NAZARBAYEV UNIVERSITY

School of Engineering and Digital Sciences

Bachelor of Engineering in
Mechanical and Aerospace Engineering

Dimensionality Reduction in Design Optimization

(Final Capstone Project Report)

by

Yernur Koshkarbay, Suienish Sultangazin, Sanzhar Kaiyrov, and Aisultan Omirbay

Lead Supervisor: Prof. Konstantinos Kostas

Co-Supervisor: Prof. Yerkin Abdildin

April, 2024

Declaration

We, Yernur Koshkarbay, Suienish Sultangazin, Sanzhar Kaiyrov, and Aisultan Omirbay, hereby declare that this report, entitled "Dimensionality Reduction in Design Optimization" is the result of our own project work except for quotations and citations which have been duly acknowledged. We also declare that it has not been previously or concurrently submitted for any other degree at Nazarbayev University or elsewhere.

Signature:

Name: Yernur Koshkarbay

Date: April 28, 2024

Signature:

Name: Suienish Sultangazin

Date: April 28, 2024

Signature:

Name: Sanzhar Kaiyrov

Date: April 28, 2024

Signature:

Name: Aisultan Omirbay

Date: April 28, 2024

Acknowledgements

This project owes its success to the collaborative efforts and contributions of many individuals and institutions. First and foremost, our gratitude extends to our academic supervisors, whose guidance and expertise have been invaluable throughout the course of this research. Special thanks are due to the scholars Masood, Kostas, Khan, Iznat and Kaklis, whose seminal work in the field of dimensionality reduction provided a critical foundation for our methodology. Appreciation is also extended to the developers of Rhino 7, XFLR5, and MATLAB whose software tools were instrumental in our modeling and analysis.

We also wish to thank our peers and colleagues at Nazarbayev University, for their insightful feedback and constructive criticism during various stages of this project. Their perspectives have been crucial in refining our approach and methodologies.

Abstract

sec:app

The engineering design of complex free-form surfaces such as turbine blades and ship hulls presents significant challenges due to the intricate geometries involved. This study investigates the application of dimensionality reduction techniques to optimize the design process of such surfaces, with a focus on enhancing computational efficiency and maintaining high design quality. We propose a novel approach that integrates geometric moments with Karhunen-Loeve Expansion (KLE) to form a reduced-dimensional design space that retains essential shape characteristics while reducing computational demands. Our methodology leverages the collection of geometric moments to enrich the latent space representation, which is crucial for capturing the physical and geometric nuances of the designs. By employing this approach, we aim to address the curse of dimensionality often encountered in engineering optimization tasks. The findings suggest that incorporating geometric moments noticeably improves the quality of the resultant design space by maintaining variance and enhancing the validity of the designs without extensive computational resources. This research not only contributes to the academic field by providing a feasible solution to a common problem in engineering design but also suggests practical applications in optimizing design processes across various industries.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

chap: introduction

This chapter provides background of the study, its objectives, rationale, and application; a brief description of the approach; the outline of the report and workload distribution among group members.

1.1 Background and significance

A great deal of engineering products rely heavily on free-form functional surfaces, including turbine blades, ship hulls, wings, etc. The quality of such surfaces plays a significant part in defining the product performance, including its energy efficiency, structural integrity, durability, safety, and financial feasibility, while their complexity goes far beyond simple shapes. They usually possess complicated geometries, the modeling of which requires advanced techniques like parametric modeling to allow an efficient representation and grasping of their free-form essence. Modeling of geometric objects has been under perpetual advancement through the collective effort of mathematicians, engineers, and computer scientists, which significantly enhanced the possibilities offered by parametric modeling. This modern technique deals effectively with the intricacy of the forms encountered in a wide range of fields, including - but not limited to – turbomachinery, aeronautic engineering, and automotive design [1, 2].

Meeting the demand for avant-garde designs requires global optimization over wide design spaces with more design variables. However, this task faces the curse of dimensionality, which hinders optimization performance with the increase in the design space dimension number [3, 4]. Initially, methods for reducing the design space dimensionality aim to recognize significant factors and variables, while eliminating the less important ones by making their values constant [5]. However, this approach oversimplifies the problem and may not yield optimal results because it disregards variables' interdependencies and their impact on optimization. Variance-based sensitivity analysis, utilizing Sobol indices, provides a more refined understanding of variable significance but demands a large number of samples and substantial computational power with increasing dimensionality [6]. To tackle these challenges, different techniques for reducing dimensionality have been used. These techniques include unsupervised learning, principal component analysis, and feature extraction, etc. They aim to capture the main directions in the design space while keeping important features intact [1]. The latter set of methods helps in optimizing efficiently and gaining knowledge by revealing the underlying structures.

Using upfront methods to reduce the design space's dimensionality can be beneficial for tackling complex simulation-driven optimization problems in designing functional surfaces. These modern methods evaluate the variability within the design space and then shrink it accordingly. This allows for effective dimensionality reduction before starting the optimization process. For instance, a method applying the Karhunen-Loeve expansion (KLE) was introduced by [7, 2]. KLE is a technique based on providing an approximation of a stochastic or unpredictable process with a finite number of dimensions. It is used to assess variability in shape modification and creates a lower dimension latent feature space for the shape adjustment vector. Although this method is effective, it still has some downsides, one of which is its inability to fully capture the complexity of shapes and their geometric structure. Consequently, the resulting latent spaces may not efficiently produce a wide range of valid shapes of sufficient diversity. This lack of the validity of shapes can hamper the optimizing process by spending too many resources analyzing shapes that are not feasible or valid. Additionally, these methods mainly focus only on the features associated with the geometry of the shape and do not consider performance criteria and the physics involved in evaluating the design. Therefore, using more advanced representations that include information about the structure and physical features of the shape is crucial.

Briefly restating it, the advancement of parametric modeling techniques has significantly improved the representation of complex geometric objects crucial in engineering design. However, challenges such as the curse of dimensionality persist in optimizing design spaces, requiring the utilization of various dimensionality reduction methods.

1.2 Proposed approach

This work proposes a methodology aimed at solving the issues presented above. A set of moments is exploited to gain the maximum level of variance of physical and geometric features accumulated in the latent space by breaking down the wing body geometry into numerous sets of shapes in this approach. Following that, Gauss's divergence theorem is applied for the evaluation of the specific order geometric moments of shape sets, which provide vital insights into their potential performance attributes and volume distribution. The geometric moments are collected before creating the design space of lower dimensionality. They are used to improve the quality of the data stored in the latent space by integrating physical characteristics into the shape characteristics of the design. Despite geometric moments being less informative compared to analyzing actual performance criteria of wings through computational simulation, the latter tends to be extremely demanding of resources, both computational and time. Thus, combining geometric features with moments in creating the latent space results in a diverse and robust design space containing the description of wings of a physics-informed nature, which has the potential to have a high convergence rate in design optimization applications without being computationally intensive.

Coming back to the brief overview of the procedure, after acquiring geometric moments of shapes, they are applied, in combination with the modification function, to create a Shape Signature Vector (SSV) function. This function plays the role of a descriptor that represents each variant in the latent space. Then, the Karhunen-Loeve Expansion of the shape signature vector is calculated, and the features resulting from this variational problem are expressed in the form of eigenfunctions. The recognition of each feature as active or inactive is allowed by their KL-values, active features acting as a new foundation for the expansion of the subspace while retaining maximum geometry variance. Finally, a variety of instruments are applied to evaluate the quality of created subspaces, which will "filter" created designs, thus, partially negating the effect of the absence of actual performance characteristics in the design space.

The main purposes of this research are to examine the feasibility of shape- and physics-informed dimensionality reduction in the optimization process and to evaluate the required number of dimensions that allow to retain the required level of variance, diversity, and validity. This research was motivated by the work of Masood et al (2023) [1] and Khan et al. (2022) [2], which will be covered in the "Literature review" section.

This study addresses the critical need for effective optimization of engineering products with complex free-form surfaces, such as turbine blades and ship hulls. The curse of dimensionality poses significant challenges in optimizing these surfaces, negatively impacting the development of product performance and competitiveness. By exploring dimensionality reduction techniques, this research aims to potentially provide means for the enhancement of optimization efficiency. Through this study, we seek to provide valuable insights into the practical application of dimensionality reduction methods in engineering design, contributing to advancements in the field and addressing industry needs for innovative designing techniques.

The application of the outcome of this study is not limited to designing and optimizing 3D wings, which we are focusing on, but can include the development of any engineering products ranging from turbine blades to ship hulls, i.e., structures relying on free-form surfaces.

1.3 Report structure

The remainder of the report is organized as follows and each chapter has a brief description of the content in the beginning.

Chapter 1 Introduction - This chapter provides background of the study, its objectives, rationale, and application; a brief description of the approach; the outline of the report and workload distribution among group members.

Chapter 2 Literature review - The literature review chapter covers previous works related to the topic of this study - their outcomes and gaps; and provides a theoretical basis that briefly covers subjects required in the Methodology section.

Chapter 3 Methodology - This part of the report includes a brief formulation of the problem, and covers the procedure, software used, variables, and parameters used in the study.

Chapter 4 Implementation - This chapter covers the implementation of the proposed approach through MATLAB. All source codes are provided in Appendix A.

Chapter 5 Results and Discussion - This section covers the results, including the efficiency of dimension reduction, latent space quality, and how these two are impacted by the method of point distribution along the curve and the inclusion of geometric moments.

Chapter 6 Conclusions

Appendix A

1.4 Workload distribution among group members

- Wing DB collection - Aisultan
- Discretization of the wing profile - Yernur
- Dimensionality Reduction - Yernur, Aisultan
- Calculation of moments - Sanzhar, Suienish
- Parametric modeling - Suienish, Sanzhar
- Preparing the report - Suienish, Sanzhar, Aisultan

Chapter 2

Literature review

chap: literature review

The literature review chapter covers previous works related to the topic of this study - their outcomes and gaps; and provides a theoretical basis that briefly covers subjects required in the Methodology section.

2.1 Related works

One of the well-known dimensionality reduction methods is Locality Preserving Projection (LPP), which is an unsupervised technique based on feature extraction [2]. It is a variance of the method called Laplacian Eigenmaps (LE) with a linear approximation [8]. LPP aims to reduce the number of dimensions of the space by applying linear transformation while projecting the initial high-dimensional data space onto the new space with a lower number of dimensions and preserving the links between the nearest neighboring points. Put simply, LPP strives to keep the local relationships and neighborhood arrangement intact as it reduces the dimensionality of the data.

