Forschungsbericht 2024-22

A Predictive Design Method for Compressor Blisks

Nicolai Forsthofer

Deutsches Zentrum für Luft- und Raumfahrt
Institut für Bauweisen und Strukturtechnologie
Stuttgart

Deutsches Zentrum
DLR für Luft- und Raumfahrt

Forschungsbericht 2024-22

A Predictive Design Method for Compressor Blisks

Nicolai Forsthofer

Deutsches Zentrum für Luft- und Raumfahrt
Institut für Bauweisen und
Strukturtechnologie
Stuttgart

164 Seiten
59 Bilder
12 Tabellen
110 Literaturstellen

DLR Deutsches Zentrum für Luft- und Raumfahrt

Herausgeber:

Deutsches Zentrum
für Luft- und Raumfahrt e. V.
Wissenschaftliche Information
Linder Höhe
D-51147 Köln

ISSN 1434-8454
ISRN DLR-FB-2024-22
Erscheinungsjahr 2024
DOI: 10.57676/n1bj-4w35

D 93 (Dissertation der Universität Stuttgart)

Gasturbine, Triebwerk, Strukturmechanik, Verdichter, Rotorscheiben, Blisk, Maschinelles Lernen, Neuronale Netze, Optimierung, Synthetische Datenerzeugung, Automation, GTlab

Nicolai FORSTHOFER
DLR, Institut für Bauweisen und Strukturtechnologie, Stuttgart

Ein prädiktives Entwurfsverfahren für Verdichter-Blisks
Universität Stuttgart

Die vorliegende Arbeit beschreibt die Anwendung von ML Methoden auf ein klassisches Ingenieursproblem: Die strukturmechanische Dimensionierung von Verdichterkomponenten. Der Fokus liegt dabei in gleichen Teilen auf der Erzeugung synthetischer Daten mithilfe von vollautomatisierten Prozessen sowie dem anschließenden Training der ML Modelle. Die Entwicklung der Methode folgt dem folgenden Aufbau: Zunächst wurde eine geeignete geometrische Beschreibung der Verdichterscheibe erarbeitet. Um die anschließende strukturmechanische Berechnung zu ermöglichen, wurden an die Geometrie angepasste Vernetzungsprozesse, sowohl für das Verdichterblatt als auch für die Scheibe, entwickelt. Für die Ermittlung der Bauteilauslastung wurde die finite Element Methode angewandt. In einem Löser- und Analysevergleich zeigte der open-source Löser CalculiX mit einer linear statischen Analyse die beste Kombination aus benötigter Berechnungszeit und erreichter Genauigkeit. Mithilfe von speziell entwickelten Methoden können die benötigten Ziel- und Kontrollgrößen aus den Ergebnisdaten extrahiert werden. Der bis hier beschriebene Prozess bildet die Grundlage für eine Parameteroptimierung der Scheibengeometrie. Gesucht sind Scheiben mit der geringsten Masse die dennoch die notwendigen Restriktionen einhalten. Die Integration in das GTlab Framework ermöglicht die vollständige Automatisierung des Optimierprozesses. Dies ist zwingend notwendig um eine ausreichende Menge an synthetischen Datensätzen zu generieren die die optimalen Scheibenformen für verschiedene Eingabeparameter enthalten. Der erzeugte strukturierte und gereinigte Datensatz dient als Trainingsgrundlage für verschiedene Ersatzmodellarchitekturen. Bei einem Vergleich hinsichtlich Vorhersagegenauigkeit und Robustheit erwies sich das neuronale Netz unter Einbeziehung von Maßnahmen gegen Rauschen, wie Dropout-Schichten und angepasste Loss Funktionen, als die überlegene Lösung. Nach der Validierung wurde das Modell als Teil des strukturmechanischen Moduls mit dem GTlab Framework zur Verfügung gestellt. Die vorgestellte Methode ermöglicht die Auslegung von Verdichterscheiben nahezu in Echtzeit. Das trainierte Modell ist in der Lage, optimale Scheibendesigns basierend auf Eingabemerkmalen, hauptsächlich der Schaufelgeometrie und den herrschenden Lasten, vorherzusagen und fasst damit ein multidimensionales Optimierungsproblem in einem recheneffizienten Ersatzmodell zusammen. Eine Bewertung lässt sich anhand zweier state-of-the-art Methoden durchführen. Bei einem Vergleich mit einer schnellen Vorauslegungsmethode überzeugt die Neuentwicklung sowohl in der Gegenüberstellung der Berechnungszeit mit Millisekunden gegenüber mehreren Sekunden, als auch deutlich im Detailgrad. Die Vorauslegungsmethode eignet sich hier lediglich zur Gewichtsabschätzung. Im Vergleich mit etablierten Detailauslegungsmethoden wird ein ähnlicher Detailgrad bei gleichzeitig drastischer Reduktion der Berechnungszeit erreicht. Abschließend wird ein Ausblick gegebenen mit der Absicht die Methode zu generalisieren und zu verbessern um sie auf komplexere Fragestellungen anwenden zu können.

Gas turbine, Jet Engine, Structural Mechanics, Compressor, Rotor disks, Blisk, Machine Learning, Neural networks, Optimization, Synthetic Data Generation, Automation, GTlab

(Published in english)

Nicolai FORSTHOFER
German Aerospace Center (DLR), Institute of Structures and Design, Stuttgart

