the **intrinsically greater variability** of pressure-side thickness along the Pareto front.

In transonic turbine cascades, aerodynamic performance is primarily governed by the **suction-side boundary layer and shock structure**; the pressure side plays a **secondary aerodynamic role**, and the relationship between pressure-side geometry and objective values is consequently weak. Even NSGA-II, which progressively learns to associate geometric changes with performance improvements through crossover and selection, **struggles to identify consistent patterns** on these dimensions: individuals with very different pressure-side shapes produce nearly identical loss and loading values, so the optimizer cannot distinguish between them and the Pareto front retains a **wide geometric scatter** in these parameters. This is a **fundamental characteristic of the design space**, not a limitation of the model, and cannot be resolved by increasing the dataset size.

Table 6.7: Prediction accuracy — pressure-side thickness parameters.

Parameter	RMSE	MAE	R^2	RMSE/mean
$t_{l,1}$	1.62×10^{-2}	1.09×10^{-2}	0.8725	14.51%
$t_{l,2}$	1.51×10^{-2}	1.00×10^{-2}	0.8793	11.66%
$t_{l,3}$	2.19×10^{-2}	1.38×10^{-2}	0.8931	15.11%
$t_{l,4}$	1.33×10^{-2}	8.38×10^{-3}	0.8337	23.71%
$t_{l,5}$	6.40×10^{-3}	4.11×10^{-3}	0.8615	22.62%
$t_{l,6}$	2.63×10^{-3}	1.70×10^{-3}	0.7778	17.85%

Chapter 7

Results and Discussion

This chapter presents the results of the complete data-driven pipeline developed in this work. The pipeline is first applied to three of the four **corner points** — C2, C3, and C4 — which represent the most aerodynamically challenging duty conditions in the design space and collectively cover the full range of behaviours observed. The analysis is then extended to **unseen duty points** — operating conditions never encountered during training — to demonstrate the generalization capability of the model across the transonic design space.

Section 7.1 describes the inference pipeline. Section 7.2 presents the results on the selected corner points, including the surrogate predictions in the objective space and a physical study of how blade geometry and loading distribution vary with M_2 and p. Section 7.3 assesses the outlet flow angle accuracy across both training and unseen conditions. Finally, Section 7.4 summarizes the contributions, discusses the limitations of the approach, and outlines directions for future work.

7.1 Inference Pipeline

Once trained, the surrogate model is deployed within a fully automated inference pipeline that transforms a prescribed aerodynamic duty into a complete, `MISES`-simulated blade. The pipeline accepts four scalar inputs — the **inlet flow angle** α_1, the **target outlet flow angle** $\alpha_{2,\text{target}}$, the **isentropic exit Mach number** M_2, and the **Pareto position parameter** $p \in [0, 1]$ — and returns the aerodynamic performance of the resulting blade as evaluated by the flow solver. The four stages are described below.

Stage 1 — Surrogate inference.
The input vector (α_1, $\alpha_{2,\text{target}}$, M_2, p) is standardized using the scaler fitted on the training set and passed through the neural network. The 21 raw outputs are back-transformed to physical space by the corresponding output scaler, yielding the complete

set of **21 `ParaBlade` design parameters**: the camberline descriptors (stagger, θ_{in}, θ_{out}, d_{in}, d_{out}), the leading-edge radius r_{in}, the pitch-to-axial-chord ratio t/c_{ax}, and the suction- and pressure-side thickness control points. This step requires a single forward pass through the network and is essentially instantaneous.

Stage 2 — Blade geometry generation and trailing-edge treatment.
The predicted parameters are passed to `MakeBlade`, which reconstructs the full 2D blade profile as a set of surface coordinates. As discussed in Chapter 3, `ParaBlade` produces a **closed trailing edge**, whereas `MISES` requires an **open trailing edge** to apply its wake model correctly. The **blade pitch** is recovered as $t = (t/c_{ax}) \cdot c_{ax}$ (with $c_{ax} = 1$) and is used to define the cascade periodicity.

Stage 3 — Reynolds number scaling.
`MISES` requires the **inlet Reynolds number** Re_1 as a boundary condition, whereas the surrogate was trained at a fixed **exit Reynolds number** $Re_2 = 10^6$. The inlet value is computed from Re_2 via a one-dimensional isentropic analysis combined with axial mass-flow conservation. Given the inlet and outlet flow angles and M_2, the inlet thermodynamic state is recovered iteratively, and Re_1 is obtained by scaling Re_2 by the ratio of inlet to outlet momentum per unit area:

$$Re_1 = Re_2 \cdot \frac{\rho_1 V_1}{\rho_2 V_2}. \tag{7.1}$$

Stage 4 — MISES simulation and post-processing.
The blade geometry, cascade pitch, and aerodynamic boundary conditions (M_2, Re_1) are passed to `MISES`, which performs a fully viscous coupled Euler/boundary-layer simulation. Upon convergence, two quantities are extracted:

- the **kinetic energy loss coefficient** ξ_{kin}, used as the primary aerodynamic performance metric;
- the **simulated outlet flow angle** $\alpha_{2,\text{sim}}$, extracted from the mass-averaged exit flow direction.

The quantity $\alpha_{2,\text{sim}}$ is the angle that the blade generated by the surrogate *actually produces* under the prescribed operating conditions. It is not imposed as a constraint by the network: the surrogate predicts a geometry associated with $\alpha_{2,\text{target}}$, but the aerodynamic response is determined entirely by `MISES`. The discrepancy

$$\Delta\alpha_2 = \alpha_{2,\text{sim}} - \alpha_{2,\text{target}} \tag{7.2}$$

measures the **end-to-end accuracy** of the pipeline in satisfying the prescribed duty, and is one of the central metrics assessed in the following sections.

7.2 Results on Corner Points

The four **corner points** C1–C4, corresponding to the extreme combinations of inlet and outlet flow angles in the design space, have been shown throughout this work to be the most aerodynamically challenging duty conditions. Three of them — **C2**, **C3**, and **C4** — are selected here to present the surrogate results, as they collectively cover the full range of behaviours observed.

7.2.1 Surrogate Predictions in the Objective Space

As discussed in Chapter 6, the training dataset was built using a binning strategy that includes not only true Pareto-optimal blades but also **near-Pareto dominated blades** to fill empty bins. The surrogate therefore does not learn the true Pareto front, but an approximation of it. In this section, the surrogate is queried at a discrete set of p values, each predicted blade is re-simulated with `MISES`, and the resulting $(\xi_{\text{kin}},\, t/c_{ax})$ pairs are plotted in the objective space.

Figure 7.1 shows the surrogate predictions for **C4** at all five training Mach numbers, illustrating how the loss–pitch trade-off evolves with M_2 at fixed flow angles.

These predictions do not form exact Pareto fronts, and two types of imperfection are observable. First, at the **low-p end**, dominated solutions occasionally appear — blades with simultaneously higher loss and lower pitch than neighbouring solutions — due to the presence of near-Pareto dominated blades in the training bins at the extremes of the front. Second, at the **high-p end**, particularly after the **knee** of the curve where many solutions cluster in a narrow pitch range, small pitch inversions are visible. In this region the arc-length parametrization assigns very different p values to blades with nearly identical pitch, which can produce mild non-monotonicities.

Both imperfections can be reduced by improving the resolution and quality of the optimization output, so as to minimize the number of near-Pareto blades used in training. On a single duty point this is straightforward; defining universally valid criteria across all 125 duty points is however prohibitive, and this pipeline is primarily designed as a **preliminary design tool**.