One of the main limitations of this method in dimensionality reduction, despite its effectiveness in some of the datasets, was its high sensitivity to outliers and the use coordinate system for data nodes due to the use of the L2-norm as a distance criterion. One of the variants of LPP called Improved Locality Preserving Projection (ILLP-L1) was aimed at negating or minimizing the effect of those limitations by applying L1-norm instead [9], and it succeeded in achieving a more robust data point translation. However, this fundamental limitation of the LPP method expressed in its lack of invariance to translation is a substantial drawback in our study.

Another framework used in relevant literature is Generative Adversarial Networks (GANs), which is a deep learning architecture that puts a pair of neural networks in a training competition to acquire more authentic data from the feed data set [10]. However, the adoption of standard GANs for dimensionality reduction of aerodynamic designs such as airfoils or wings is impossible, as this method is specialized in generating discrete features like images and fails at retaining continuity properties that are critical for the designing of free-form surfaces. Therefore, [10] tried to augment GANs and proposed a Bezier-GANs generative model, which incorporated Bezier curves by limiting the shapes generated by standard GANs to shapes that conform to Bezier curves [10]. This allowed them to achieve a higher quality of optimization compared to earlier techniques. [11] tried to achieve similar objectives by augmenting GANs with the more complex parametric curve – B-spline curves. The main difference between Bezier and B-spline curves is that the former does not possess internal knots, which makes it a less flexible and less generalized version of the latter. There is no direct comparison between the results of these two methods due to their being accomplished at around the same time. Despite their success in testing with a limited data set, both BSplineGAN and Bezier-GANs were prone to criticism in the same vein – optimization of shapes using the spaces reduced through these frameworks tend to return appropriate results only if a sufficient quantity of modes is integrated [1, 12], whereas incorporating too many modes requires additional computation resources, thus, nullifying the benefits of dimensionality reduction. Moreover, the optimal number of modes is not priorly known [12], and creating datasets for training GANs can be challenging [2].

Furthermore, all aforementioned methods lacked the incorporation of physics-associated features. This issue could be handled by supervised Karhunen-Loeve Expansion, as was demonstrated by [13], who combined KLE of the shape modification vector with low-fidelity physical variables, such as pressure distribution, wave elevation, ship resistance, in the form of distributed and lumped physical parameter vectors. However, the necessity of direct evaluation of physics quantities and their gradients resulted in a computationally demanding model [2].

Another study related to our topic applied a conditional variational autoencoder (CVAE) for designing an airfoil [14]. Their initial hypothesis was that this method would enable them to evaluate the relationship between the geometry of aerodynamic shapes and their aerodynamic performance. A conditional variational autoencoder (CVAE) is an extension of the standard variational autoencoder (VAE) that incorporates additional conditional information into the model. It allows for the generation of data samples conditioned on specific attributes or labels. By integrating this conditional information into both the encoder and decoder, the CVAE can learn a richer latent representation that captures both the variability of the data and the conditioning factors. Aerodynamic part's shapes were fed as an input, while their respective performance data like a lift coefficient were fed as a label of a continuous character. The model was generated after specifying the latent vector and label [14]. The model was trained to minimize the latent and reconstruction loss, however, the results of this study demonstrated that the networks considered optimal in terms of CVAE's model did not always coincide with networks reproducing the optimal aerodynamic performance.

One more variation of GANs was applied by [15] which was named Performance Augmented Diverse Generative Adversarial Network (PaDGAN), which the authors claimed to be able to produce samples with 28% higher mean quality score with higher diversity compared to ordinary GANs. The quality score was evaluated based on the correlation of this model's lift-and-drag ratio with the ratios acquired from XFOIL for the same airfoils. Despite the positive results, the limitation of this method was its being supervised [2], which implies that the performance labels required to be evaluated over a large dataset of shapes. In case of the dataset's unavailability, its generation tends to be a computationally intense procedure.

It was mentioned in the previous section that this study was motivated by the works of Khan et al. (2022) [2] and Masood et al. (2023) [1]. The methodology proposed by the present study mainly follows the procedure described in those research with slight variations, which will be discussed further.

The approach proposed by [2] exploits KLE which was first introduced in the supervised model of [13]. However, while the latter tried to combine low-fidelity physical characteristics of shapes with shape geometry to create latent space, the former used geometric moments instead of physical data. The rationale for this decision was that, although geometric moments may not be as accurate in predicting aerodynamic characteristics of shapes as other more specific properties of the design, they are less resource-demanding and can significantly improve the latent space by making it physics-informed. Also, the negative effects of reducing the dimensionality of the data space can be minimized by applying "filters" including the required level of diversity, variance, and validity.

Testing of the model was conducted on 3D wing and ship hull models. The results of the former will be discussed further. The authors based their wing model on NACA 2410 airfoil [2], which was defined by 12 profile parameters, such as thickness, chord length, root/tip angles, etc. The wing parameters were kept fixed with a span length of 1.2 meters and a sweep angle of 4.29°. The surface of the wing was discretized with 90×25 nodes.

The subspace dimensionality required for the percentage of variance preserved by each subspace to reach the 95% threshold for the physics-informed case was approximately 10 regardless of the degree of geometric moments. Whereas, for only shape-informed cases this value was about 14. While the average diversity did not differ between these cases, the contrast in the average number of invalid designs was impressive: for physics- and shape-informed subspaces this value ranged between 10-12%; for shape-informed subspaces, it reached 43%.

Although the approach proposed by [2] demonstrated the effect of incorporating geometric moments into the creation of the reduced dimensionality latent space well, the study examined a very limited database of wing models, which might be one of the reasons for such a drastic divergence between the percentage of invalid designs.

The methodology proposed by [1] is very similar to the previously discussed. The main differences from the previously discussed approach are, firstly, it focuses on the airfoil profile instead of a wing model, secondly, the database includes a vast number of existing profiles and was enriched by slightly differentiating them. Although the methodology underlying both approaches was almost identical, their results substantially diverged. The diversity levels in [1] did not experience any significant change, while the percentage of self-intersecting designs, i.e., invalid designs, differed according to the type of the latent space and method of discretization (point distribution). Also, the dimensionality of latent space required for surpassing the 95% variance threshold decreased from 10-14 in the previous study to 3-5 in this one, with physics-informed spaces with uniform spacing of points being on the lower side of the range.

The main limitation of this study is that the authors analyzed purely airfoils and airfoil profile parameters. Although the results of this approach are applicable in the development of airfoils, the absence of wing analysis significantly limits the range of its application.

2.2 Theoretical basis

Parametric modeling is a fundamental concept in computer-aided design (CAD) and engineering, which provides a systematic approach to representing and manipulating geometric shapes using mathematical parameters. At its core, parametric modeling involves defining geometric entities, such as points, lines, curves, and surfaces in terms of mathematical equations or functions. These equations are parameterized, meaning they contain variables that can be adjusted to control the shape, size, and orientation of the modeled object [16, 17].

One of the key advantages of parametric modeling is its ability to capture design intent and facilitate design exploration by using parameters and constraints. Parameters, such as dimensions or angles, act as adjustable variables that govern the geometry of the model. By modifying these parameters, designers can quickly iterate through different design alternatives and evaluate their performance against specified criteria. Constraints are mathematical relationships that enforce design requirements and relationships between geometric elements. For example, a constraint may specify that two lines must remain parallel or that a certain distance must be maintained between two points. Parametric modeling also supports parametric dependencies, where the value of one parameter is dependent on another. This enables the creation of dynamic relationships within the model, allowing changes made to one parameter to propagate and affect other related parameters accordingly. As a result, parametric models are inherently flexible and responsive to design changes, facilitating rapid prototyping and design iteration [16].

In addition to design experimentation, parametric modeling enhances design automation and documentation. By capturing design intent through parameters and constraints, parametric models can be easily modified and updated, reducing the time and effort required to make design revisions. Furthermore, parametric models can generate accurate and consistent engineering drawings and documentation automatically, ensuring that design changes are reflected accurately throughout the design process [18].

Geometric moments are mathematical descriptors used to characterize the shape and spatial distribution of objects in an image or geometric space. They are computed by integrating the product of image intensity values (or geometric coordinates) with specific polynomial functions over the entire image or region of interest. Geometric moments provide valuable information about the centroid, area, orientation, and higher-order shape characteristics of objects [19]. They are widely used in image processing, computer vision, pattern recognition, and shape analysis tasks. By analyzing the geometric moments of objects, researchers can quantify and compare their shapes, enabling tasks such as object classification, shape matching, and object recognition. Although geometric moments offer less insight compared to evaluating the actual performance of wings via computational simulation, the latter often requires significant computational resources and time. Therefore, integrating geometric features with moments to construct the latent space produces a varied and robust design space that contains physics-associated descriptions of wings.

Chapter 3

Methodology

chap: methodology

This part of the report includes a brief formulation of the problem, and covers the procedure, software used, variables, and parameters used in the study.

3.1 Problem formulation

For applying the proposed method of dimensionality reduction, a rich design space of airfoil profiles {C} is assumed, which can be modified or represented through a design vector $v \in V$. The design space V is defined by a set of limits that constrain it to physically and geometrically valid airfoil profiles, i.e., $V = \{v \in \mathbb{R}^n : g_i(v) \leq 0,\, i = 1, \ldots, m\}$, where $g_i(v),\, i = 1, \ldots, m$ corresponds to the functional limits of the requirements for the design. Furthermore, if optimizing the design is an objective, we make an assumption of at least one performance characteristic, e.g., $f : \mathcal{V} \rightarrow \mathbb{R}$, that determines the performance $f(v)$ of the respective design encoded by v. Thus, we can state the simplified design optimization problem that aims at maximizing the performance characteristic as:

$$\text{Find } v^* \in \mathcal{V} : f(v^*) = \max_{v \in \mathcal{V}} f(v). \tag{1}$$

The cost of computational resources required for solving even the simple shape optimization problem provided above rises exponentially with each dimension of V and increases even quicker in case the calculation of the performance characteristic $f(v)$ is complex and demanding in time [1]. In this study, we create a space {C} initially rich with airfoil profile designs collected from public databases. The process of creating the data space is further discussed along with the required discretization instruments used in the determination of **v** and its augmentation with geometric moments. Then, we will unfold the proposed technique for dimensionality reduction.