A Predictive Design Method for Compressor Blisks

University of Stuttgart

The present work describes the application of ML methods to a classical engineering problem: The structural mechanical sizing of compressor components. The thesis focuses on the generation of synthetic data via fully automated procesesses and the subsequen training of ML models in equal depth. The development of the method follows the structure outlined below: Initially, a suitable geometric description of the compressor disk was developed. Consequently, the disk can be described based on eight independent parameters while maintaining a very high degree of design freedom. To enable subsequent structural mechanical calculations, meshing processes tailored to the geometry were developed for both the compressor blade and the disk. The finite element method was applied to determine component stress and the displacement field. In a solver and analysis comparison, the open-source solver CalculiX, using a linear static analysis, showed the best combination of required computation time and achieved accuracy. Using specifically developed methods, the necessary target and control variables can be extracted from the result data. The process described so far forms the basis for parametric optimization of the disk geometry. The goal is to find disks with the lowest mass that still comply with the necessary constraints, such as maximum stresses and displacement ratios. Integration into the GTlab Framework enables the complete automation of the optimization process. This is essential in order to generate a sufficient quantity of synthetic data points containing the optimal disk shapes for various input parameters. The structured and scrubbed dataset serves as a training ground for various surrogate model architectures. In a comparison regarding prediction accuracy, required data volume, and robustness, the neural network, incorporating measures against noise such as dropout layers and tailored loss functions, proved to be the superior solution. Following validation, the model was made available as part of the structural mechanics module within the GTlab Framework. The introduced method enables the design of compressor disks almost in real-time. The trained model is capable of predicting optimal disk designs based on input features, primarily blade geometry and prevailing loads, effectively encapsulating the multidimensional optimization problem into a computationally efficient surrogate. An evaluation is conducted using two state-of-the-art methods. In comparison with a rapid preliminary design method, the new development succeeds both in terms of computational time, with milliseconds versus several seconds, and in the level of detail achieved. The preliminary design method is suitable merely for weight estimation. In comparison with established detailed design methods, a similar level of detail is achieved while simultaneously drastically reducing the needed computation time. The detailed methods require minutes to hours depending on the optimization procedure used. Finally, an outlook is given with the intention for the method to be generalized and improved in order to apply it to more complex problems.

A Predictive Design Method for Compressor Blisks

A thesis submitted to the Faculty of Aerospace Engineering and Geodesy of the University of Stuttgart in partial fulfilment of the requirements for the degree of
Doctor of Engineering Sciences (Dr.-Ing.)

Submitted by

Nicolai Forsthofer

born in Eichstätt

Main referee:	Prof. Dr.-Ing. Heinz Voggenreiter
Co-referee:	Prof. Dr. Homayoun Najjaran
Day of the Exam:	27.08.2024

Institute of Aircraft Design
University of Stuttgart

2024

Contents

Abstract

The present work describes the application of ML methods to a classical engineering problem: The structural mechanical sizing of compressor components. The thesis focuses on the generation of synthetic data via fully automated procesesses and the subsequen training of ML models in equal depth. The development of the method follows the structure outlined below: Initially, a suitable geometric description of the compressor disk was developed. Consequently, the disk can be described based on eight independent parameters while maintaining a very high degree of design freedom. To enable subsequent structural mechanical calculations, meshing processes tailored to the geometry were developed for both the compressor blade and the disk. The finite element method was applied to determine component stress and the displacement field. In a solver and analysis comparison, the open-source solver CalculiX, using a linear static analysis, showed the best combination of required computation time and achieved accuracy. Using specifically developed methods, the necessary target and control variables can be extracted from the result data. The process described so far forms the basis for parametric optimization of the disk geometry. The goal is to find disks with the lowest mass that still comply with the necessary constraints, such as maximum stresses and displacement ratios. Integration into the GTlab Framework enables the complete automation of the optimization process. This is essential in order to generate a sufficient quantity of synthetic data points containing the optimal disk shapes for various input parameters. The structured and scrubbed dataset serves as a training ground for various surrogate model architectures. In a comparison regarding prediction accuracy, required data volume, and robustness, the neural network, incorporating measures against noise such as dropout layers and tailored loss functions, proved to be the superior solution. Following validation, the model was made available as part of the structural mechanics module within the GTlab Framework. The introduced method enables the design of compressor disks almost in real-time. The trained model is capable of predicting optimal disk designs based on input features, primarily blade geometry and prevailing loads, effectively encapsulating the multidimensional optimization problem into a computationally efficient surrogate. An evaluation is conducted using two state-of-the-art methods. In comparison with a rapid preliminary design method, the new development succeeds both in terms of computational time, with milliseconds versus several seconds, and in the level of detail achieved. The preliminary design method is suitable merely for weight estimation. In comparison with established detailed design methods, a similar level of detail is achieved while simultaneously drastically reducing the needed computation time. The detailed methods require minutes to hours depending on the optimization procedure used. Finally, an outlook is given with the intention for the method to be generalized and improved in order to apply it to more complex problems.

Kurzfassung

Die vorliegende Arbeit beschreibt die Anwendung von ML Methoden auf ein klassisches Ingenieursproblem: Die strukturmechanische Dimensionierung von Verdichterkomponenten. Der Fokus liegt dabei in gleichen Teilen auf der Erzeugung synthetischer Daten mithilfe von vollautomatisierten Prozessen sowie dem anschließenden Training der ML Modelle. Die Entwicklung der Methode folgt dem folgenden Aufbau: Zunächst wurde eine geeignete geometrische Beschreibung der Verdichterscheibe erarbeitet. Um die anschließende strukturmechanische Berechnung zu ermöglichen, wurden an die Geometrie angepasste Vernetzungsprozesse, sowohl für das Verdichterblatt als auch für die Scheibe, entwickelt. Für die Ermittlung der Bauteilauslastung wurde die finite Element Methode angewandt. In einem Löser- und Analysevergleich zeigte der open-source Löser CalculiX mit einer linear statischen Analyse die beste Kombination aus benötigter Berechnungszeit und erreichter Genauigkeit. Mithilfe von speziell entwickelten Methoden können die benötigten Ziel- und Kontrollgrößen aus den Ergebnisdaten extrahiert werden. Der bis hier beschriebene Prozess bildet die Grundlage für eine Parameteroptimierung der Scheibengeometrie. Gesucht sind Scheiben mit der geringsten Masse die dennoch die notwendigen Restriktionen einhalten. Die Integration in das GTlab Framework ermöglicht die vollständige Automatisierung des Optimierprozesses. Dies ist zwingend notwendig um eine ausreichende Menge an synthetischen Datensätzen zu generieren die die optimalen Scheibenformen für verschiedene Eingabeparameter enthalten. Der erzeugte strukturierte und gereinigte Datensatz dient als Trainingsgrundlage für verschiedene Ersatzmodellarchitekturen. Bei einem Vergleich hinsichtlich Vorhersagegenauigkeit und Robustheit erwies sich das neuronale Netz unter Einbeziehung von Maßnahmen gegen Rauschen, wie Dropout-Schichten und angepasste Loss Funktionen, als die überlegene Lösung. Nach der Validierung wurde das Modell als Teil des strukturmechanischen Moduls mit dem GTlab Framework zur Verfügung gestellt. Die vorgestellte Methode ermöglicht die Auslegung von Verdichterscheiben nahezu in Echtzeit. Das trainierte Modell ist in der Lage, optimale Scheibendesigns basierend auf Eingabemerkmalen, hauptsächlich der Schaufelgeometrie und den herrschenden Lasten, vorherzusagen und fasst damit ein multidimensionales Optimierungsproblem in einem rechen-effizienten Ersatzmodell zusammen. Eine Bewertung lässt sich anhand zweier state-of-the-art Methoden durchführen. Bei einem Vergleich mit einer schnellen Vorauslegungsmethode überzeugt die Neuentwicklung sowohl in der Gegenüberstellung der Berechnungszeit mit Millisekunden gegenüber mehreren Sekunden, als auch deutlich im Detailgrad. Die Vorauslegungsmethode eignet sich hier lediglich zur Gewichtsabschätzung. Im Vergleich mit etablierten Detailauslegungsmethoden wird ein ähnlicher Detailgrad bei gleichzeitig drastischer Reduktion der Berechnungszeit erreicht. Abschließend wird ein Ausblick gegebenen mit der Absicht die Methode zu generalisieren und zu verbessern um sie auf komplexere Fragestellungen anwenden zu können.