Figures 7.2–7.4 compare the surrogate predictions against the **original training Pareto fronts** from the NSGA-II logs, for C4 at $M_2 = 1.2$, C3 at $M_2 = 0.8$, and C2 at $M_2 = 1.0$.

For **C4** (Figure 7.2) and **C3** (Figure 7.3), the surrogate predictions **extend beyond the original training Pareto**, with blades confirmed by `MISES` to achieve lower

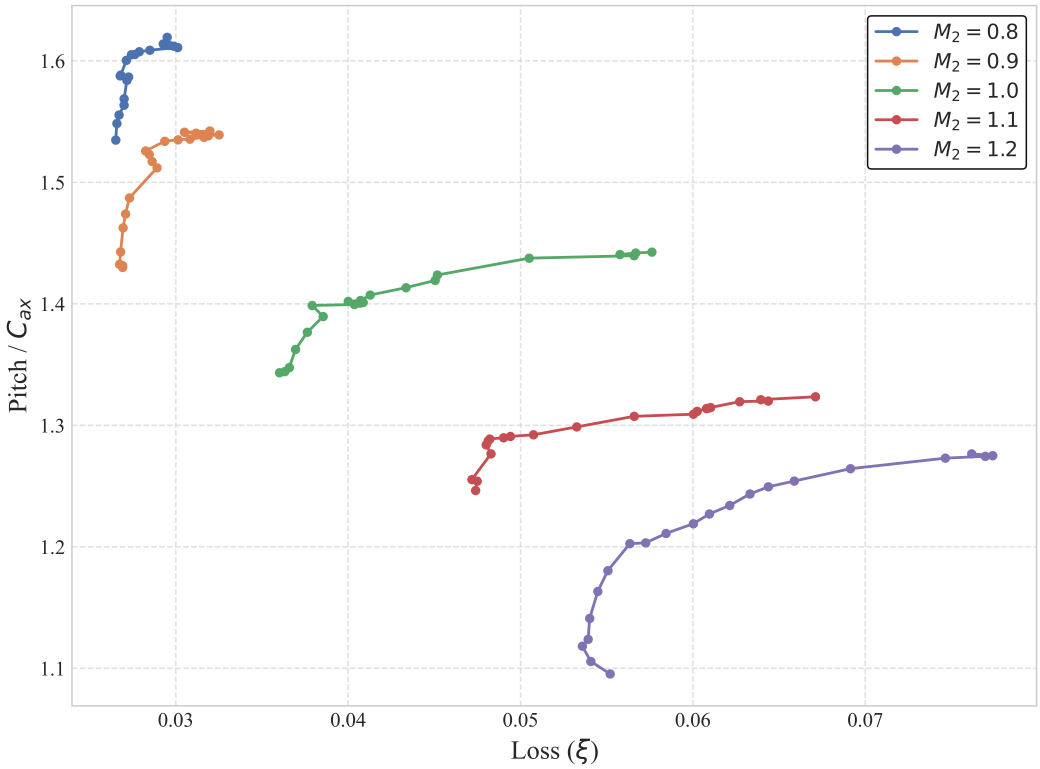


Figure 7.1: Surrogate predictions in the objective space for C4 at all five training Mach numbers. Each point corresponds to a surrogate-generated blade re-simulated with `MISES` at a given p. Colour encodes M_2.

loss or higher pitch than any blade in the training data for that duty point. This result is explained by the **limited CFD budget** of the NSGA-II optimization: on these challenging duty conditions, a significant fraction of evaluations produced non-convergent blades, effectively reducing the budget available for exploring the feasible objective space. The optimizer therefore converged to a **local Pareto front** that does not represent the full achievable trade-off.

The surrogate, trained simultaneously on all 125 duty points, compensates through **cross-duty learning**: it implicitly captures geometric patterns from neighbouring duty points and generates blades in regions of the design space that the single-duty optimizer never reached. These blades are physically valid and aerodynamically superior, as confirmed by `MISES` re-simulation. This result is consistent with the well-established role of surrogate models in the optimization literature [39]; the present work uses the surrogate as a **standalone design tool** rather than as an optimizer assistant, and demonstrates that cross-duty generalization can nonetheless recover performance missed by the single-duty optimization.

A notable observation in the C3 case is that the surrogate generates blades reaching $t/c_{ax} = 2.4493$, marginally exceeding the **maximum pitch in the entire training database of** 2.4327, which belongs precisely to C3 at $M_2 = 0.8$. This should not be

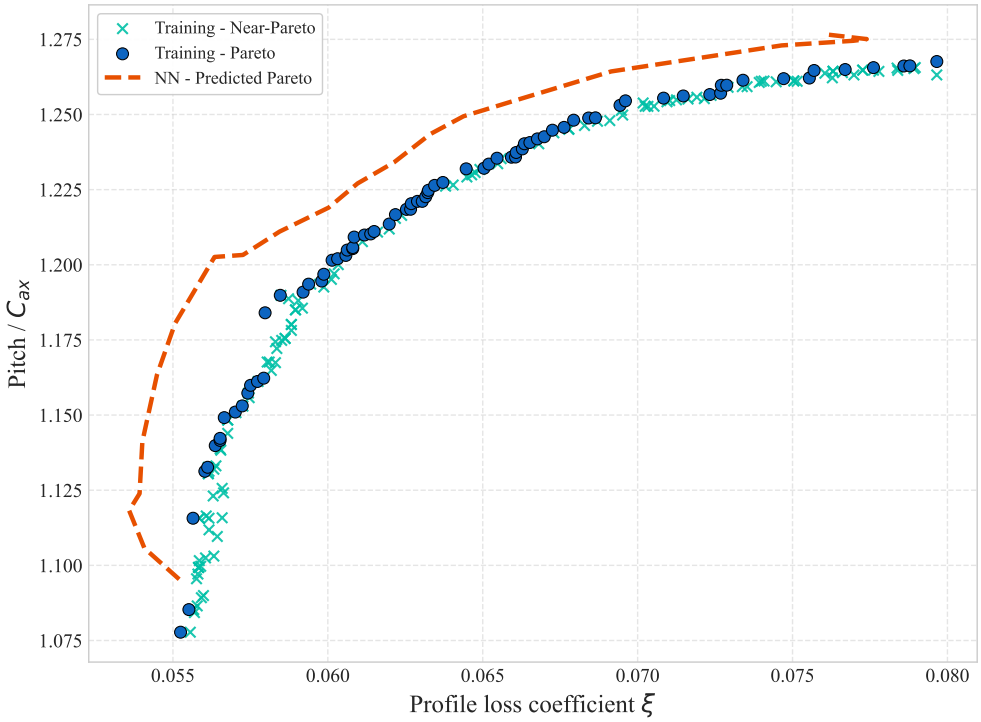


Figure 7.2: C4, $M_2 = 1.2$: surrogate predictions re-simulated with `MISES` (filled markers) vs original training Pareto front (open markers).