3.2 Procedure

3.2.1 Data set

Our approach to the collection of airfoil profile database was based on publicly available airfoil designs available at [20], the total number of which proved to be moderately sufficient for our study and included many popular families of foils. This method was considered more rational than creating an arbitrary database using one of the families as the baseline and enriching it by modifying it. This way we acquired a database consisting of more than 1200 foil profiles arranged in the form of sets of points. However, the varying cardinality of sets of points posed a new issue related to the absence of standardization, i.e., the lack of a common number of points describing each foil profile. This problem was tackled by applying parametric modeling, which allows a generation of each airfoil profile's continuous high-accuracy approximation in the form of the NURBS curve. Following this, the NURBS curve can be discretized into any necessary number of points. Then, we created a database of 3D wings consisting of 4 identical airfoil profiles, i.e., consisting of 3 sections, with some fixed parameters, including a sweep angle of 5°, wing span of 10 m, chord length of 1 m equal at both root and tip.

3.2.1.1 Parametric modelling

In this study, we used the parametric model that was first introduced by [21] and later extended by [17] for generating original design space. Various techniques of airfoil parametrization were covered in the Literature Review chapter, where we reviewed and discussed their strengths and shortcomings. Nevertheless, the use of the parametric modeling developed by [17] was justified by the specific needs of this research, namely, the future perspective of applying the findings in design optimization, where such features

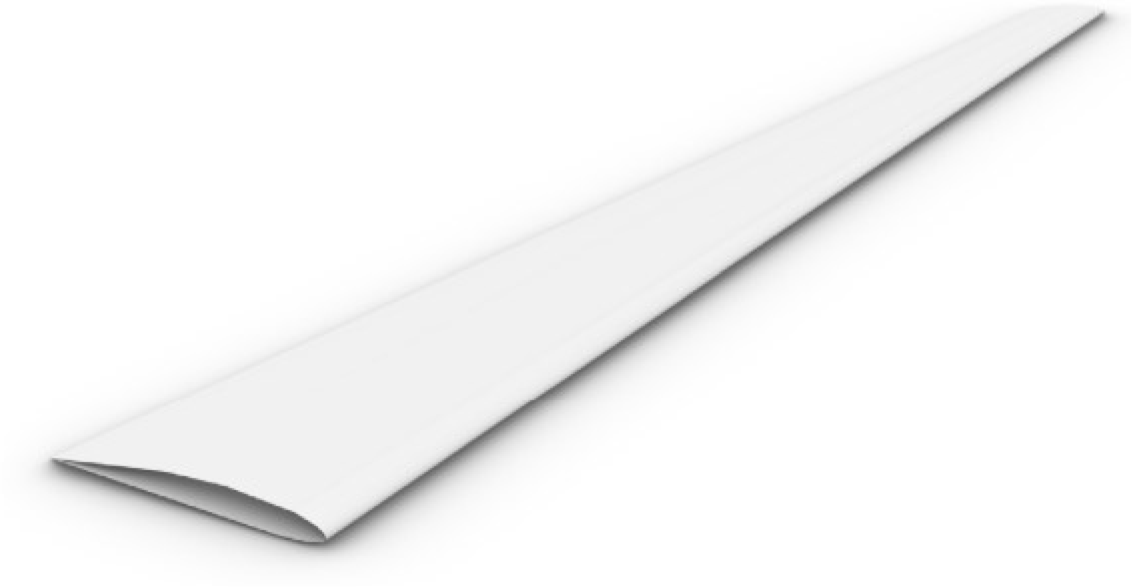

Figure 3.1: Sample wing.

of that method like the relatively short list of parameters that represent the profile shapes with a high accuracy might be pivotal.

The method that was chosen uses a baseline parametric model that includes 11 parameters, which, despite being relatively small in number, accurately describe approximately 70% of the airfoil profiles from the UIUC database [17].

Each airfoil instance generated by the parametric model is described by the following k-order NURBS curve [1]:

$$C(t; v) = (x(t; v), y(t; v)) := \sum_{i=0}^{n} b_i(v) R_{i,k}(t), \quad t \in [t_{k-1}, t_{n+1}], \tag{2}$$

Here, $\{R_{i,k}(u)\}_{i=0}^{n}$ is the rational B-spline basis set of degree $= k - 1$ that is defined over the sequence of knots $\{t_0, t_1, \dots, t_{n+k}\}$, and that possesses weights of $\{w_i\}, i = 0, 1, 2, \dots, n$, whereas $b_i(v)$ are the respective control points, which are determined by the v parameter vector. In the present study, cubic representation is used, and $k = 4$. There are 17 airfoil profile parameters along with 13 control points.

Design discretization As it was mentioned earlier, the parametric model creates curve representations in a continuous manner, which requires a discretization for applying the dimensionality reduction technique proposed in this study. In this case, discretization is done by converting continuous curves into approximations of a polygonal form. These polygonal approximations can be stored as a point coordinates vector. In analyzing performance, it's crucial to ensure we have enough points distributed evenly along the foil's profile. This ensures that performance metrics like drag and lift estimation yield consistent results. In our study, we used the findings of [1], who exploited XFOIL for evaluating foil performance and found that using 160 points, distributed based on curvature, generally gives reliable results. However, we also need to consider how the number and distribution of points affect the accuracy of the dataset used for dimensionality reduction. To reduce geometric deviation, we opted for a higher number of points, 200, and examined how different distributions affect results in four specific cases:

1. Uniform Spacing (Parametric Domain): We evenly distribute 200 parametric values, $\{t_i\}$, across the parametric domain $[t_{k-1}, t_{n+1}]$, as shown in Eq. (2). These 200 points are then used to define the profile. However, it's important to acknowledge that the distribution of points on the profile curve is not uniform, as illustrated in Fig. 3.2a.

2. Cosine Spacing: The cosine function is applied for adjusting Eq. (2) NURBS curve's parameterization. This adjustment concentrates points near the leading and trailing edges of the profile, as depicted in Fig. 3.2b.

3. Curvature-Based Spacing: Here, we utilize the curvature of the profile to determine the distribution of parametric values. This method leads to a higher concentration of points close to the leading edge and guarantees that the curvature integral over each of the parametric sections has the same value, as shown in Fig. 3.2c.

4. Uniform Spacing (Physical Domain): This approach involves discretizing the profile by dividing its curve length into 199 equal-length segments. Each segment represents an equal arc length along the curve, as depicted in Fig. 3.2d.

All of these discretization methods are applied to define the design space. Their impact on the process of dimension reduction and the validity of the resulting latent space is thoroughly examined in the Results and Discussion Chapter with a detailed comparison.

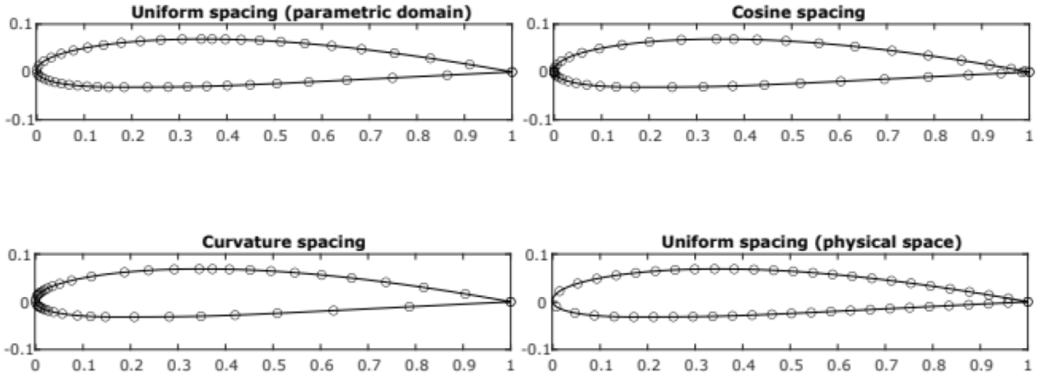

Figure 3.2: Point distribution techniques [1].

fig:3.2

3.2.2 Computation of geometric moments

Traditional techniques for reducing the dimensionality of design spaces often rely solely on the geometric representation of designs, such as point sets or polyhedral approximations. However, this approach often falls short of preserving the shape's full complexity and underlying structures, resulting in subspaces that lack diversity and struggle to create valid shapes. Building on the work of [2] and [1], we enhance the design description by incorporating integral properties of shape geometry. By combining geometric information from discretized profiles with geometric moments, we create an enhanced design vector that more accurately captures the underlying characteristics of shapes and provides a signature unique for each. Additionally, integrating geometric moments serves the dual purpose of introducing features that are correlated with aerodynamic performance criteria in a resourceful and effective manner. While many performance metrics, such as lift and drag, typically require expensive simulations, several of these quantities can be evaluated with significantly lower computational intensity due to being correlated to the geometric moments.

To create the unique Shape Significant Vector that possesses both the geometry of the shape and its geometric moments represented by the equation $P(\tilde{v}, t) = (G(\tilde{v}, t), M(\tilde{v}, t))$, where $M(\tilde{v}, t)$ is the lumped geometric moment vector and $G(\tilde{v}, t)$ is the continuous shape modification vector, we apply the 3D body $\mathcal{R}$'s finite moments number that are defined by the equation below [2] :

$$M_{p,q,r} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q z^r \rho(x, y, z) d\Gamma, \quad p, q, r \in \{0, 1, 2, \ldots\}, \tag{3}$$

This equation calculated the Γ's geometric moments of order s, where s is a non-negative integer equal to $p + q + r$. Assuming the s integer that was defined, we consider that all $M_{p,q,r}$ moments where $p + q + r = s$. and $\rho(x, y, z)$ are $\begin{cases} 1 & \text{if } (x, y, z) \in \Gamma \\ 0 & \text{otherwise} \end{cases}$ are contained in the vector $\tilde{M}^s$. Typically, computer-aided design packages use 0th $(M_{0,0,0})$ and 1st $(M_{1,0,0}, M_{0,1,0}, M_{0,0,1})$ order moments in evaluating the volume of the object and its centroid, respectively [2]:

$$V = M_{0,0,0},$$
$$\vec{c} = \{c_x, c_y, c_z\} = \left\{ \frac{M_{1,0,0}}{M_{0,0,0}}, \frac{M_{0,1,0}}{M_{0,0,0}}, \frac{M_{0,0,1}}{M_{0,0,0}} \right\}$$

Moreover, if the $\rho(x, y, z)$ is a probability density function of a continuous random variable, it means that M^0 is the volume, M^1 represents the mean and M^2 represents the variance of the random variable.