1 Introduction

Aerospace research and development is in a constant struggle to increase efficiency and as a consequence build lighter airplanes which need less fuel. While the technology matures, improvements are harder to gain with classic methods, yet the matter of climate change urges for greener jet engines and new technologies. The rise of artificial intelligence could be the disruptive technology needed to speed up development and reach the ambitious goals to mitigate the negative effects of aviation on our planet.

1.1 Background and Motivation

The impact of aviation on climate change has typically been estimated at 1.8-2.5% [1], but a recent study by Lee et al. [2] puts the contribution of global aviation to anthropogenic global warming at 3.5%, while taking into account both the direct influence of emitted carbon dioxide (CO2) and non-CO2 effects such as nitrogen oxides (NOx), water vapor and other particulate emissions, which have been difficult to estimate to date. This approach allows a quantitative comparison of the impact of different emitters. The still seemingly small share can lead to a neglect of aviation's impact

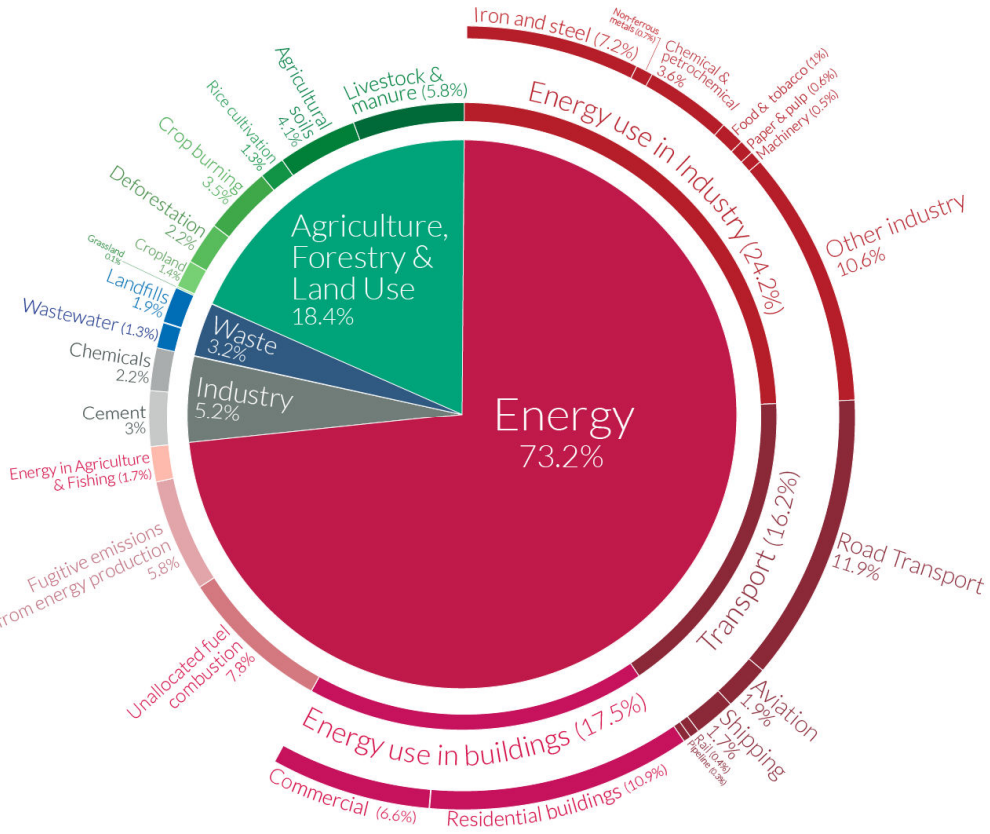
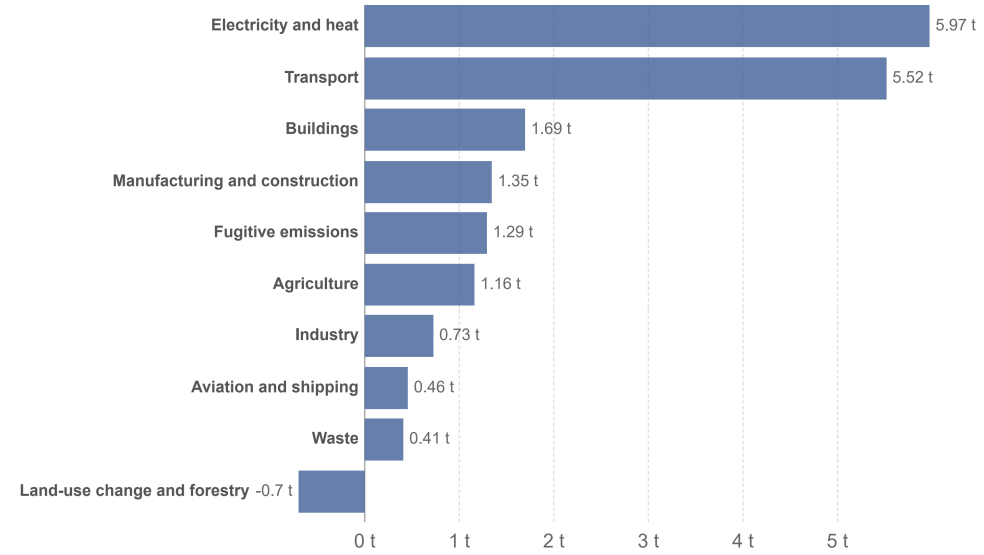