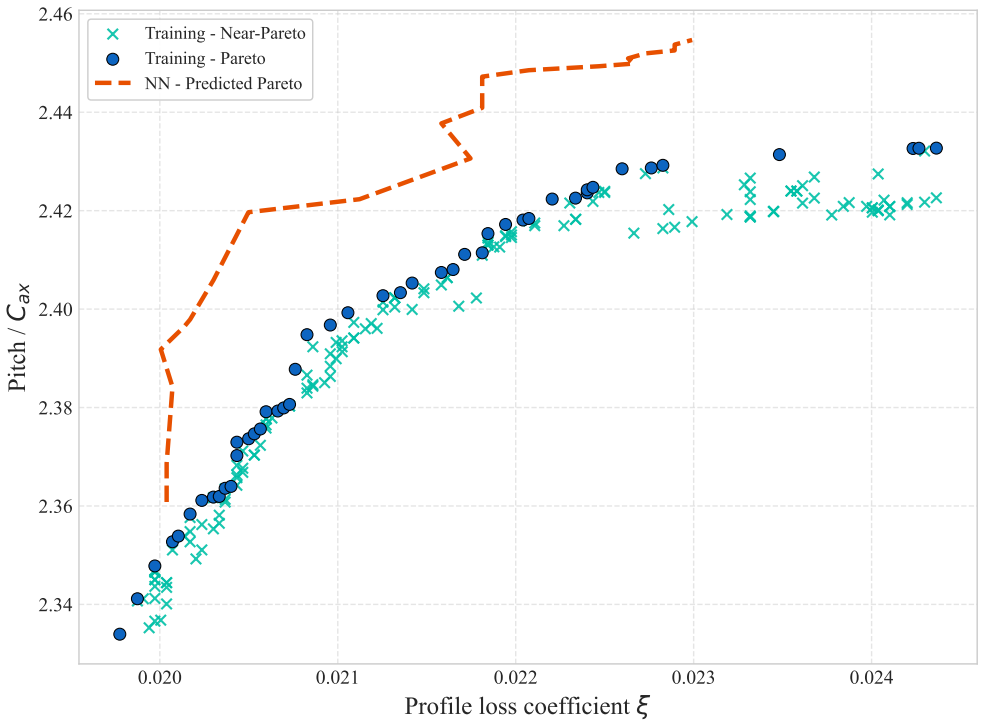


Figure 7.3: C3, $M_2 = 0.8$: surrogate predictions re-simulated with `MISES` (filled markers) vs original training Pareto front (open markers).

interpreted as extrapolation: the network is trained simultaneously on all 125 duty points, spanning a pitch range from 0.73 to 2.4327, so an exceedance of 0.017 corresponds to just **0.01% of the full training range**. This is the natural behaviour of any smooth function at the boundary of its training domain — the network carries no hard constraint on its output range and continues smoothly beyond the last training point.

C2 (Figure 7.4) presents a qualitatively different picture. The surrogate predictions **closely overlap** with the training Pareto front, with no significant extension in either objective. As discussed in the preceding chapters, C2 exhibits the highest rate of non-convergent blades throughout the optimization campaign. The near-perfect overlap between the surrogate predictions and the training Pareto therefore suggests that the optimization has already reached the **physical limit** of this operating condition: the extreme angle combination places the cascade at the boundary of flow separation and choking, leaving little or no room for further improvement regardless of the blade geometry.

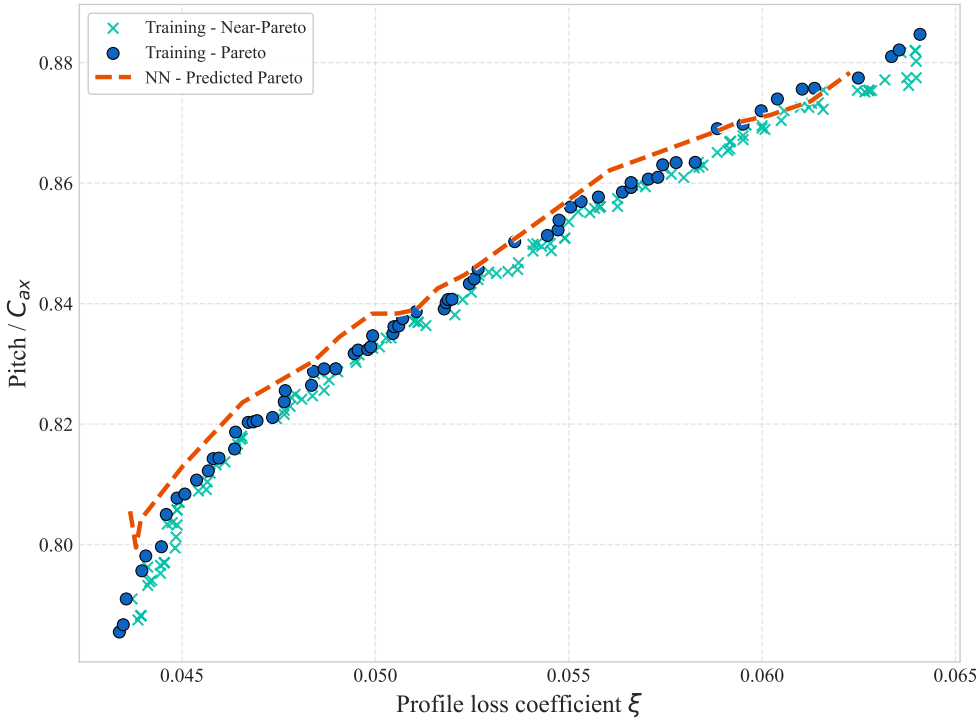


Figure 7.4: C2, $M_2 = 1.0$: surrogate predictions re-simulated with `MISES` (filled markers) vs original training Pareto front (open markers). Non-converged cases are marked with crosses.

7.2.2 Effect of M_2 and p on Blade Geometry and Loading

To provide a physical interpretation of the surrogate outputs, this section examines how **blade shape** and **isentropic Mach number distribution** vary when M_2 or p is

swept while the other is held fixed, for the three duty points C2, C3, and C4. For each duty point, both sweeps are shown.

For **C2**, Figure 7.5 shows the effect of varying p at fixed $M_2 = 1.2$, while Figure 7.6 shows the effect of varying M_2 at fixed $p = 0$.

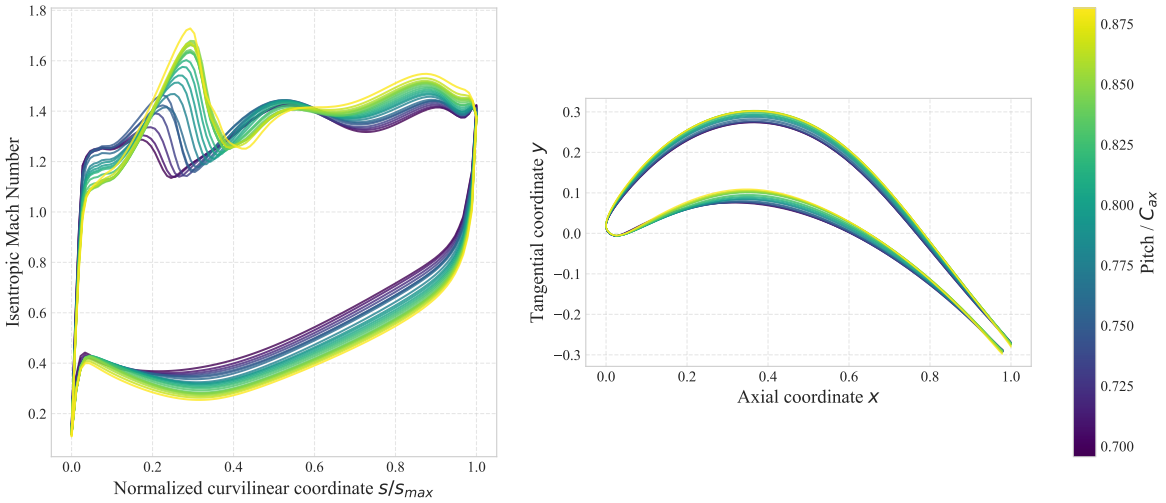


Figure 7.5: C2, $M_2 = 1.2$: isentropic Mach number distributions (left) and blade profile shapes (right) for p sweeping from 0 to 1. Colour encodes *pitch*.