Various methods are applied in calculating geometric moments, Gauss' divergence theorem being the most prevalent approach [22, 23, 2]. This theorem converts volume integrals into surface integrals. Here, we provide a brief explanation of how geometric moments are evaluated by implementing the divergence theorem for $S = \bigcup_{i=1}^{N} T_i$, which approximates the surface bounding Γ. The number of triangles in S is represented by N here.

Geometric moments typically change with rigid and non-rigid transformations like translation, rotation, and scaling [24]. However, many physical quantities remain unaffected by one or more of them. That is why, prior to including geometric moments in

$P(\tilde{v}, t)$, it's essential to ensure they are invariant to translation and scaling. The discussion below focuses on geometric moments that remain unchanged by translation and scaling, particularly when considering moments about the centroid $c(\Gamma) = (C_x, C_y, C_z)$ of Γ. These are known as central geometric moments and are defined as follows based on Eq. (3):

$$\mu_{p,q,r} = \int_{\Gamma} (x - C_x)^p (y - C_y)^q (z - C_z)^r d\Gamma.$$

To proceed with the calculation of scaling invariance of $\hat{\mu}^{p,q,r}$, we make an assumption that Γ is scaled by a γ factor uniformly, which results in [2]:

$$\hat{\mu}_{p,q,r} = \gamma^{p+q+r+3} \mu_{p,q,r}.$$

Here, we conclude with the following equation:

$$M\hat{I}_{p,q,r} = \frac{\mu_{p,q,r}}{(\mu^{0,0,0})^{1+\frac{(p+q+r)}{3}}}$$

which is Γ's invariant geometric moment under uniform translation and scaling [24].

3.2.3 Dimensionality reduction

As discussed in Chapter 1, the computational expense of geometry optimization rises significantly as the dimension of V expands. It escalates even higher when the calculation of the performance index f becomes complex. To address this issue, our current study aims to overcome the challenge of high dimensionality using feature extraction techniques through dimensionality reduction. Typically, it involves extracting latent features from V to reduce its dimension while preserving as much geometric variability as possible in the newly produced domains. However, because of the limitations associated with the standard design space dimensionality reduction method, we aim to create a space with latent variables that extend beyond those extracted from V. Corresponding geometric moments calculated from the shape geometry are one of these variables.

To create this physics-informed space, we assume that along with the continuous shape modification vectors $\tilde{G}(\tilde{\mathbf{v}}, t)$, there is a lumped vector containing the geometric moments

$M(\tilde{\mathbf{v}}_M, t) \in \mathbb{R}^n_M$ with $n_M = 1, 2, \ldots$. The latter vector corresponds to an arbitrary point $\tilde{\mathbf{v}}_M$, and has a null measure. We further make an assumption that G and M are the definition domains for $\tilde{G}(\tilde{\mathbf{v}}, t)$ and $M(\tilde{\mathbf{v}}_M, t)$, respectively. Now we have a vector that combines geometric and moments features:

$$P(\tilde{v}, t) = \left(\tilde{G}(\tilde{\mathbf{v}}, t), M(\tilde{\mathbf{v}}_M, t)\right).$$

This vector constructs a unique function of the shape signature vector containing the baseline design information. Moreover, consider that $P(\tilde{v}, t)$ is part of a Hilbert space described by the inner product as illustrated in Figure 3.3 [2]:

$$\langle a, b \rangle_f = \int_P f(v)\mathbf{a}(v) \cdot \mathbf{b}(v) dv$$

$$= \int_G \tilde{f}(\tilde{\mathbf{v}})\tilde{\mathbf{a}}(\tilde{\mathbf{v}}) \cdot \tilde{\mathbf{b}}(\tilde{\mathbf{v}}) d\tilde{\mathbf{v}} + f(\tilde{\mathbf{v}}_M)\mathbf{a}(\tilde{\mathbf{v}}_M) \cdot \mathbf{b}(\tilde{\mathbf{v}}_M),$$

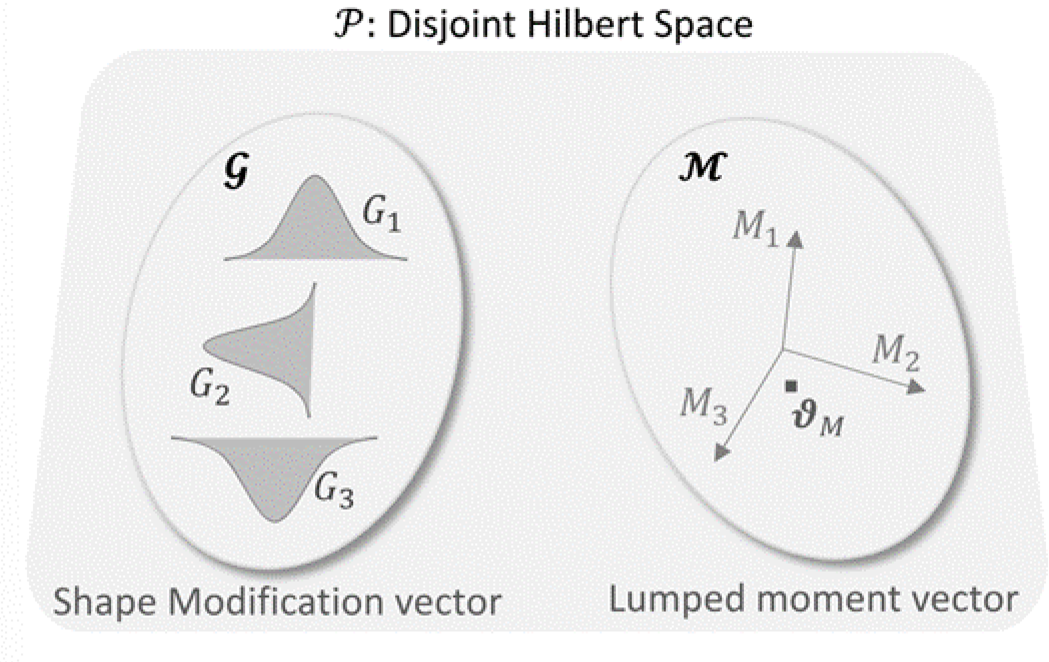

Figure 3.3: Domains for $\tilde{G}(\tilde{\mathbf{v}}, t)$ and $M(\tilde{\mathbf{v}}_M, t)$, in a disjoint Hilbert space [2]. fig:3.3

The process of identifying the best design may face uncertainties of knowledge [7]. To address this, we can view t as part of a stochastic space V, which has a probability density function $p(t)$. This function indicates the likelihood of finding the optimal design within V. Defining $p(t)$ appropriately can be challenging, so it's common to start with a uniform distribution function. This means that any specific value of **t** has an equal chance of being the optimal solution t^*. After defining $p(t)$, we can calculate the mean and variance of the shape signature vector as follows:

$$\langle \mathbf{P} \rangle = \int_T \mathbf{f(v)}\mathbf{P}(\tilde{\mathbf{v}}, \mathbf{t})p(t)\mathbf{dt},$$

$$\sigma^2 = \left\| \langle \mathbf{P} \rangle^2 \right\| = \int_T \int_P f(\mathbf{v})\mathbf{P}(\tilde{\mathbf{v}}, \mathbf{t}) \cdot \mathbf{P}(\tilde{\mathbf{v}}, \mathbf{t})p(\mathbf{t})\mathbf{dtdt},$$

Here, $\tilde{\mathbf{P}}$ is SSV's deviation from the mean. The objective of dimension reduction is to determine the representation of $\mathbf{P}(\tilde{\mathbf{v}}, \mathbf{t})$, with lower dimensionality, i.e., $\tilde{\mathbf{P}}(\tilde{\mathbf{v}}, \mathbf{V})$ that depends on a geometrically-functionally active latent variable (GFALV) vector **v** instead of **t** [2].

Once we've built the SSV vector using invariant geometric moments, we apply KLE. The goal of KLE is to discover the best set of orthonormal functions to linearly represent the SSV. This process aims to achieve the following:

$$\tilde{P}(\tilde{v}, t) \approx \sum_{i=1}^{m} v_i \phi_i(\tilde{v}),$$

Here, $\{\phi_i(v)\}_{i=1}^{m}$ are orthonormal functions that form the basis of the subspace V, which will preserve the variance in moments and shapes to the extent possible.

KLE's optimal condition is to create functions preserving the highest geometric variance σ^2:

$$\sigma^2 = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \langle v_i v_j \rangle (\phi_i(\tilde{v})), \phi_i(\tilde{v})_f$$

$$= \sum_{j=1}^{\infty} |v_j|^2 = \sum_{j=1}^{\infty} \langle \left(\tilde{P}, \phi_i(\tilde{v}) \right)_f^2 \rangle.$$

Next problem's solution subject to $(\omega(\tilde{v}), \omega(\tilde{v}))_f^2 = 1$, provides the basis that retains the highest variance [7]:

$$\min_{\omega \in L_f^2(\tilde{P})} J(\omega(\tilde{v})) = \langle (\tilde{P}, \omega(\tilde{v}))_f^2 \rangle$$

Additionally, eigenvectors correlate with KL modes, eigenvalues represent the variance:

$$\sigma^2 = \sum_{i=1}^{\infty} \lambda_i,$$

Now, we select k eigenvectors as a foundation of the approximation. The choice of k is decided by the necessary ϵ (where $0 < \epsilon \leq 1$) of variance. In simpler terms, k is so picked that the chosen eigenvectors enclose minimum $\epsilon \times 100\%$ of the data's total variance [2].

3.2.4 Latent space

In addition to reducing dimensionality to create the latent space, we must also consider the possibility of using parameter bounds. These bounds restrict exploration to a permissible space that contains minimal or no infeasible or invalid designs. This step is crucial because, while we ideally want to examine a vast space, the presence of invalid geometries in it can hinder subsequent design optimization efforts. Typically, functional or side limits outlining the optimal design spaces determine these bounds. In the original design space, the design vector v usually includes features with physical significance, making it relatively straightforward for designers to introduce constraints [1].