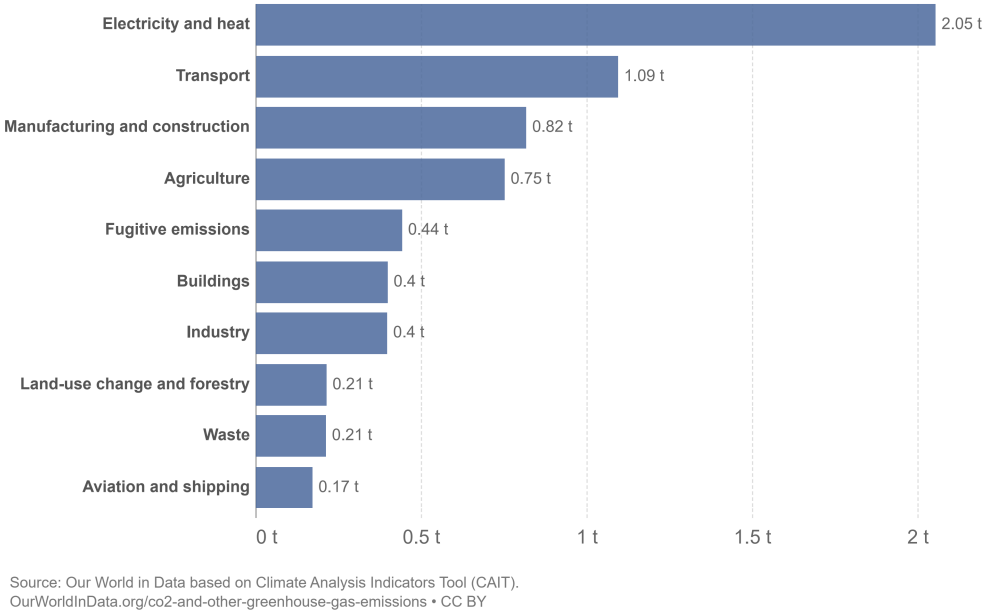
Figure 1.1: CO2 emissions by sector [3]

compared to other climate damaging actors such as electricity or heat generation. If the total global greenhouse gas emissions are considered, the transport sector accounts for about 16%, according to Ritchie et al. [3] (Figure 1.1). Both the share of transportation in total emissions and particularly the share of aviation in the transportation sector of 11.1% have remained almost unchanged since the end of the 1990s [1], while the absolute amount of CO2 emitted by aviation alone has more than doubled worldwide in the same period, from about 500 million tons to more than a billion [4]. The constant share of these sectors therefore simply means that total

emissions have continued to rise and will probably continue to do so unless global climate policy changes. The impact of aviation is even more relevant when the global view is replaced by a distinction between developed, developing and emerging countries. A look at the distribution of the emissions in the United States already shows a significantly higher share for the transport sector, at around 27% [1]. Even if this value cannot be directly compared with figure 1.1 due to possible differences in clustering, the global inequality of distribution becomes clear when converting to per capita emissions. As shown in figure 1.2, average per capita emissions in the United States in 2019 were more than five times the global average. Yet, the growth in air travel has not yet peaked, even in the developed world. Combined with rising demand in developing and emerging markets, a study by the German Aerospace Center (DLR) [5] predicts that annual air traffic will grow by an average of 3.7% over the next 20 years, with most of the growth expected in Asian countries. In absolute terms, this means an increase in the number of passengers from four billion in 2016 to over 9.4 billion in 2040. While North America and Europe follow very similar trajectories, Asia will account for most of the growth in the short term and Africa and the Middle East in the medium term [2]. Increasing air traffic at the current rate while maintaining emissions is incompatible with climate goals of 1.5 or even 2.0 degrees [6]. In consequence the aviation industry is far from irrelevant to mitigating the effects of climate change. In order to significantly improve the environmental performance of aviation in the future, there is an increasing focus on future technologies such as full electric flight or hydrogen as an



(a) United States



(b) World

Figure 1.2: Per capita greenhouse gas emissions by sector 2019 [3]

energy source. These technologies are currently at a low technical readiness level [7] and will only find application in commercial aviation in the long term. The use of hydrogen alone could reduce the impact of aviation by 50 to 75% [8]. But even ambitious targets put 2035 as the earliest date for the introduction of a hydrogen-powered medium-haul aircraft [8]. Alternative fuels, so-called biofuels, can be used in the medium term because they can directly replace or supplement the kerosene used in conventional engines [9]. The reduction in climate-damaging emissions that can be achieved in this way is subject to considerable uncertainty, as many fuel blends and additives are currently being researched. Studies suggest a reduction of 4 to 22.5% in 2050 compared to 2022 emissions corrected for COVID-19 effects [10]. While this reduction is promising, it is not sufficient on its own. Therefore, the most effective way to reduce CO2 emissions in the short term is to further improve the current technology of the already very efficient turbofan engine, while also accelerating the development of the mentioned new technologies to market maturity. To this end, there are numerous research programs, both national and international, such as the Clean Sky program between the European Commission and the European aviation industry. Since the inception of commercial aviation, continuous improvements in aircraft and engine design have resulted in significant reductions in fuel consumption and climate-damaging emissions. According to a 2005 study [11], the increase in efficiency in air transportation, measured in terms of fuel consumption per passenger kilometer, was about 53% between 1965 and 2004. This large increase in efficiency can be attributed to sev-

eral technological advances. The study also concludes that progress in fuel efficiency has been largely driven by engine development, as only minimal improvements were made to the overall aircraft design during the period 1955-1990. The aircraft and the engine cannot be considered in isolation, especially in view of the computational accuracy required today, as there are for example interactions due to the position of the engine mount. Technical developments in engine research have had a particularly significant impact in the following areas:

- Materials and alloys
- Maintenance and repair
- Major Engine concept changes

Especially in the first decade after the introduction of the turbofan engine, the rate of efficiency improvement has been steep due to the many opportunities for improvement offered by this still young technology. As with most technologies, the rate of improvement has gradually declined and, while still steady in recent years, is now in the single digits. This correlation can be seen in Figure 1.3, where the data points have been taken from an IPCC study [6] and checked and corrected in a meta-study from the Netherlands Aerospace Center (NLR) [11]. Major innovations, such as the use of fiber-reinforced ceramics, are still possible today, particularly in areas of high operating temperature in the combustor and turbine. Less obvious is the impact of improved maintenance and repair methods, especially the prediction of maintenance intervals. Yet these ensure the smooth

operation of the engines, enabling very high levels of efficiency to be maintained over a product life of up to 30 years. Progress in the overall configuration of the engines, has been more erratic and can be attributed to several specific technological breakthroughs. The increase in the number of shafts from two to three, patented by Rolls-Royce PLC in 2009, leads to fuel savings of up to 11%. The PW1000G geared turbofan engine, which entered service in 2016, again leads to savings of 16%, according to Pratt and Whitney [12], which includes the gains of the three shaft technology and the gearbox. Both inventions are aimed at increasing the bypass ratio, which has been shown to have a major impact on engine efficiency [13] and are responsible for the steeper decline in figure 1.3 through