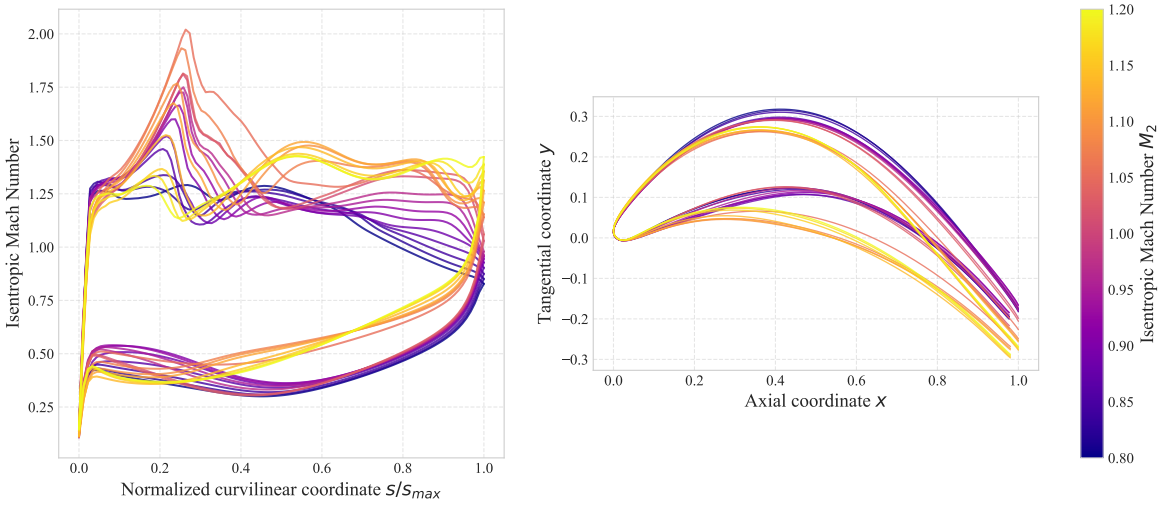


Figure 7.6: C2, $p = 0$: isentropic Mach number distributions (left) and blade profile shapes (right) for M_2 sweeping from 0.8 to 1.2. Colour encodes M_2.

The sweep over p at fixed $M_2 = 1.2$ (Figure 7.5) produces a physically expected result: as pitch increases, the blade becomes **more cambered** and the **peak Mach number on the suction side increases**, reflecting the higher aerodynamic loading demanded by a wider blade spacing.

The sweep over M_2 at fixed $p = 0$ (Figure 7.6) reveals a more interesting pattern. The best-loss solutions across the Mach range exhibit a **bimodal behaviour**: two

distinct aerodynamic configurations emerge, with no stable intermediate state. At **lower** M_2, the optimal blades tend towards a **front-loaded** configuration, characterised by a normal shock on the suction side located around **20% of the axial chord**. At **higher** M_2, the optimal blades instead adopt a **back-loaded** configuration with a lower peak Mach number, distributing the suction-side acceleration further downstream.

A contributing factor to this pattern is that the **generation-zero population** shared the same geometric distribution across all Mach numbers (except from the pitch ranges): this may have biased certain duty points towards one configuration over the other, potentially excluding mixed solutions such as a back-loaded blade at lower M_2. Investigating such configurations on individual duty points represents a natural direction for future targeted studies.

For **C3**, Figure 7.8 shows the effect of varying M_2 at fixed $p = 0.5$, and Figure 7.7 shows the effect of varying p at fixed $M_2 = 0.8$.

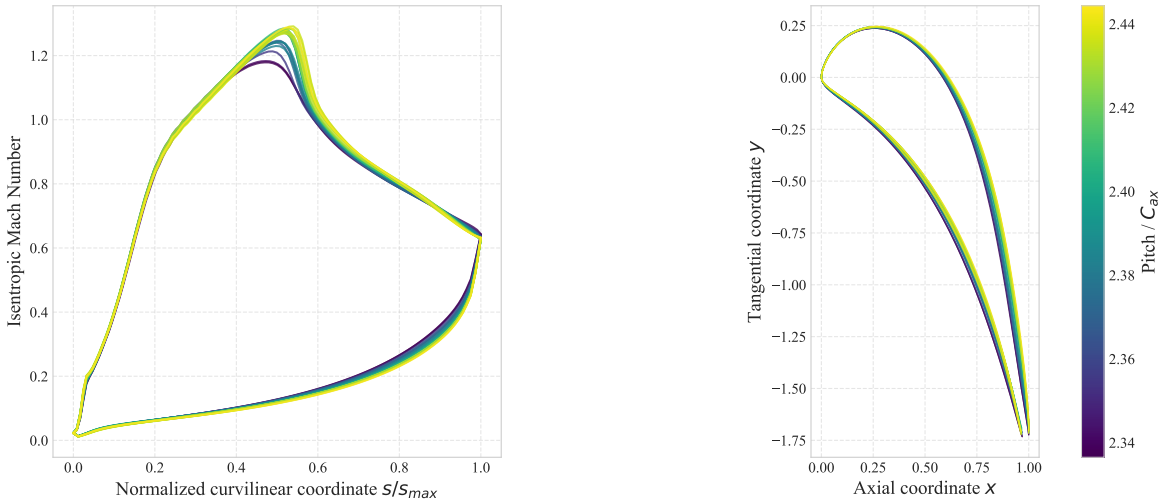


Figure 7.7: C3, $M_2 = 0.8$: isentropic Mach number distributions (left) and blade profile shapes (right) for p sweeping from 0 to 1. Colour encodes *pitch*.

For C3, the sweep over p at fixed $M_2 = 0.8$ (Figure 7.7) produces only **contained geometric changes**: the blade shape varies little across the full p range, consistent with the relatively **narrow pitch range** of this duty point — C3 has one of the most compact Pareto fronts in the training set in terms of achievable loading variation, so the surrogate has little room to differentiate across p.

The sweep over M_2 at fixed $p = 0.5$ (Figure 7.8) shows a more pronounced effect. As M_2 increases, the blade exhibits **less camber**, likely as a response to reduce the wetted surface area and the associated friction losses at higher Mach numbers. The isentropic Mach distribution also exhibits a slightly **bimodal behaviour** with varying M_2: at lower exit Mach numbers a well-defined **normal shock** appears on the suction side, while at higher M_2 this transitions to a more complex **shock reflection pattern**,

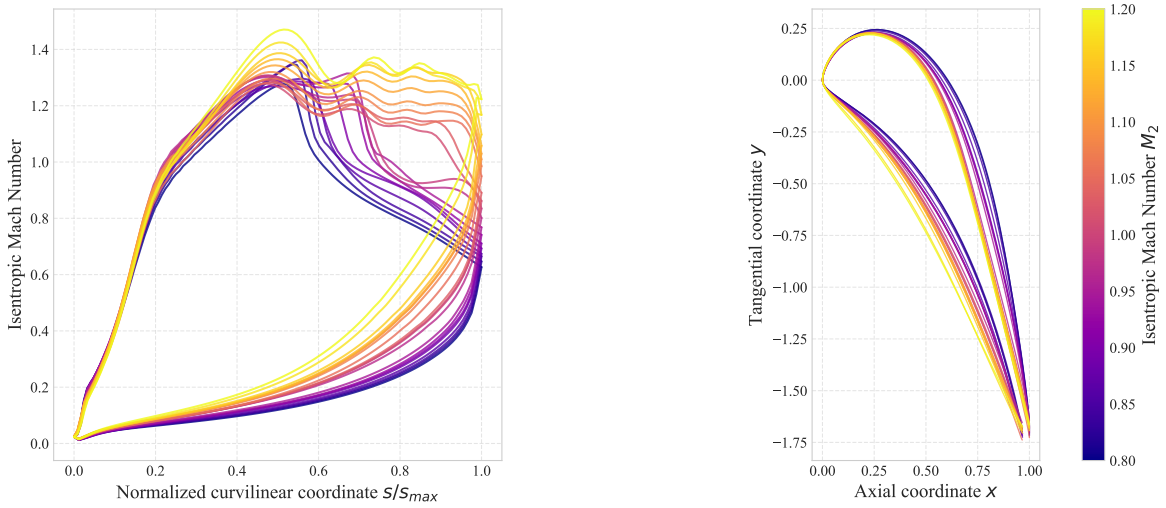


Figure 7.8: C3, $p = 0.5$: isentropic Mach number distributions (left) and blade profile shapes (right) for M_2 sweeping from 0.8 to 1.2. Colour encodes M_2.

without a clearly identifiable single normal shock on the blade surface.