However, there is an issue with this method - the resulting optimization problem becomes complex. This complexity is common in constraint optimization when numerous functional limits are involved, making the optimization process time-consuming. Side constraints, such as parameter value bounds, are easier for optimization algorithms to handle and are preferable over complicated functional constraints. Nonetheless, even when only side constraints are considered, challenges remain, and determining these side constraints is tricky for designers. Unlike **v**, which represents physical parameters like lengths and angles, **u** is a latent design vector with no direct physical interpretation. Consequently, it's difficult to determine the lower and upper values that ensure designs within the feasible region of U are also valid in V. In essence, bounding U in a way that satisfies all design constraints and requirements in the original space poses a significant challenge.

A common approach to setting bounds in the latent space is by using the standard deviation from the mean shape, which is located at the design space's center. This technique strikes a balance between the amount of invalid geometries and the degree of allowable variation, thus, we adopt this method. Here the limits of the ith element of the latent variable are defined as:

$$u_i \in \left[-\sqrt{\alpha\lambda_i}, \sqrt{\alpha\lambda_i}\right],$$

3.2.5 Design space quality analysis

subsection: 3.2.5

To assess how well a latent space can generate a wide variety of diverse and valid shapes, we need to introduce suitable metrics. One commonly used metric for diversity is the Hausdorff distance [25], which is often used to measure the distance between two sets in a metric space. It can also be used to quantify the similarity or dissimilarity between two free-form surfaces. In our scenario, we assume both shapes can be discretized using a finite set of points. Then, using the following method we can evaluate the Hausdorff distance H between the finite sets {C1 and {C2} [1]:

$$H(\tilde{C}_1, \tilde{C}_2) = \max\{h(\tilde{C}_1, \tilde{C}_2), h(\tilde{C}_2, \tilde{C}_1)\}, \text{where}$$

$$h(\tilde{C}_1, \tilde{C}_2) = \max_{c_1 \in \tilde{C}_1} \min_{c_2 \in \tilde{C}_2} \|c_1 - c_2\|,$$

Here, $h(\tilde{A}, \tilde{B})$ is the H directed from set $\tilde{A}$ to set $\tilde{B}$, and $\|\cdot\|$ denotes the Euclidean norm.

The measure of the diversity of the latent space is then defined as the average H between the densely sampled designs acquired from the latent space and the reference shape. Higher diversity values are preferable because they indicate the richness of latent spaces. Nevertheless, high diversity also means that the probability of invalid shapes, including self-intersections, oscillatory behavior, etc. is also increased. We use the definition of validity provided by [1], which describes it as the ratio of invalid designs to valid designs collected in the latent space from the dense sampling.

3.3 Software use

The main software package used for programming purposes was MATLAB. It is a powerful software environment commonly used in industry and academia for computing and programming developed by MathWorks [26]. It offers a wide range of functions and toolboxes for various applications such as data analysis, simulation, and image processing. Its user-friendly interface and rich library of built-in mathematical functions make it a popular choice for conducting scientific research and solving complex mathematical problems.

During the present study, we used a wide range of functionality of this software. The matrix-based essence of the language provides an opportunity to navigate easily through numerous functions that we created for our use by applying the set of embedded mathematical instruments. Also, it excels at numerical computation, which is critical for software and language exploited for machine learning purposes.

However, MATLAB is mildly limited when it comes to working with 3D designs, again, owing to its matrix-based nature. It still offers several tools for 3D visualization, modeling, and analysis, but is substantially inferior to software specialized in it. One of such alternatives that we considered using and used for some period was the Rhinoceros 3D and Rhino engine. The main advantages of Rhino were that it supported built-in tools for analyzing 3D objects like wings, and could facilitate the calculation of geometric moments and application of parametric modeling [27]. Despite possessing these features, the intuitive understandability and user-friendly interface of MATLAB made it a better environment. Moreover, the Rhino engine was especially uncomfortable for complex numerical analysis that it was not specialized to work with.

3.4 Variables

3.4.1 Number of points on each wing

When assessing performance, it's vital to have enough number of points evenly spread across the foil's profile. This ensures that metrics like drag and lift estimation provide consistent outcomes. Our research builds on [1], which utilized XFOIL to evaluate foil performance and concluded that using 160 points distributed according to curvature typically produces reliable results. However, we also need to account for how the quantity and arrangement of points impact the accuracy of the dataset used for dimensionality reduction. To minimize geometric deviation, we chose to increase the number of points to 200.

3.4.2 Number of sections of wing

In the preliminary stage of the project, we used the XFLR5 tool to determine the minimal number of sections to divide the wing that would not hamper the accuracy of aerodynamic performance characteristics. We created a sample wing with a continuous surface, which was then divided into an increasing number of segments. The results of this brief experiment are presented in Table 3.1. We initially decided to use five airfoils that would divide the wing into four sections, but later reduced it by one, as the difference in the performance results was negligible compared to the complexity of the wing database collection.

Table 3.1: Table from the interim report

tab:3.1

No. of Sections	Lift Coefficient	Drag Coefficient	Lift-to-Drag Ratio
3	0.30166	0.00914	33.49234
4	0.31046	0.00911	34.07903
5	0.30370	0.00914	33.23895
6	0.30176	0.00914	33.02188
7	0.30468	0.00910	33.47802
8	0.30798	0.00905	34.02649
9	0.30378	0.00914	33.23654

3.4.3 Point distribution method

The four methods of point distribution along the curve that we discussed previously were cosine spacing, uniform spacing over physical and parametric domains, and curvature-based spacing. It was decided to use all these methods to compare them, thus, the method of point spacing on the airfoil profile curve was one of the parameters that we applied. Their performance will be covered and compared in the Results and Discussion section of the report.

3.4.4 Parameters included in the model

Along with the aforementioned fixed wing parameters – chord length, span, sweep angle – we included 11 airfoil profile parameters, which were introduced by [21]:

Table 3.2: Parameters' definition

tab:3.2

Nr.	Name	Description	Symbol	Actual Range
0	Length	Length of foil's chord	L	free
1	Upper-side max width	Maximum width of suction side w.r.t. Chord	u_max	$[0,1] \rightarrow \left[\frac{L}{500}, \frac{L}{5}\right]$
2	Upper-side max width position	Longitudinal position of suction side's max width	x_u_max	$[0,1] \rightarrow \left[\frac{L}{10^{10}}, \frac{7L}{10^{10}}\right]$
3	Upper-side angle	Suction's side angle at trailing edge w.r.t. Chord	a_b_u	$[0,1] \rightarrow [0, 90°]$
4	Upper-side tip shape	Leading edge upper part form factor	tip_u	$[0,1] \rightarrow [0.05, 1]$
5	Upper-side Inflection point position	Inflection point position	s_u	$[0,1]$
6	Lower-side max width	Maximum width of lower side w.r.t. Chord	l_max	$[0,1] \rightarrow \left[\frac{L}{500}, \frac{L}{5}\right]$
7	Lower-side max width position	Longitudinal position of lower side's max width	x_l_max	$[0,1] \rightarrow \left[\frac{L}{10^{10}}, \frac{7L}{10^{10}}\right]$
8	Lower-side angle	Suction's side angle at trailing edge w.r.t. Chord	a_b_l	$[0,1] \rightarrow [-a\_b\_l, a\_b\_l]$
9	Lower-side tip shape	Leading edge lower part form factor	tip_l	$[0,1] \rightarrow [0.05, 1]$
10	Lower-side Inflection point position	Inflection point position	s_l	$[0,1]$
11	Tangent angle at leading edge	The angle between the vertical axis and foil's tangent direction at the leading edge	a	$[0,1] \rightarrow [-20, 20°]$

Chapter 4

Implementation

chap: implementation This chapter covers the implementation of the proposed approach through MATLAB. All source codes are provided in Appendix A.

4.1 Reading the database table

After collecting the UIUC public airfoil profiles database into a spreadsheet, it was rearranged into an array through the code provided in subsection A1. The number of points on each profile was set to 200, and the number of profiles used for generating each wing was set to 4. Then a “vect” matrix was initialized that would contain the locations of points on wings profiles.

4.2 Parametric modeling

We used the parametric model developed by [17] and their MATLAB source code. Additionally, we assigned the fixed-wing parameters and defined the relations between wing and airfoil parameters.

4.3 Design discretization

We developed a discretizing function similar to the one introduced by [1]. It includes a simple switch for choosing the suitable method of point distribution along the curve from four techniques – uniform spacing in the parametric domain, uniform spacing in the physical domain, cosine spacing, and curvature-based spacing.

4.4 Computation of geometric moments

The computation of geometric moments and moment vectors was done using the “sthOrderGeometricMoment” and “sthOrderGeometricMomentVector” functions that were developed and published by [2].

4.5 Dimensionality reduction

At this stage, we employed the “KLE” function published by [2] to calculate KL modes and values and normalized the design grid to have unit variance.

Also, we created a “varRetained” variable that would accumulate the retained variance by acting as KL values sum and added a condition that would display the current dimension number and reduction percentage right after the “varRetained” variable had reached a required threshold of 95%.

4.6 Design space quality analysis

Self-intersection analysis was done using the “robustness” function and diversity evaluation was done by the “diversity” function, both of which were published by [2].

Chapter 5

Results and Discussion

chap: results and discussion

This section covers the results, including the efficiency of dimension reduction, latent space quality, and how these two are impacted by the method of point distribution along the curve and the inclusion of geometric moments.

5.1 Dimension reduction and creation of latent spaces

To assess the efficiency of the suggested approach, we explore latent spaces containing different design or shape signature vectors, and various types of discretization.

The effectiveness of the produced latent spaces is determined by the degree of dimensionality reduction achieved, while their overall quality is evaluated based on the criteria outlined in Subsection 3.2.5. Initially, we assess the reduction in space dimensionality and its correlation with the method of point distribution applied, as shown in Figures 5.1, 5.2, 5.3, and 5.4. Subsequently, we analyze the diversity and robustness of these spaces.