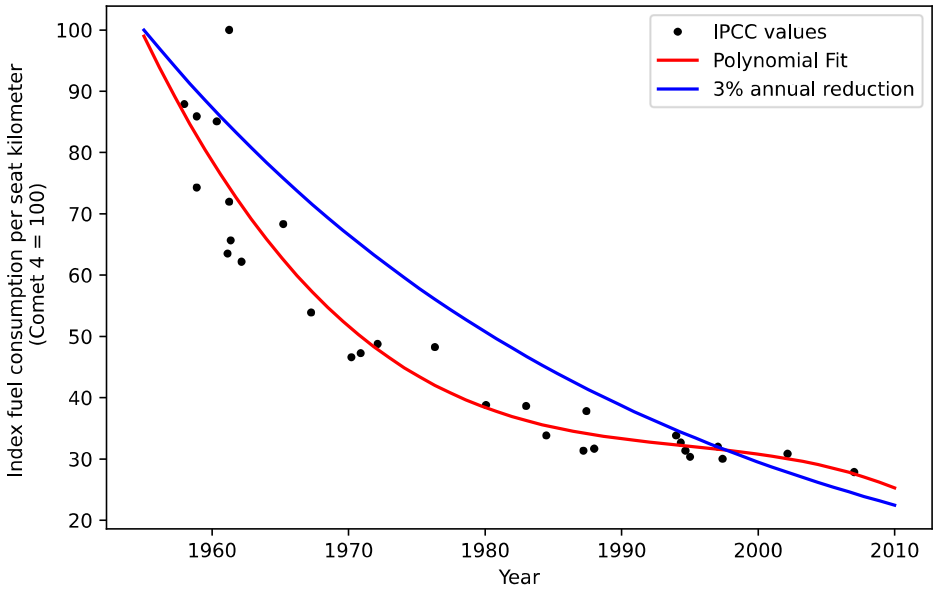


Figure 1.3: Fuel consumption reduction over time (Data from [11])

the last data point. In addition, new component technologies such as variable stator and rotor blades and consistent lightweight design throughout the engine are responsible for high efficiency gains. Even with an optimistic extrapolation, the trend does not greatly exceed the 3% annual reduction average in the future. With the predicted annual flight passenger increase of also 3% per year stated earlier [5], the advancements of regular jet engine technology will merely hold the status quo in terms of overall emissions and that only in the best case scenario. In all cases these efficiency gains have one consequence in common: an increase in engine complexity for all processes involved. This complexity could be controlled by improving the simulation capabilities.

The aerospace industry has always been at the forefront of simulation technology users and developers, due to the complex physics involved. The development of simulation technology in the last century has therefore been mainly focused on physical phenomena and their better understanding in combination with increasingly accurate theoretical models. Especially the field of engineering, computational fluid dynamics (CFD) in aerodynamics and the finite element method (FEM) in structural mechanics have made great improvements in terms of the accuracy of the representation of real-world phenomena and the speed of the computation. These advances were mainly driven by the introduction of computer technology into nearly every engineering field starting in the late 1970s. While the improvement of simulation methods offered a satisfactory improvement of the prediction accuracy, the focus shifted in the later years of the

century. With the increase in computational power following an exponential growth commonly known as Moore's Law (Figure 1.4), more and more data-intensive methods became possible. Especially data science is a rapidly growing field in the recent years with the potential to revolutionize many industries, including traditional engineering. In the near future, data science could play an increasingly

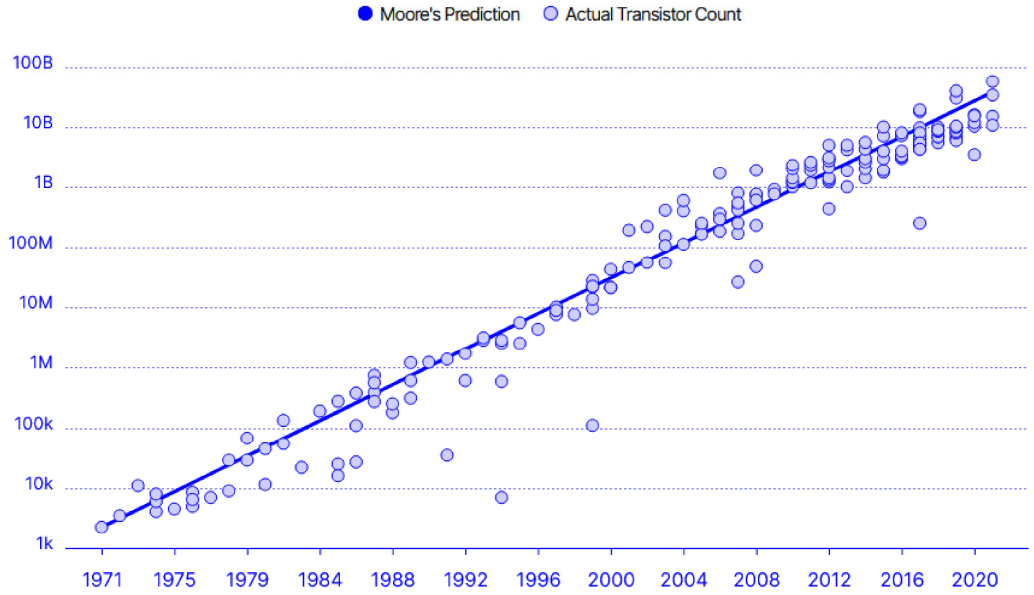


Figure 1.4: Moore's law [14]

important role in the design, development, and optimization of mechanical parts and components with the key technology of machine learning. These algorithms are capable of analyzing large amounts of data and identifying patterns and relationships that would be difficult or impossible for humans to detect. This allows engineers to optimize traditional designs and identify the factors that drive design decisions. A special feature of traditional engineering is the

widespread use of simulation tools and the associated high degree of digitalization and automation. Only when processes from design, manufacturing and maintenance are digitally supported, can data science unleash its full potential. The early adoption of computational methods and the efforts to digitalize and automate design processes enables the application of the new methods to nearly every field of the aerospace industry and makes machine learning a candidate to be the key technology to address the challenges described at the beginning of this chapter and lead aviation into a green future in a much shorter time.