For **C4**, Figure 7.10 shows the effect of varying M_2 at fixed $p = 1$, and Figure 7.9 shows the effect of varying p at fixed $M_2 = 1.2$.

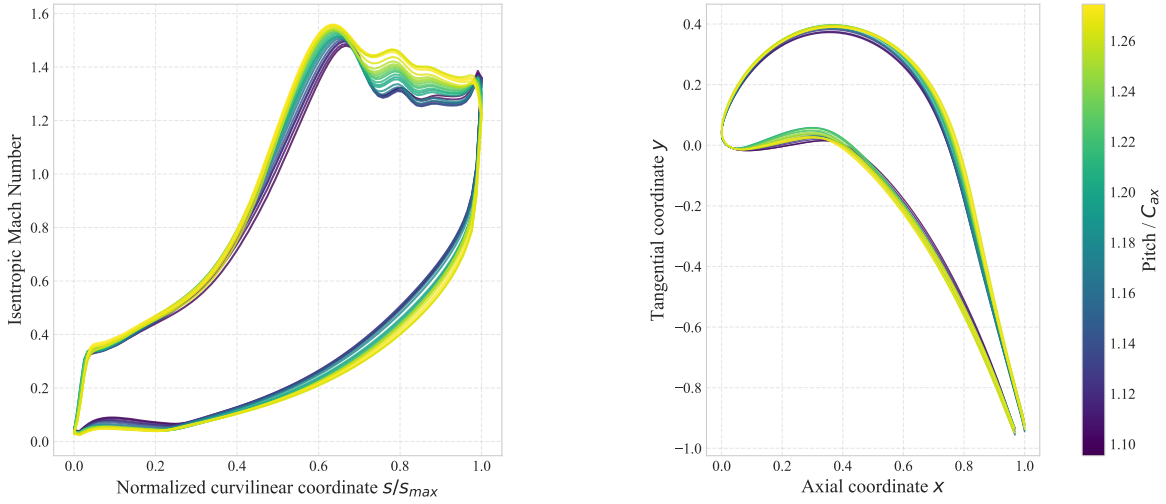


Figure 7.9: C4, $M_2 = 1.2$: isentropic Mach number distributions (left) and blade profile shapes (right) for p sweeping from 0 to 1. Colour encodes *pitch*.

The behaviour observed for C4 closely mirrors that of C3: contained geometric variation with p and a more pronounced effect of M_2, with the same bimodal pattern in the Mach distribution transitioning from a normal shock at lower exit Mach numbers to a complex shock reflection scheme at higher ones.

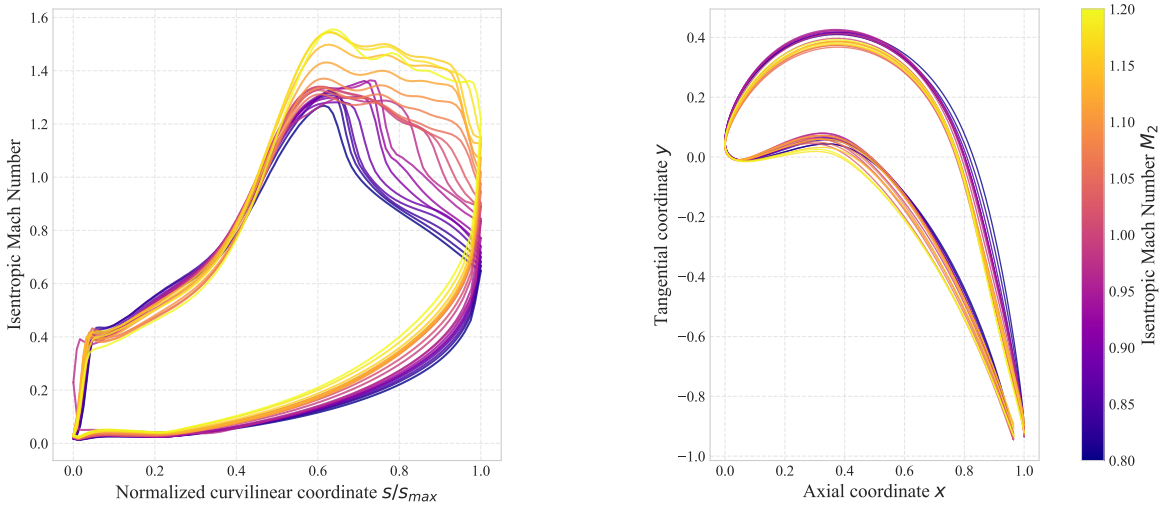


Figure 7.10: C4, $p = 1$: isentropic Mach number distributions (left) and blade profile shapes (right) for M_2 sweeping from 0.8 to 1.2. Colour encodes M_2.

7.3 Outlet Flow Angle Accuracy

The end-to-end accuracy of the pipeline in satisfying the prescribed outlet flow angle is assessed across four **training duty points** and four **unseen duty points**, each evaluated at all five training Mach numbers $M_2 \in \{0.8, 0.9, 1.0, 1.1, 1.2\}$ and at $p \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$, for a total of **100 cases per group**. The selected duty points are listed in Tables 7.1 and 7.2.

Table 7.1: Selected training duty points for outlet flow angle assessment.

Duty	α_1 [°]	α_2 [°]
A9	0	−65
A19	15	−75
A7	40	−60
A16	30	−70

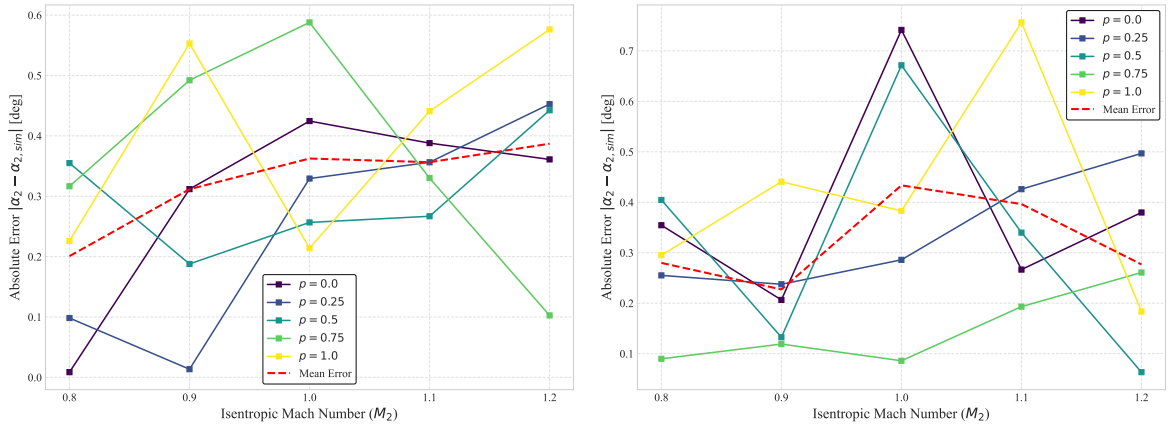
Table 7.2: Unseen duty points.