Table 5.1: Dimension reduction efficiency tab:5.1

Spacing method	**Dimensionality of subspace to achieve 95% variance**	**Percentage reduction**	**Mean square error**
Uniform spacing (Parametric domain)	5	54.55%	8.607e-05
Uniform spacing (Parametric domain) with second-order moments	4	63.63%	2.502e-05
Cosine spacing	5	54.55%	6.573e-05
Cosine spacing with second-order moments	4	63.63%	1.462e-05
Curvature-based spacing	6	45.45%	10.701e-05
Curvature-based spacing with second-order moments	5	54.55%	6.627e-05
Uniform spacing (Physical domain)	4	63.63%	0.414e-05
Uniform spacing (Physical domain) with second-order moments	4	63.63%	0.378e-05

The dimensionality of the initial space was 11, and its percentage reduction is presented in Table 5.1. We set the threshold minimum of the variance retained to 95% as was discussed previously, and were able to achieve an impressive reduction to 4-6 dimensions.

Evaluations of only shape-informed uniform spacing (parametric domain) and cosine spacing along the curve given in Figures 5.1 and 5.2, respectively, demonstrated very similar results, the curvature-based distribution (Figure 5.3) had a slightly worse performance. The best results were achieved through the uniform spacing in the physical domain (Figure 5.4).

Adding the physics-associated features by incorporating second-order geometric moments resulted in an overall increase in the efficiency of reduction. All spacing methods except the physical domain uniform spacing experienced a reduction in the minimum dimensionality required for reaching the threshold variance of 95% by one. The improvement needed to achieve this progress in efficiency was not substantial as can be seen from Figures 5.1, 5.2, and 5.3, where three curves corresponding to the spaces that experienced an improvement tend to be already close to having a lower dimension number than what is given in Table 3.2. Although the same can be observed from Figure 5.4, the curve of the uniform spacing in the physical domain seems to be the furthest from the 95% variance line. Nonetheless, here we can conclude that the incorporation of second-order geometric moments had a positive effect on the efficiency of dimensionality reduction while retaining high variance despite the improvement being mildly significant.

The mean square errors demonstrate the same trend – physics-informed spaces performed slightly better compared to the ones containing purely geometrical features.

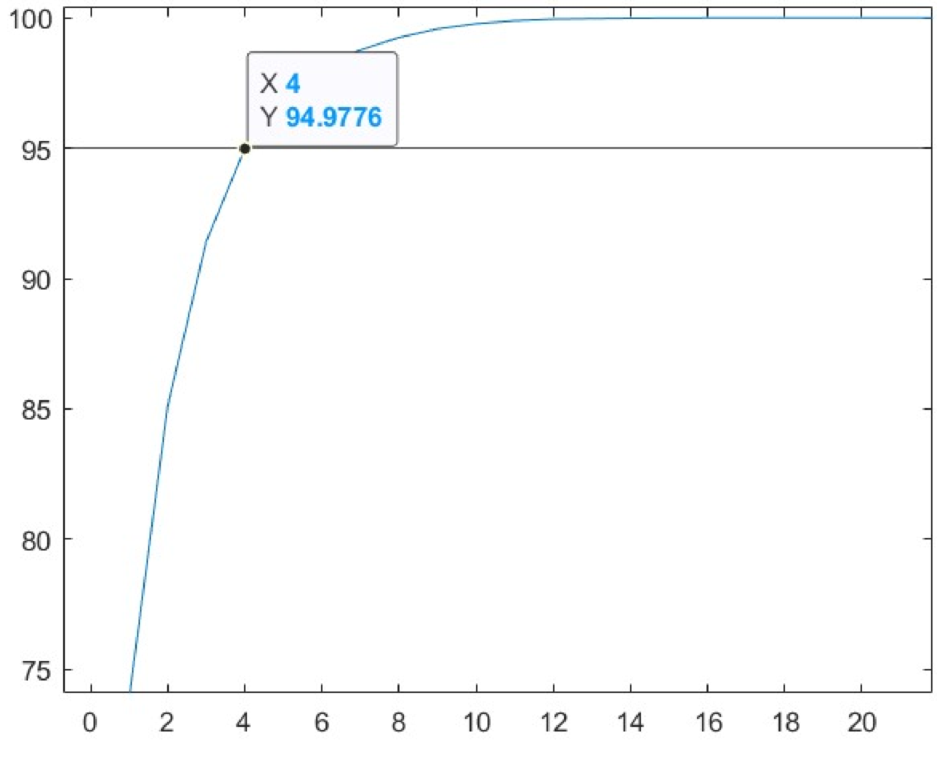

Figure 5.1: Variance [%] vs dimensionality, uniform spacing over the parametric domain. fig:5.1

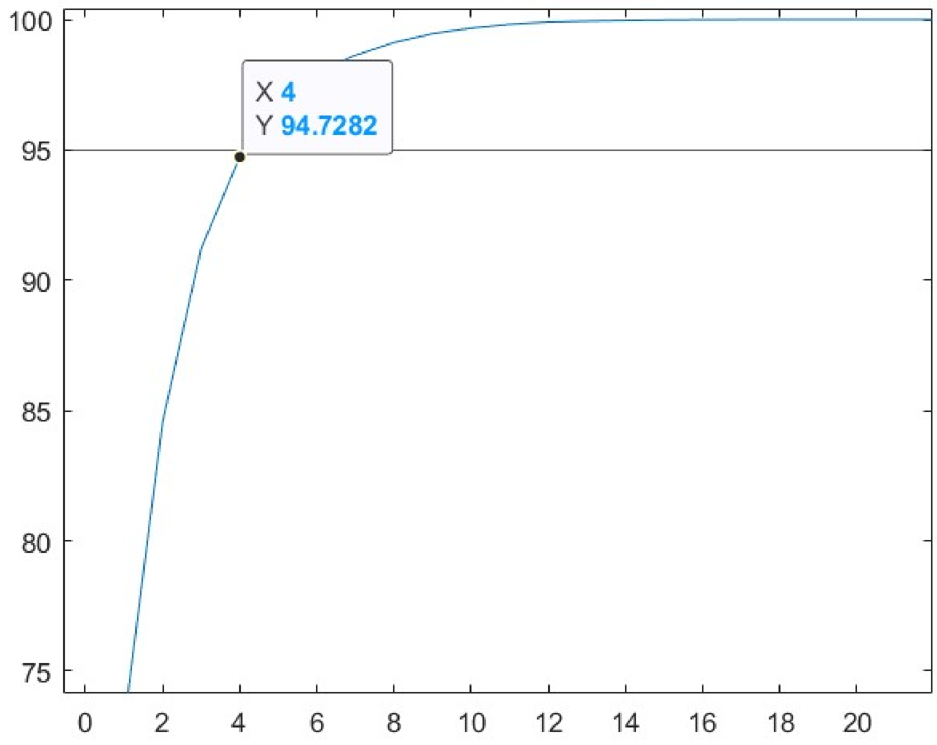

Figure 5.2: Variance [%] vs dimensionality, cosine spacing. fig:5.2

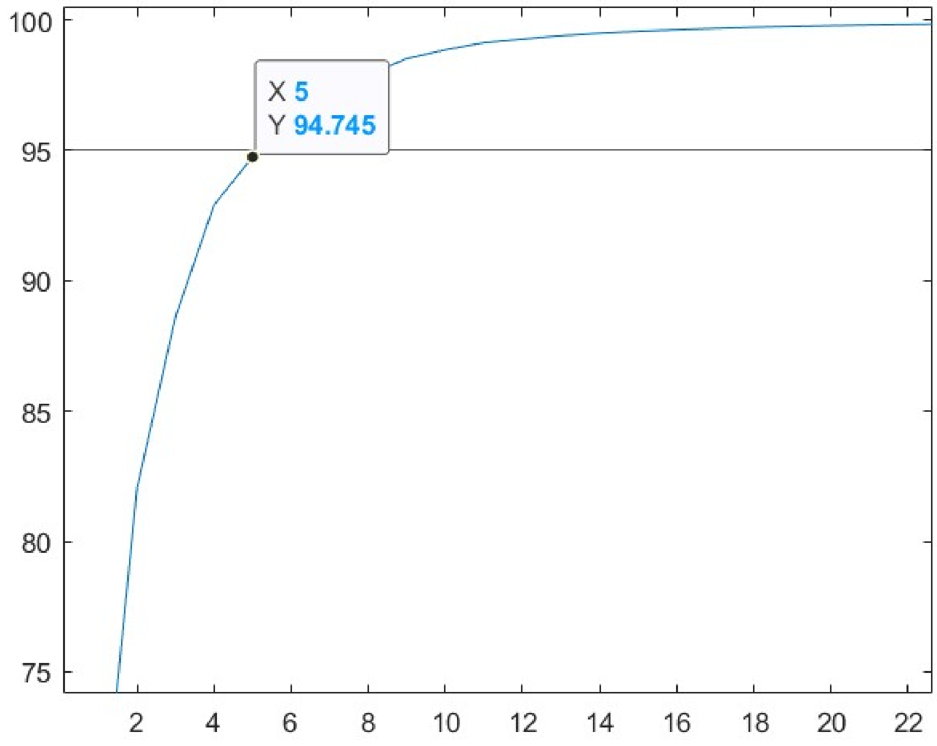

Figure 5.3: Variance [%] vs dimensionality, curvature-based spacing. fig:5.3

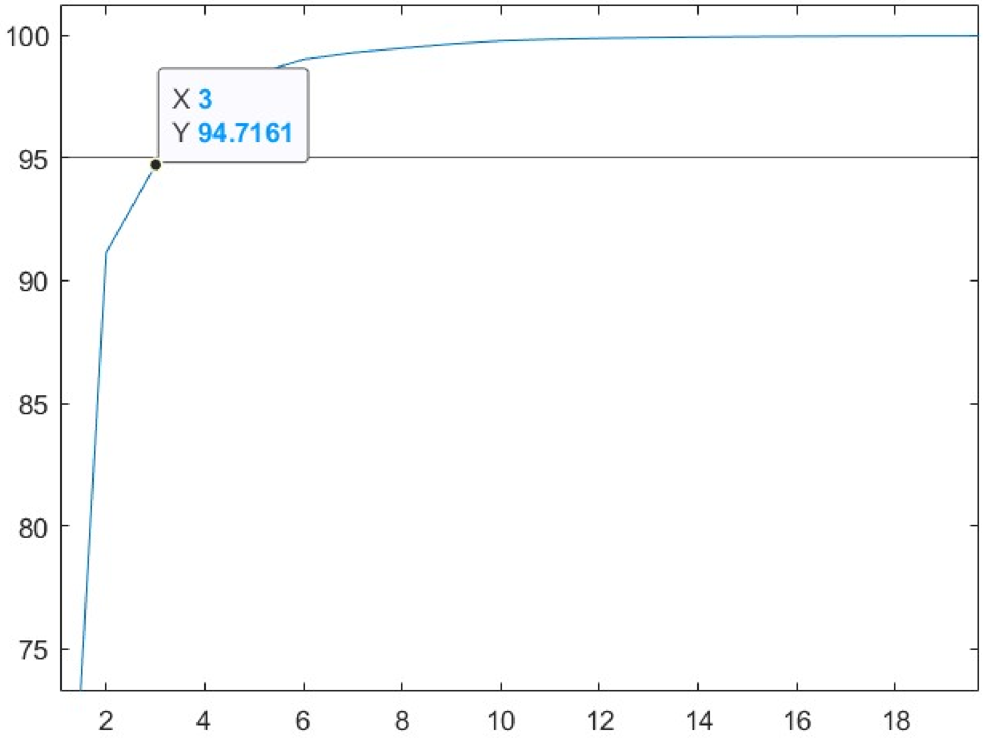

Figure 5.4: Variance [%] vs dimensionality, uniform spacing over the physical domain. fig:5.4