1.2 Objective and Scope

The application of machine learning methods for mechanical design purposes will be demonstrated using a structural mechanics process. In order to generate as large a database as possible, the prototype process must be fully automated and repeatable, and the design task must not be trivial, so that conclusions can be drawn for more complex applications in the future. In the Component Design and Manufacturing Technology department, in-depth knowledge of the structural-mechanical design and manufacturing of engine components has been acquired through numerous projects. The results of these calculations have been verified, validated and calibrated on DLR's own test rigs. The blade design for rotors was last demonstrated in 2022 during the CRISP (Counter Rotating Integrated Shrouded Propfan) test bed trials at the M2VP in Cologne. The

methods for designing the rotor disks of a low-pressure compressor were further developed and validated in cooperation with MTU Aero Engines in a collaborative project from 2014 to 2019. Therefore, the demonstration of a predictive design method on compressor blisks proved to be the most promising application. Turbomachinery blisks are used in all axial flow gas turbines. It is a safety critical component as blade loss and disk rupture must be prevented at all costs. Blisks and their associated shafts account for a large portion of the absolute weight of the engine, so optimizing these components has long been a high priority for engine manufacturers. Blade and disk are designed using multidisciplinary optimization processes, where the blade shape is aerodynamically dominated and the disk design driven by structural mechanic criteria. The blade geometries required for the disk design are provided by pre-design processes of the Institute of Propulsion Technology within the GTlab framework.

The goal is to develop a parametric geometry model of the blade-disk pairing that allows robust optimization with high degrees of freedom. The resulting geometries must be meshed in an automated and fault-tolerant manner. After setting the boundary conditions, a static FE analysis is performed. The previous steps are repeated until an optimal solution of the disk geometry with respect to its weight is found. The input parameters such as blade properties, material, loads and boundary conditions are stored in a database. This database is then used for the training of the surrogate model with selected machine learning methods. Finally, the model and the generated results are validated. Through this process the surrogate

model is able to predict the shape of optimal blisks based on input features, mainly the blade geometry, loading and design space, effectively encapsulating the multidimensional optimization problem in a computationally efficient surrogate. The selection of structural mechanical requirements for the disk design is based on the most current and comprehensive understanding available to the author. Nonetheless, the design of the process is designed to accommodate the incorporation of new or modified structural requirements for the design of compressor blisks as they arise.

The following chapter will give an overview over the state of the art design and manufacturing technologies used in modern aeroengine engineering.

2 Literature Review

In order to develop a method for the efficient and robust design of compressor blisks, a basic understanding of the component must first be acquired. This starts with the question of what makes the new design method different from existing methods and what new challenges arise. Additionally, the present available manufacturing methods have an impact on the design constraints and subsequent behavior of the component.

2.1 Bladed Disks

A blisk is a rotor with an integral connection between the blade and the disk, i.e. a one-piece design. Prior to the introduction of the blisk design, all rotors in axial turbomachinery were multi-piece. In the downstream stages, the multi-piece design is still used because of the high temperatures and the resulting demands on material properties. For this purpose, the blades are usually mounted on the disk by means of form-fit connections. There are numerous designs of blade-disk connections, but the most commonly used design is the firtree. A comparison of the integral and multipart designs can be seen in figure 2.1. The multi-piece design simplifies repair, as dam-

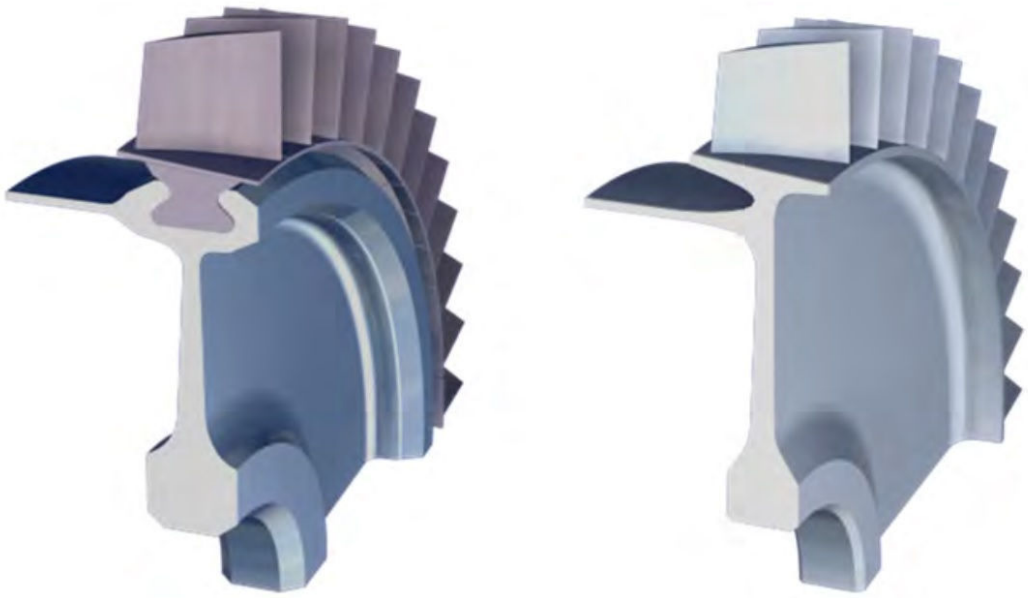


Figure 2.1: Blade and disk vs blisk design [15]

aged blades can be easily replaced, and the form fitting connection also decouples the vibration behavior of the disk and blade. However, the major disadvantage compared to the blisk design is the increased weight. The continuous development and optimization of high-pressure axial compressors is accompanied by an increasing load on the rotor blade rows. Higher compressor pressures lead to higher stage pressure ratios. The increasing cost pressure is met by reducing the number of components and stages, which means that blades with longer chord lengths have to be used due to the high flow velocities at higher rotor speeds. These blades place a higher load on the blade-disc connection than older blades due to the higher centrifugal forces. The integral design has a clear advantage here because the stresses are much lower due to the lack of notches. The first application of this design was demonstrated as

early as the 1980s in the General Electric T700 military helicopter engine [16], in which the axial stages were of integral design. One of the first serial productions was marked by the presentation of the EJ200 engine for the Eurojet in 1995. Even in the first ver-