Duty	α_1 [°]	α_2 [°]
U1	7	−55
U2	22	−63
U3	35	−68
U4	45	−72

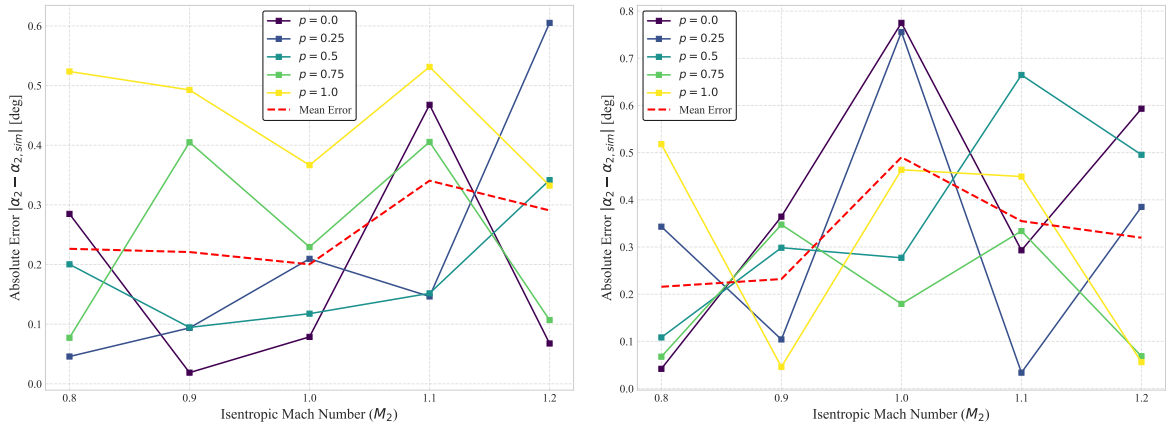
All 200 cases converged without any cascade non-convergence issues, confirming that

the surrogate generates geometrically valid blades across the full range of operating conditions tested.

Figures 7.11 and 7.12 show the absolute outlet flow angle error $|\Delta\alpha_2|$ as a function of p for the training and unseen duty points respectively.



A9 — $\alpha_1 = 0°$, $\alpha_2 = -65°$ A19 — $\alpha_1 = 15°$, $\alpha_2 = -75°$



A7 — $\alpha_1 = 40°$, $\alpha_2 = -60°$ A16 — $\alpha_1 = 30°$, $\alpha_2 = -70°$

Figure 7.11: Absolute outlet flow angle error $|\Delta\alpha_2|$ as a function of p for the four selected training duty points.

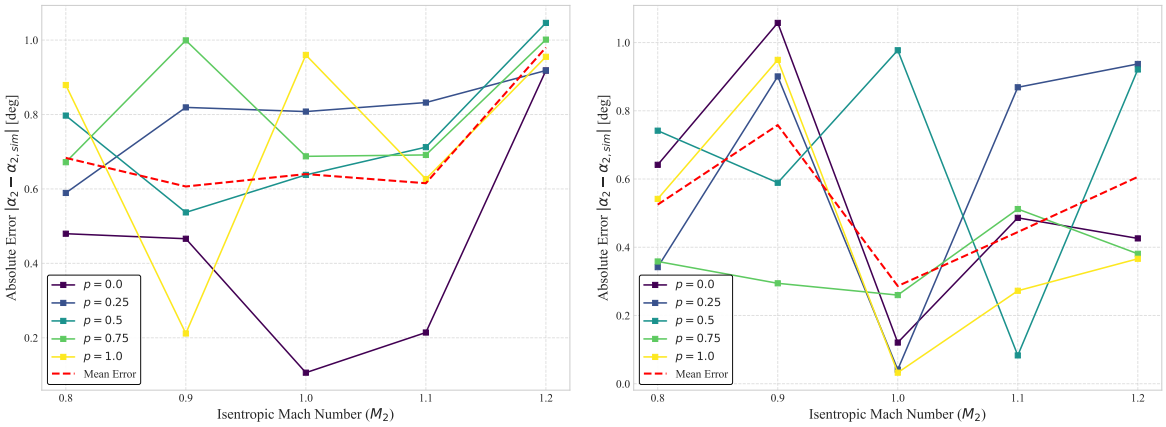
The worst performance is consistently observed at the **extreme values of** p ($p = 0$ and $p = 1$), where the surrogate operates at the boundaries of the Pareto parametrization and the training data is least dense. Table 7.3 summarises the accuracy metrics across all 100 seen cases.

Given the well-known sensitivity of `MISES` in the transonic regime — as discussed in the validation study of Chapter 3 — this level of accuracy is considered a **good result** and a solid starting point for more targeted analyses on individual duty conditions.

For the unseen duty points, the absolute errors are larger, as expected for operating conditions never encountered during training. Table 7.4 reports the global accuracy

Table 7.3: Outlet flow angle error statistics — seen duty points.

Metric	Value
Mean $\lvert\Delta\alpha_2\rvert$	$0.306° \pm 0.191°$
RMSE	$0.360°$
Maximum $\lvert\Delta\alpha_2\rvert$	$0.775°$



U1 — $\alpha_1 = 7°$, $\alpha_2 = -55°$ U2 — $\alpha_1 = 22°$, $\alpha_2 = -63°$

U3 — $\alpha_1 = 35°$, $\alpha_2 = -68°$ U4 — $\alpha_1 = 45°$, $\alpha_2 = -72°$

Figure 7.12: Absolute outlet flow angle error $|\Delta\alpha_2|$ as a function of p for the four unseen duty points.

metrics across all 100 unseen cases.

Table 7.4: Outlet flow angle error statistics — unseen duty points.

Metric	Value
Mean $\lvert\Delta\alpha_2\rvert$	$0.512° \pm 0.308°$
RMSE	$0.597°$
Maximum $\lvert\Delta\alpha_2\rvert$	$1.058°$

7.4 Conclusions and Future Work

This work has developed a complete **data-driven pipeline** for the preliminary aerodynamic design of transonic 2D turbine cascades, from the generation of a physically consistent Design of Experiments to the deployment of a neural network surrogate capable of producing near-optimal blade geometries for any prescribed operating condition within the design space.

7.4.1 Summary of Contributions

The four principal components of the pipeline — correlation-preserving DoE, `MISES`-based multi-objective optimization, surrogate training, and inference — were developed and validated in sequence.

The **Design of Experiments** built upon a reference database of 30 transonic blades spanning low, medium, and high turning regimes. By fitting a multivariate Gaussian model to the 21 `ParaBlade` parameters and enforcing the inter-parameter correlations via the **Iman–Conover algorithm**, the DoE generates geometrically feasible blade populations that respect the physical dependencies of the parameter space, reducing the rate of non-convergent CFD evaluations compared to uncorrelated sampling.

The **multi-objective optimization** with NSGA-II, applied across 125 duty points, generated Pareto fronts that simultaneously minimize **aerodynamic loss** and maximize **blade loading**, for a total of **1.25 million `MISES` simulations**.

The arc-length parametrization of the Pareto front via the coordinate $p \in [0,1]$ — equivalent to a **normalized pitch** selector — converts the multi-valued optimization output into a well-posed single-valued regression problem. The **neural network surrogate** (4 hidden layers, 256 units, **SiLU** activation) was trained on approximately 12,500 samples and achieves a **mean R^2 of 0.94** across all 21 geometric outputs, with sub-degree angular errors on camberline parameters and physically interpretable accuracy degradation on pressure-side thickness, which is weakly constrained by the aerodynamic objectives. The model is trained on all 125 duty points simultaneously, enabling cross-duty generalization.