5.2 Latent space quality analysis

The objective of this analysis is to determine if the produced subspaces are appropriate for optimizing designs, which is one of the application perspectives of this study. We evaluate their potential to accurately describe the underlying shape using the latent parameter vector u and to produce diverse and valid geometries. Diversity is crucial for optimizing designs, for it measures the differences between shapes in the latent space. Robustness, or validity, indicates the ratio of invalid shapes in the latent space to the total number. Ideally, we aim to produce spaces with minimal invalid designs. Both metrics are equally vital for shape optimization because diverse designs and the absence of invalid shapes benefit the process. While robustness is associated with the average rate of invalid geometries in samples, diversity is related to the average distance between designs, as explained in Subsection 3.2.5. The obtained estimates for all discretization types are illustrated in Table 5.2.

Table 5.2: Latent space quality

tab:5.2

Distribution method	Diversity of space	% of self-intersecting designs
Curvature-based spacing	19.749	0
Curvature-based spacing with second-order moments	19.749	0
Uniform spacing (Physical domain)	27.907	0
Uniform spacing (Physical domain) with second-order moments	27.907	0

Table 5.2 demonstrates peculiar results for spaces that include second-order moments, as they have the same diversity values compared to purely geometric spaces. However, it needs to be mentioned that the results obtained by [1], which we discussed in the literature review, were of a similar manner to the results provided in Table 5.1. They analyzed 2D airfoil profiles instead of 3D wings.

The same could be applied to the self-intersection rates, as the results indicate their complete absence. There are two hypotheses on why we got these results. One of them implies that the reason is that we used the public database of existing airfoil profiles for generating wings not expanding it by, for example, randomly modifying the baseline design, which could potentially create those invalid designs. The other one is that there is an error in our implementation of the method, which most probably occurred at the latent space quality assessment stage.

Following this part, we will discuss the results presented in Table 5.1 assuming the first explanation, but, the close examination of this possible error will be listed in future works.

The difference in diversity between the two discretization methods – curvature-based spacing and uniform spacing on the physical domain – is significant, which is an indication of the better distribution of points along the airfoil profile in the latter one. Both self-intersection and diversity values seem to be insensitive to the incorporation of geometric moments. These results are in alignment with the outcomes reported by [1] and [2], although directly interpolating their results to the present study is impossible due to critical differences in setups.

Chapter 6

Conclusions

chap: conclusions

This study addressed the critical necessity for resource-effective optimization of engineering products possessing complex free-form surfaces, such as turbine blades and ship hulls. The curse of dimensionality, which is the exponential rise of the required computational intensity, poses significant challenges in optimizing these surfaces, negatively impacting product performance and competitiveness. By exploring dimensionality reduction techniques, this research proposed an approach that could potentially enhance optimization efficiency and product development. Through this study, we provided valuable insights into the practical application of dimensionality reduction methods in engineering design and aimed at eliminating the current gaps.

The main purposes of this research were to examine the feasibility of shape- and physics-informed dimensionality reduction and to evaluate the required number of dimensions that allow to retain the required level of variance, diversity, and validity. This study was motivated by [1] and [2].

The primary findings of this study were the impact of the method of discretization on the level of preserved variance of the space, and the effect of incorporating physics-associated features of geometric shapes in the form of geometric moments on various criteria, including variance, robustness, and diversity of the latent space. We concluded that the incorporation of 2nd order geometric moments had a positive effect on the efficiency of dimensionality reduction while retaining high variance despite the improvement being mildly significant, while its effect on the diversity and self-intersection rate were ambiguous. We provided the possible origins of the error causing this ambiguity and discussed those findings with the necessary assumptions priorly stated.

An extensive literature review was made before starting the study and the findings of related research were integrated into our work, which is one of its strengths, alongside its filling the knowledge gaps that appeared in the process of rapid development and innovation taking place in the field of designing and optimizing free-from functional geometries. However, weaknesses are also present in this study such as the already mentioned uncertainty in the interpretation of findings. Moreover, the model of the wing used in this research tends to be simple, which might hinder the application of the gathered knowledge on complex shapes. Nevertheless, findings obtained in the present work have the potential to be pivotal or act as a foundation for future works on the same or adjacent topic.

The initial hypothesis, which stated that the incorporation of geometric moments into the dimensionality reduction might result in increased efficiency and diversity of the generated latent space, has been partially backed by the obtained results. While the impact of this newly adopted technique on the quality of the retained spaces might not be remarkable, we can conclude that the effect is notable and positive.

As it was previously mentioned, the topic of this study has gaps, to the filling of which this study has contributed, and considering the increasing demand for applying modern tools and techniques in solving such complex problems as the optimization of free-form functional surfaces, this subject has bright prospects. The wide range of fields where free-form surfaces play a pivotal role implies that this study has a high degree of applicability.

The future work, besides introducing a more advanced wing model and expanding the database examined, also includes the development of an optimization model that would exploit the benefits acquired through the implementation of the findings of this study.

Bibliography

[1] Z. Masood, K. V. Kostas, S. Khan, and P. D. Kaklis, "Shape-Informed Dimensional Reduction in Airfoil/Hydrofoil Modeling," *Journal of Marine Science and Engineering*, vol. 11, p. 1851, Sep. 2023.

[2] S. Khan, P. D. Kaklis, A. Serani, M. Diez, and K. V. Kostas, "Shape-supervised Dimension Reduction: Extracting Geometry and Physics Associated Features with Geometric Moments," *Computer-Aided Design*, vol. 150, p. 103327, Sep. 2022.

[3] M. Köppen, "The curse of dimensionality," in *5th online world conference on soft computing in industrial applications (WSC5)*, vol. 1, pp. 4–8, Sep. 2000.

[4] N. Altman and M. Krzywinski, "The curse(s) of Dimensionality," *Nature Methods*, vol. 15, pp. 399–400, May 2018.

[5] D. C. Montgomery and G. Weatherby, "Factor Screening Methods in Computer Simulation Experiments," tech. rep., Institute of Electrical and Electronics Engineers (IEEE), 1979.

[6] I. M. Sobol', "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates," *Mathematics and Computers in Simulation*, vol. 55, pp. 271–280, Feb. 2001.

[7] M. Diez, E. F. Campana, and F. Stern, "Design-space dimensionality reduction in shape optimization by Karhunen–Loève expansion," *Computer Methods in Applied Mechanics and Engineering*, vol. 283, pp. 1525–1544, Jan. 2015.

[8] M. Belkin and P. Niyogi, “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation,” *Neural computation*, vol. 15, pp. 1373–1396, Jun. 2003.

[9] W. Yu, R. Wang, F. Nie, F. Wang, Q. Yu, and X. Yang, “An improved locality preserving projection with 1-norm minimization for dimensionality reduction,” *Neurocomputing*, vol. 316, pp. 322–331, Nov. 2018.

[10] W. Chen, K. Chiu, and M. D. Fuge, “Airfoil Design Parameterization and Optimization Using Bézier Generative Adversarial Networks,” *AIAA Journal*, vol. 58, pp. 4723–4735, Nov. 2020.

[11] X. Du, P. He, and J. R. Martins, “A B-Spline-based Generative Adversarial Network Model for Fast Interactive Airfoil Aerodynamic Optimization,” in *AIAA Scitech 2020 Forum*, Jan. 2020.

[12] J. Li and M. Zhang, “On deep-learning-based geometric filtering in aerodynamic shape optimization,” *Aerospace Science and Technology*, vol. 112, p. 106603, May 2021.

[13] A. Serani, F. Stern, E. F. Campana, and M. Diez, “Hull-form stochastic optimization via computational-cost reduction methods,” *Engineering with Computers*, vol. 38, pp. 2245–2269, Mar. 2021.

[14] K. Yonekura and K. Suzuki, “Data-driven design exploration method using conditional variational autoencoder for airfoil design,” *Structural and Multidisciplinary Optimization*, vol. 64, pp. 613–624, Jul. 2021.

[15] W. Chen and F. Ahmed, “PaDGAN: A Generative Adversarial Network for Performance Augmented Diverse Designs,” in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Jun. 2020.

[16] B. Kulfan and J. Bussoletti, “"Fundamental" Parameteric Geometry Representations for Aircraft Component Shapes,” in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Sep. 2006.

[17] K. Kostas, A. Amiralin, S. Sagimbayev, T. Massalov, Y. Kalel, and C. Politis, “Parametric model for the reconstruction and representation of hydrofoils and airfoils,” *Ocean Engineering*, vol. 199, p. 107020, Mar. 2020.

[18] D. Ziemkiewicz, “Simple Parametric Model for Airfoil Shape Description,” *AIAA Journal*, vol. 55, pp. 4390–4393, Dec. 2017.

[19] M. S. Hickman, “Geometric Moments and Their Invariants,” *Journal of Mathematical Imaging and Vision*, vol. 44, pp. 223–235, Dec. 2011.

[20] “UIUC airfoil coordinates database.” https://m-selig.ae.illinois.edu/ads/coord_database.html. Accessed: 2024-04-24.

[21] K. Kostas, A. Ginnis, C. Politis, and P. Kaklis, “Shape-optimization of 2D hydrofoils using an Isogeometric BEM solver,” *Computer-Aided Design*, vol. 82, pp. 79–87, Jan. 2017.