Figure 2.2: EJ200 low pressure compressor blisks [15]

sion of the engine, the entire low-pressure compressor was of blisk design. This invention, driven by MTU Aero Engines, was necessary for the engine's required service life because the high speeds and associated centrifugal forces would have led to fretting corrosion in conventional designs. Fretting corrosion occurs on the small edges and imperfections of the blade roots and disk recesses and can eventually lead to cracking and blade loss. This problem is not

limited to the low-pressure compressor, which is why MTU used the blisk design for the first time in the high-pressure compressor on the second version of the EJ200 engine. Whereas the service life had previously been limited to 400 flight hours, it was now possible to achieve the required 4000 hours [17]. Another positive aspect is the reduction of leakage through gaps between disk and blade, which has a direct impact on aerodynamic efficiency [18]. These performance improvements ultimately led to the breakthrough of the blisk design. Originally limited to military engines, the blisk design is now used in virtually all types of engines. One of today's most advanced and efficient engines, the Trent XWB also uses blisks up to the first three stages of the high-pressure compressor. Rolls-Royce Ltd. reports a weight saving of 15 percent compared to a multipart design [19]. The engine cross-section shows that almost all stages of the low- and high-pressure compressors are designed as blisks. The resounding success of the design is also reflected in the production figures, as Rolls-Royce announced the delivery of the 10,000th blisk from the Oberursel plant in Germany in 2019 [19]. The key advantages and disadvantages of the blisk design can be summarized as shown in table 2.1. The specifics of blisk manufacturing, maintenance and

Advantages	**Disadvantages**
Weight reduction	Complex manufacturing
Assembly cost savings	Complex maintenance
Reduction of leakage flow	High costs
Reduced installation space	

Table 2.1: Advantages and Disadvantages of the Blisk Technology

design are covered in the following sections.

2.1.1 Manufacturing and Maintenance

Most blisks used in axial compressors are made of titanium alloys, mainly TI6242 and TI64. Only in the rear stages of the high-pressure compressor, the temperatures are too high that nickel alloys such as Inconel have to be used. Since the 1950s, the amount of titanium alloys used in aircraft engines has increased from three to 33 percent [20] because of its high strength to density ratio and temperature resistance up to 540 °C. The process developed in this paper is demonstrated on the low-pressure compressor, so only the manufacturing and maintenance processes for titanium alloy blisks are described below.

Milling

The most commonly used process is high-speed milling from solid plates. Blade geometries and disk contours are machined from cold-forged blanks on CNC milling machines. For simple geometries, turning processes may be sufficient for the disk contours. Due to high-cost pressure, high-speed milling has become established, which stands out from the usual milling processes with cutting speeds of 100 m/s in fine machining and up to 350 m/s in rough machining [21]. Very high accuracies are achieved for titanium alloy blisks with sufficiently large blade dimensions. The high volumes required, as described in the previous section, have led to a high degree of automation in blisk production. Milling makes the process highly

repeatable and standardized. In 2013, MTU Aero Engines opened one of the world's most modern factories for the series production of compressor blisks for the Pratt Whitney PW1000G engine. The

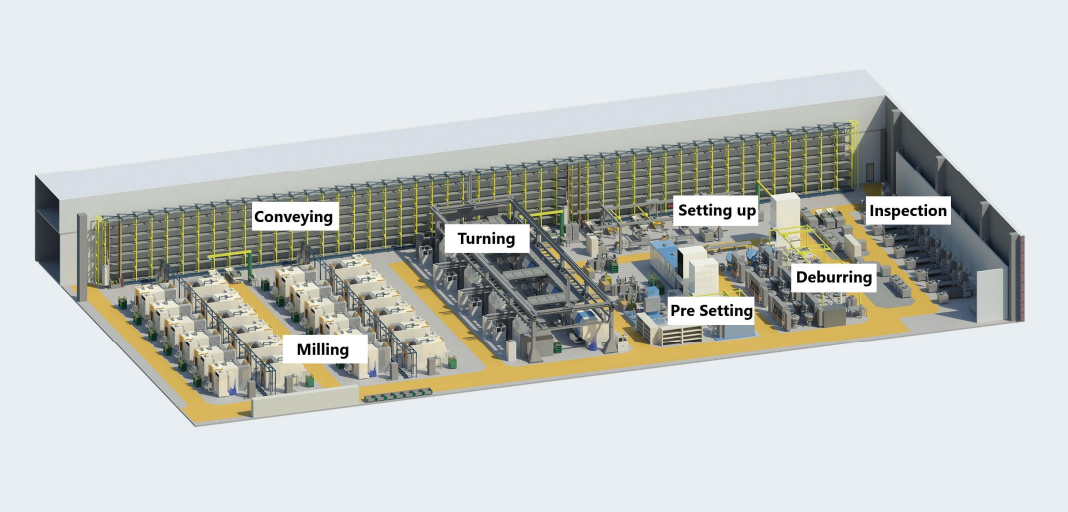


Figure 2.3: MTU Aero Engines: Modern blisk manufacturing facility [22]

10000 square meter facility allows for the automated production of up to 3500 blisks per year. The use of 24 Micron HPM 800U milling machines allows all blisks required for the GTF program to be produced in a single integrated manufacturing cell, rather than having to use different machines for each machining step [22].

Electrochemical Processing

Particularly in military applications, blisks are used with very small blade dimensions that make milling difficult or even impossible. With the use of blisk technology in high pressure compressors, this problem has also found its way into commercial engine manufacturing. For titanium or nickel alloy blisks with small diameters and some-

times highly complex blade geometries, the process of electrochemical machining has been developed. In this process, a cathodic tool is lowered into the anodically poled workpiece and the desired areas are removed by means of a rapidly flowing electrolytic solution. A major advantage is the very good reproducibility of the process, since there is almost no tool wear. For this reason, electrochemical processing is becoming increasingly commercially attractive for civil applications and have been in use in Pratt and Whitney's PW1100G-JM since 2015 [23]. The electrochemical manufacturing process used by MTU for the fifth and sixth HPC blisk has been tested and approved by Pratt and Whitney for the A320neo engine.

Linear Friction Welding

Linear friction welding, originally developed by MTU Aero Engines for repair purposes, is now increasingly being used as a manufacturing process. Under high pressure, the blade is welded to the disk in a linear oscillating motion. The process has already been applied to the EJ200 engine described in the previous section. A large amount of test and application data is available for components manufactured using the linear friction welding process. The process is divided into the following steps [15]:

1. The components to be joined are brought into contact under pressure. Frictional heat is generated and the actual contact area increases.