The **validation campaign** confirmed the pipeline's ability to generate aerodynamically valid blades for both training and unseen operating conditions. On **seen duty points** the outlet flow angle RMSE is 0.360°, remaining below 1% relative error in all cases. On **unseen duty points** the RMSE rises to 0.597° with a maximum of 1.058°, and the 1% threshold can be exceeded, particularly for conditions falling in the largest gaps of the training angle grid.

Furthermore, an interesting result emerged from the comparison with the training Pareto fronts: in some corner point cases, the surrogate generates blades that achieve

slightly better objective values than the training Pareto front, with marginal improvements in loss or pitch confirmed by direct `MISES` re-simulation. This is an encouraging sign of the **cross-duty generalization** capability of a model trained simultaneously on all 125 duty points, motivating the use of surrogate models not only as standalone design tools, but also as active components within the optimization loop itself.

7.4.2 Limitations

The current implementation carries several limitations that are acknowledged and inform the future work directions below.

The **binning strategy** used to build the training dataset introduces near-Pareto dominated blades in bins not covered by true Pareto solutions. This causes mild imperfections in the surrogate predictions — dominated solutions at low p and pitch inversions near the knee of the Pareto curve — that could be mitigated by higher-quality optimization outputs.

The **outlet flow angle accuracy on unseen duty points** exceeds the 1% threshold in some cases, in particular at $\alpha_2 = -55°$. This condition falls in the largest gap of the training angle grid, between $\alpha_2 = -50°$ and $\alpha_2 = -60°$, where the surrogate must interpolate across the widest angular distance. This is the most direct indicator that the current training grid is insufficiently dense for reliable generalization across the full design space.

The **shared generation-zero population** across all Mach numbers may have introduced a bias in the optimization results for some duty points. While the pitch range was adjusted for each Mach condition, the underlying blade geometry distribution was the same for every Mach, meaning the optimizer started from the same geometric seed regardless of the operating condition. This may have led to a preferential convergence towards loading configurations well-represented in the initial population, potentially limiting the geometric diversity of the resulting Pareto fronts at individual Mach conditions.

The **surrogate accuracy degrades on pressure-side thickness parameters**, which exhibit the highest intrinsic variability along the Pareto front due to their weak aerodynamic selection. Individuals with very different pressure-side shapes produce nearly identical loss and loading values, so the optimizer cannot distinguish between them. This is a fundamental characteristic of the design space rather than a modelling limitation, but it implies that pressure-side shape predictions carry higher uncertainty and should be interpreted accordingly.

7.4.3 Future Work

Several directions for future development emerge naturally from the results of this work.

Denser and more uniform duty grid

The most direct improvement is to refine the training grid of duty points, reducing the angular spacing between adjacent conditions — particularly in the α_2 direction where the current step of 10° between −50° and −60° represents the largest interpolation gap. A uniform and finer grid would reduce the surrogate prediction error on unseen conditions and bring all duty points within the 1% accuracy threshold.

Dataset quality and Pareto resolution

Increasing the CFD budget per duty point would reduce the fraction of near-Pareto blades in the training data and improve the Pareto front resolution. A long-term goal is to train the surrogate on exclusively **true Pareto-optimal solutions**, progressively reducing the near-Pareto contribution. This is achievable by combining a larger optimization budget with **surrogate-assisted optimization**: after an initial NSGA-II campaign, the surrogate can be queried to identify promising regions not yet explored, and targeted CFD evaluations can be added iteratively via active learning to densify the Pareto front where coverage is sparsest.

Extension of operating conditions

The current pipeline fixes the Reynolds number at $Re_2 = 10^6$ and the trailing-edge thickness at $t_{\mathrm{TE}}/c_{ax} = 3\%$. Extending the framework to multiple exit Reynolds numbers and trailing-edge thicknesses would broaden the applicability of the surrogate to a wider range of turbomachinery design scenarios.

Three-dimensional extension

The framework operates exclusively in 2D, optimizing profile losses only. A natural next step would be to extend the pipeline to 3D blade design, incorporating spanwise effects and targeting not only profile losses but also secondary flow losses [40].

Loading style characterization and clustering

The direct extension of the work of Porta [6] to the transonic regime would require the ability to prescribe, alongside the aerodynamic duty, the **aerodynamic style** — that is, the Mach number distribution along the blade. As discussed in Chapter 2, the strong non-linearities of transonic flow and the bimodal behaviour observed in the present results suggest that prescribing a precise target Mach distribution may be significantly more challenging in the transonic regime than in the subsonic case.

However, the present pipeline provides the data foundation to begin characterizing the key features of transonic Mach distributions through **clustering**: rather than targeting a specific peak Mach number, its axial position, or a precise suction-side acceleration profile, the designer could specify the *type* of loading — front-loaded versus back-loaded, presence or absence of a normal shock, shock intensity as measured by the Mach gradient — and the surrogate could be queried accordingly. Reduced-order models and unsupervised clustering techniques applied to the 1.25 million `MISES` simulations already available represent a natural tool for this analysis.

Multi-objective design with diversity

The present pipeline focuses exclusively on Pareto-optimal and near-Pareto solutions, optimizing the trade-off between loss and loading. A natural extension is to broaden the design space to an **ε-Pareto band** — a wider neighbourhood of the Pareto front that includes geometrically diverse but aerodynamically competitive blades. This would allow the designer to impose additional requirements beyond aerodynamic performance: **structural constraints** such as blade thickness or cross-sectional area, geometric properties such as trailing-edge wedge angle, or loading style requirements as described above. The resulting framework would transition from a pure performance optimizer to a genuine **multi-criteria preliminary design tool**, directly enabling the kind of design space exploration that motivated this work from the outset.

Bibliography

[1] J. Li, X. Du, and J. R. R. A. Martins. «Machine learning in aerodynamic shape optimization». In: *Progress in Aerospace Sciences* 134 (2022), p. 100849 (cit. on pp. 1, 2).

[2] R. Yondo, E. Andrés, and E. Valero. «A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses». In: *Progress in Aerospace Sciences* 96 (2018), pp. 23–61 (cit. on pp. 1, 62).

[3] M. A. Mendez, J. Dominique, M. Fiore, F. Pino, P. Sperotto, and J. Van den Berghe. «Challenges and Opportunities for Machine Learning in Fluid Mechanics». In: *arXiv preprint arXiv:2202.12577* (2022) (cit. on p. 2).

[4] H. Zhou, K. Yu, Q. Luo, L. Luo, W. Du, and S. Wang. «Design Methods and Strategies for Forward and Inverse Problems of Turbine Blades Based on Machine Learning». In: *Journal of Thermal Science* 31.1 (2022), pp. 82–95 (cit. on p. 2).

[5] J. Chen, C. Liu, L. Xuan, Z. Zhang, and Z. Zou. «Knowledge-based turbomachinery design system via a deep neural network and multi-output Gaussian process». In: *Knowledge-Based Systems* 252 (2022), p. 109352 (cit. on pp. 2, 3).

[6] F. Porta. «Development of a Machine-Learning-Based Tool for Turbomachinery Blades». Master's Thesis. Milan, Italy: Politecnico di Milano, 2023 (cit. on pp. 3, 14, 16, 18, 57, 82, 139).