[22] A. Krishnamurthy and S. McMains, “Accurate GPU-accelerated surface integrals for moment computation,” *Computer-Aided Design*, vol. 43, pp. 1284–1295, Oct. 2011.

[23] L. Yang, F. Albregtsen, and T. Taxt, “Fast Computation of Three-Dimensional Geometric Moments Using a Discrete Divergence Theorem and a Generalization to Higher Dimensions,” *Graphical Models and Image Processing*, vol. 59, pp. 97–108, Mar. 1997.

[24] D. Xu and H. Li, “Geometric moment invariants,” *Pattern Recognition*, vol. 41, pp. 240–249, Jan. 2008.

[25] P. Cignoni, C. Rocchini, and R. Scopigno, “Metro: Measuring Error on Simplified Surfaces,” in *Computer Graphics Forum*, vol. 17, pp. 167–174, Jun. 1998.

[26] MathWorks, *Getting Started with MATLAB*. MathWorks.

[27] J. de Freitas, J. Cronemberger, R. M. Soares, and C. N. Amorim, “Modeling and assessing BIPV envelopes using parametric Rhinoceros plugins Grasshopper and Ladybug,” *Renewable Energy*, vol. 160, pp. 1468–1479, Nov. 2020.

Appendices

Appendix A

Implementation on MATLAB

sec:app

A.1 Reading the database table

```
T = readtable('./Database/AirfoilParameters.xlsx','Range','C:S');

T = table2array(T);

n_points = 200;

n = 4;

vect = zeros(length(T), 3*n*n_points + 1 + 2*n);

lenT = 1263;
```

A.2 Parametric modeling

```
% Source code of the parametric model

function [ crv, crv_u, crv_l ] = airfoil_pmodel( v, l, s)

if nargin==1

    Length = 1;

    s = false;

elseif nargin==2

    s = false;

    Length = l;

else

    Length = l;

end
```

```
u_max = v(1) ∗ (0.99 ∗ Length/5 ) + 0.01 ∗ Length/5 ;
x_u_max = v(2) ∗ 0.6 ∗ Length + 0.1 ∗ Length;
angle_b_u = v(3) ∗ 90 ∗ pi / 180;
tip_u = v(4) ∗ (0.9 - 0.1) + 0.1;
s_u = x_u_max + (Length - x_u_max) ∗ (v(5) ∗ 0.9 + 0.05);
if s
    l_max = u_max;
    x_l_max = x_u_max;
    angle_b_l = angle_b_u;
    s_l = s_u;
    tip_l = tip_u;
    angle_l = 0;
else
    l_max = v(6) ∗ (0.99 ∗ Length/5 ) + 0.01 ∗ Length/5 ;
    x_l_max = v(7) ∗ 0.6 ∗ Length + 0.1 ∗ Length;
    angle_b_l = angle_b_u ∗ (2 ∗ v(8) - 1);
    tip_l = v(9) ∗ (0.9 - 0.1) + 0.1;
    s_l = x_l_max + (Length - x_l_max) ∗ (v(10) ∗ 0.9 + 0.05);
    angle_l = (v(11) ∗ 40 - 20)∗pi/180;
end
%Upper Side
point_a = [0, 0, 1];
point_b1 = [0, tip_u ∗ u_max, 1];
point_b2 = [(1 - tip_u) ∗ x_u_max, u_max, 1];
point_c = [x_u_max, u_max, 1];
point_d1 = [s_u, u_max, 1];
y_t = (Length - s_u) ∗ tan(angle_b_u);
```

```
if y_t < u_max
  point_d2 = [s_u, y_t, 1];
else
point_d2 = [Length - u_max / tan(angle_b_u), u_max, 1];
end
point_e = [Length, 0, 1];
%rotate control points around leading edge
cosa = cos(angle_l);
sina = sin(angle_l);
point_b1 = point_b1*[cosa, sina, 0;-sina, cosa, 0;0, 0,1];
%create upper side curve
upperSide = rsmak([0,0, 0, 0, .5, .5, .5, 1, 1, 1, 1],
[point_a', point_b1', point_b2', point_c',
point_d1', point_d2', point_e']);

%Lower Side
point_b1 = [0, -tip_l * l_max, 1];
point_b2 = [(1 - tip_l) * x_l_max, -l_max, 1];
point_c = [x_l_max, -l_max, 1];
point_d1 = [s_l, -l_max, 1];
if s
    point_d2(2) = -point_d2(2);
else
if angle_b_l > 0
y_t = (Length - s_l) * tan(angle_b_l);
        if y_t < u_max
            point_d2 = [s_l, y_t, 1];
        else
```

```
            point_d2 = [Length - u_max / tan(angle_b_l),
            u_max, 1];
        end
    else
        angle_b_l = -1 * angle_b_l;
        y_t = (Length - s_l) * tan(angle_b_l);
        if y_t < l_max
            point_d2 = [s_l, -y_t, 1];
        else
            point_d2 = [Length - l_max / tan(angle_b_l), -l_max, 1];
        end
end
end
%rotate control points around leading edge
point_b1 = point_b1*[cosa, sina, 0;-sina, cosa, 0;0, 0,1];
%construct lower side curve
lowerSide = rsmak([0,0, 0, 0, .5, .5, .5, 1, 1, 1,1],
[point_a', point_b1', point_b2', point_c', point_d1',
point_d2', point_e']);
%Construct a curve from both sides
crv = rsmak([0,0, 0, 0, .25, .25, .25, .5, .5,
.5, .75, .75, .75, 1, 1, 1,1],
[fliplr(upperSide.coefs),lowerSide.coefs(:,2:end)]);
crv_l = lowerSide;
crv_u = upperSide;


end
```

```
% Fixed wing parameters
chord = 1;
lambda = deg2rad(5);
b = 10;

% Relations between wing and airfoil parameters
% At root
     wing(1,:) = foil;

     % At tip
     x_le = b/2*tan(lambda);
     x_te = x_le + chord;
     foil(1:n_points) = foil(1:n_points)*chord+x_le;
     wing(n,:) = foil;
     foil = wing(1,:);

     % At other sections
     for i = 1 : n-1
        x_le = i/(n-1) * b/2*tan(lambda);
        x_te = x_le + chord;
        foil(1:n_points) = foil(1:n_points)*chord+x_le;
        wing(i+1,:) = foil;
        foil = wing(1,:);
     end
```

A.3 Design discretization

```
% Function for discretizing airfoils with 4 methods of point
```

```
distribution along
% the curve
function [points] = discretize_foil(foil,N,mode)
foil.knots = (foil.knots - foil.knots(1))./(foil.knots(end)-
foil.knots(1));
switch mode
    case 0 %uniform spacing in parametric domain
        t = linspace(0,1,N);
    case 1 %cosine spacing
        t = linspace(0,1,N/2);
        t = (1-cos(pi*t))/4;
        t = [t,t+t(end)];
    case 2 %curvature-based spacing
        NN = 50;
        t1 = linspace(0,1,N*NN);
        vals = fntlr(foil,3,t1);
        c = vecnorm(cross([vals(3:4,:);zeros(1,N*NN)],
        [vals(5:6,:);zeros(1,N*NN)]))./(vecnorm(vals(3:4,:)).^3);
        cc = zeros(1,N*NN);
        for i=2:N*NN
            cc(i) = (c(i-1)+c(i))*norm(vals(1:2,i)-vals(1:2,i-
            1))/2 + cc(i-1);
        end
        [cc, indices] = unique(cc);
        t1 = t1(indices);
        step = cc(end)/(N-1);
        k = 0:N-1;
        step = k*step;
        t = interp1(cc,t1,step,"linear","extrap");
        max_interval = 2/(N-1);
        i=1;
        while i<length(t)
            if t(i+1)-t(i)>max_interval
                t = [t(1:i) (t(i+1)+t(i))/2 t(i+1:end)];
            else
                i = i+1;
            end
        end
        while length(t)>N
            vals = fntlr(foil,1,t);
            lengths = vecnorm(vals(1:2,2:end)-vals(1:2,1:end-1));
            [~,i] = min(lengths);
            if i==1
                i=2;
            end
            t = [t(1:i-1) (t(i-1)+t(i+1))/2 t(i+2:end)];
        end
    otherwise %uniform spacing in physical domain
        L = arc_length( foil, 0, 1);
        dL = L/(N-1);
        t = zeros(1,N);
        options = optimset('TolX',1e-7);
        for i=2:N
            a = t(i-1);
            t(i) = fminbnd(@obj_fun,a,a+.25,options);
        end
```

```
end
% points = fnval(foil,t);
points = fntlr(foil,1,t);

    function res = obj_fun(x)
        res = abs(arc_length(foil,a,x)-dL);
    end
end
```

A.4 Computation of geometric moments

```
% To calculate moment(s), the shape should be composed of all
triangulated mesh
% elements
p=2; q=0; r=0;
moment =
sthOrderGeometricMoment('C:\Users\Sultangazin\Desktop\DimRed\Data
base\vect.stl', p,q,r,isCentral,isScaled);
disp(['m_' num2str(p) num2str(q) num2str(r) ' = '
num2str(moment)])


% this function will evaluate a moment vector (M^s) containing
all s-order
s = 2;
moments = sthOrderGeometricMomentVector('vect.stl',
s,isCentral,isScaled);
```

A.5 Dimensionality reduction

```
% These two lines of code normalize the Design grid to have unit
variance and
% implement KLE
normSamples = samples./sqrt(sum(var(samples)));
[KL_modes, KL_values] = KLE(normSamples);

% The following code calculates retained variance as a
cumulative KL-values sum
varRetained = (cumsum(KL_values)/sum(KL_values))*100;
[~,n_comp] = max(varRetained >= 95);
disp(['Subspace dimensionality: ' num2str(n_comp)])
disp(['Reduction percentage: ' num2str((size(designSpace,1)-
n_comp)/size(designSpace,1)*100) '%'])
```

A.6 Design space quality analysis

```
% Checking Self Intersection
SelfInterDes = robustness(designs(1:2*n*n_points));
perOfSelfIntDes = size(SelfInterDes,1)/size(designs,1)*100;
disp(['% of Self Intersecting Designs: '
num2str(perOfSelfIntDes)])

% Diversity evaluation
```

```
Hdist = diversity(designs(1:3*n*n_points),params,2, n, n_points);

diversity = max(Hdist);

disp(['Diversity of Latent Space: ' num2str(diversity)])
```