[7] I. Fruchtman. «The Limit Load of Transonic Turbine Blading». In: *ASME 1974 International Gas Turbine Conference and Products Show*. GT 74-GT-80. American Society of Mechanical Engineers. Mar. 1974 (cit. on p. 4).

[8] M. Ni, Z. Wei, W. Deng, G. Ren, and X. Gan. «Modified Zweifel Coefficient and Lift Coefficient Definition Considering Compressible Effect». In: *Aerospace* 11.8 (2024), p. 650 (cit. on pp. 4, 70).

[9] S. Wassing, S. Langer, and P. Bekemeyer. «Physics-Informed Neural Networks for Transonic Flows around an Airfoil». In: *AIAA SCITECH 2024 Forum*. (Also referenced as DLR Internal Report/Preprint). 2024 (cit. on p. 4).

[10] R. Agromayor, N. Anand, J.-D. Müller, M. Pini, and L. O. Nord. «A Unified Geometry Parametrization Method for Turbomachinery Blades». In: *Computer-Aided Design* 133 (2021), p. 102987 (cit. on pp. 5, 23, 25, 28).

[11] L. J. Pritchard. «An Eleven Parameter Axial Turbine Airfoil Geometry Model». In: *ASME 1985 International Gas Turbine Conference and Exhibit*. GT 85-GT-219. American Society of Mechanical Engineers. 1985 (cit. on p. 5).

[12] A. Prots, M. Voigt, and R. Mailach. «Improvements in Probabilistic Strategies and Their Application to Turbomachinery». In: *Aerospace* 11.5 (2024), p. 355 (cit. on pp. 5, 6).

[13] R. L. Iman and W. J. Conover. «A Distribution-Free Approach to Inducing Rank Correlation Among Input Variables». In: *Communications in Statistics - Simulation and Computation* 11.3 (1982), pp. 311–334 (cit. on pp. 6, 67).

[14] M. L. Shelton, B. A. Gregory, S. H. Lamson, H. L. Moses, R. L. Doughty, and T. Kiss. «Optimization of a Transonic Turbine Airfoil Using Artificial Intelligence CFD and Cascade Testing». In: *ASME 1993 International Gas Turbine and Aero-engine Congress and Exposition*. GT 93-GT-161. American Society of Mechanical Engineers. 1993 (cit. on p. 7).

[15] T. Sonoda, T. Arima, M. Olhofer, B. Sendhoff, F. Kost, and P. A. Giess. «A Study of Advanced High-loaded Transonic Turbine Airfoils». In: *Journal of Turbomachinery* 128.4 (2006), pp. 650–657 (cit. on p. 7).

[16] S. Chen and J. Montgomery. «Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution». In: *Applied Intelligence* 42 (2015), pp. 514–526 (cit. on p. 7).

[17] A. Jameson. *Aerodynamic Design via Control Theory*. Tech. rep. 88-64. Institute for Computer Applications in Science and Engineering (ICASE), 1988 (cit. on p. 8).

[18] A. Châtel, T. Verstraete, G. Coussement, and L. Mueller. «Single-Point Optimization of the LS89 Turbine Cascade Using a Hybrid Algorithm». In: *Proceedings of the ASME Turbo Expo* 2C (2019). GT2019-91471 (cit. on p. 8).

[19] L. Piegl and W. Tiller. *The NURBS Book*. 2nd. Springer Science & Business Media, 2012. DOI: 10.1007/978-3-642-97385-7 (cit. on p. 25).

[20] M. Drela. *A Newton method for compressible viscous-inviscid analysis of cascade flows*. Tech. rep. Massachusetts Institute of Technology (MIT), 1987 (cit. on p. 33).

[21] M. Drela and H. Youngren. *A User's Guide to MISES 2.53*. Tech. rep. MIT Computational Aerospace Sciences Laboratory, Feb. 2008 (cit. on pp. 33, 46).

[22] P. L. Andrew and H. S. Kahveci. «Validation of MISES Two-Dimensional Boundary Layer Code for High-Pressure Turbine Aerodynamic Design». In: *Journal of Turbomachinery* 131.1 (2009) (cit. on pp. 37, 42).

[23] D. B. M. Jouini. «Experimental Investigation of Two Transonic Linear Turbine Cascades at Off-Design Conditions». Ph.D. thesis. Ottawa, Canada: Department of Mechanical and Aerospace Engineering, Carleton University, 2000 (cit. on p. 37).

[24] D. Corriveau and S. A. Sjolander. «Experimental and Numerical Investigation on the Performance of a Family of Three HP Transonic Turbine Blades». In: *ASME Turbo Expo 2004: Power for Land, Sea, and Air*. ASME paper GT2004-53087. 2004 (cit. on p. 37).

[25] M. D. McKay, R. J. Beckman, and W. J. Conover. «A comparison of three methods for selecting values of input variables in the analysis of output from a computer code». In: *Technometrics* 21.2 (1979), pp. 239–245 (cit. on p. 62).

[26] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. 3rd. Wiley-Interscience, 2003 (cit. on p. 63).

[27] W. H. Kruskal. «Ordinal Measures of Association». In: *Journal of the American Statistical Association* 53.284 (1958), pp. 814–861 (cit. on p. 65).

[28] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, UK: John Wiley & Sons, 2001 (cit. on pp. 70, 107).

[29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. «A fast and elitist multiobjective genetic algorithm: NSGA-II». In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197 (cit. on pp. 81, 89, 91).

[30] R. Storn and K. Price. «Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces». In: *Journal of Global Optimization* 11.4 (1997), pp. 341–359 (cit. on p. 83).

[31] H. Kita, I. Ono, and S. Kobayashi. «Multi-parental extension of the unimodal normal distribution crossover for real-coded genetic algorithms». In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*. IEEE. 1999, pp. 1581–1588 (cit. on p. 92).

[32] O. Zweifel. «The Spacing of Turbo-Machine Blading, Especially with Large Annular Deflection». In: *Brown Boveri Review* 32.12 (1945), pp. 436–444 (cit. on p. 95).

[33] J. D. Coull and H. P. Hodson. «Blade Loading and Its Application in the Mean-Line Design of Low Pressure Turbines». In: *Journal of Turbomachinery* 135.2 (2013) (cit. on p. 96).

[34] B. H. Dennis, I. N. Egorov, Z.-X. Han, G. S. Dulikravich, and C. Poloni. «Turbomachinery cascades for minimum loss, maximum loading, and maximum gap-to-chord ratio». In: *8th Symposium on Multidisciplinary Analysis and Optimization*. Sept. 2000, p. 4885 (cit. on p. 101).

[35] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. «Combining Convergence and Diversity in Evolutionary Multiobjective Optimization». In: *Evolutionary Computation* 10.3 (2002), pp. 263–282 (cit. on pp. 112, 113).

[36] A. Paszke et al. «PyTorch: An Imperative Style, High-Performance Deep Learning Library». In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035 (cit. on p. 117).

[37] T. Akiba, S. Sano, T. Yanase audits, T. Ohta, and M. Koyama. «Optuna: A Next-generation Hyperparameter Optimization Framework». In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 2623–2631 (cit. on p. 118).

[38] D. P. Kingma and J. Ba. «Adam: A Method for Stochastic Optimization». In: *International Conference on Learning Representations (ICLR)* (2015). arXiv preprint arXiv:1412.6980 (cit. on p. 119).

[39] Alexander I. J. Forrester, András Sóbester, and Andy J. Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008 (cit. on p. 128).

[40] Cosmo Scarimbolo. «Development of a data-driven design tool for turbomachinery blades: Exploration of secondary flows in 3D turbine cascades». Submitted in collaboration with the von Karman Institute for Fluid Dynamics. MA thesis. Politecnico di Torino, 2025 (cit. on p. 139).