2. During the transition phase, wear is released until the contact

area equals 100 percent of the cross-section. The materials reach a plastic state.

3. In the equilibrium phase, heat is removed from the cutting surface while a plastic zone is created. Movement causes further wear, which shortens the length of the components.

4. In the deceleration phase, the amplitude is slowly reduced until the movement stops. The pressure is maintained.

Tensile tests on friction-welded specimens show sufficient strength compared to unwelded specimen. A comparison of the strength and elongation of two unwelded titanium alloys and their friction-welded joints in table 2.2 shows a negligible reduction in strength of only 1.34%. The high elongation value compared to the β alloy can be explained by the fact that the elongation of a composite is always determined by the part of the composite with the lower stiffness and is therefore close to the value of the α+β alloy. This process is more

Alloy	σ_{ys} **[MPa]**	σ_{uts} **[MPa]**	ϵ **[%]**
β Ti-6246	1051	1155	13
α+β Ti-6246	1044	1125	18
Welds	1030	1078	19

Table 2.2: Friction welding strength and elongation

complex to automate than milling, which is why milling is preferred for high volume applications. However, for repair processes involving large damage to one or more rotor blades, the process currently has a unique selling point.

2.1.2 Design Methods

Disk design is a long ongoing effort in engineering. The following sections detail different approaches to solve the dimensioning problem.

Analytical Dimensioning

For axisymmetric disks of arbitrary profile, a solution for the stress calculation was already proposed from Grammel [24] in 1936. With the known angular velocity ω, the following relationship exists between the disk thickness y_r as a function of the radial distance from the axis of rotation r and the stress components σ_r and σ_ϕ

$$\frac{d}{dr}(r\sigma_r y) - \sigma_\phi + \rho\omega^2 r^2 y = 0 \tag{2.1}$$

Assuming a homogeneous isotropic material, the following relationship can further be derived from the elasticity equations with the transverse contraction number ν.

$$r(\frac{d}{dr}\sigma_r - \nu\frac{d}{dr}\sigma_\phi) + (\nu + 1)(\sigma_r - \sigma_\phi) = 0 \tag{2.2}$$

For a given profile shape y_r, the simultaneous solution of the equations (2.1) and (2.2) would now solve the stress problem. Since in the case of the blisk design, the profile shape is the desired quantity, the problem must be reversed. The profile shape, assuming a known stress distribution, can generally be given as a linear differ-

r_a [m]	r_i [m]	σ_a $[\frac{kg}{cm^2 s^2}]$	ρ $[\frac{kg}{cm^3}]$
1.2	0.05	700	$4.4e^{-6}$

Table 2.3: Parameters for analytical disk design demonstration

ential equation of first order

$$y = a \exp(-\int \frac{r\frac{d}{dr}\sigma_r - \sigma_r - \sigma_\phi + \rho\omega^2 r^2}{r\sigma_r} dr) \tag{2.3}$$

where a is the integration constant. For compressor disks, some simplifications can now be made. A hyperbolic approach is sought which has a finite non-zero thickness at the inner edge $r = r_0$ and radial stress $\sigma_r = 0$ on the inside and $\sigma_r = \sigma_a$ on the outside. The differential equation (2.3) thus becomes

$$y = a(r + kr_0)^{(k^2-k+1)\rho\omega^2 c r_0^2)} \exp(-\rho\omega^2 c r[\frac{r}{2} - (k-1)r_0]) \tag{2.4}$$

with the dimensioning constant c

$$c = \frac{(r_a - r_0)(r_a + kr_0)}{\sigma_a r_a^2} \tag{2.5}$$

According to [25], k can be approximated as 1 for gas turbines, which makes formula (2.4) significantly simpler to

$$y = a(r + r_0)^{-\rho\omega^2 c r_0^2} e^{-0.5\rho\omega^2 c r^2} \tag{2.6}$$

To check the accuracy of the formula, a disk design is performed using the data in table 2.3 as an example. For better illustration, different speeds were used, and the results are shown in Figure 2.4.

With modern computing power, the evaluation of several hundred

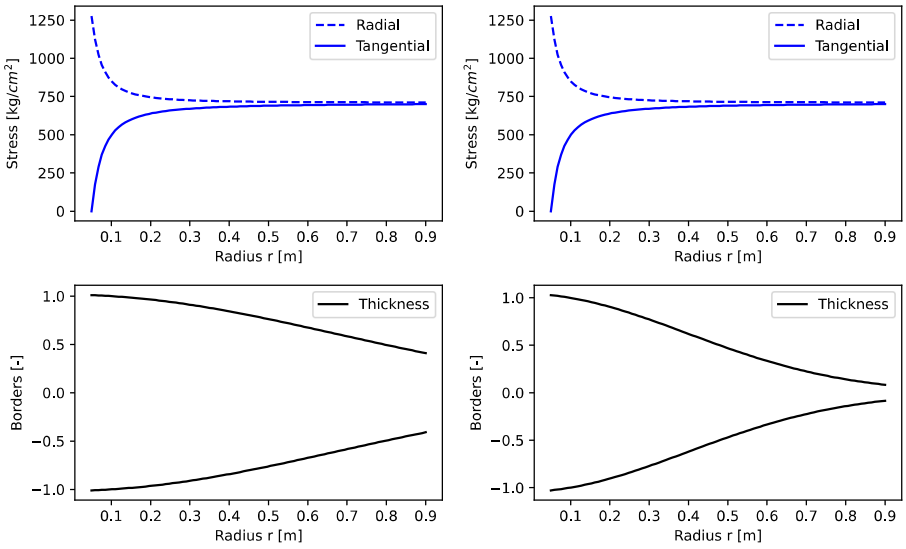


Figure 2.4: Results of the analytic disk dimensioning

different disk profiles takes only seconds, so the described approach has found application in numerous disk optimization methods. Yet computer support made it also possible to solve the underlying differential equation iteratively.

Finite Difference Optimization

The finite difference method (FDM) is the numerical method for solving partial and ordinary differential equations by approximation with a finite number of difference quotients. The method can be regarded as a precursor of FEM and is described in detail by Smith [26]. For the disk design application, the consideration of the cylindrical r, ϕ coordinate plane is still sufficient. The differential element

used for the discretization can then be defined according to figure 2.5. A particularly robust application of this method to the dimen-

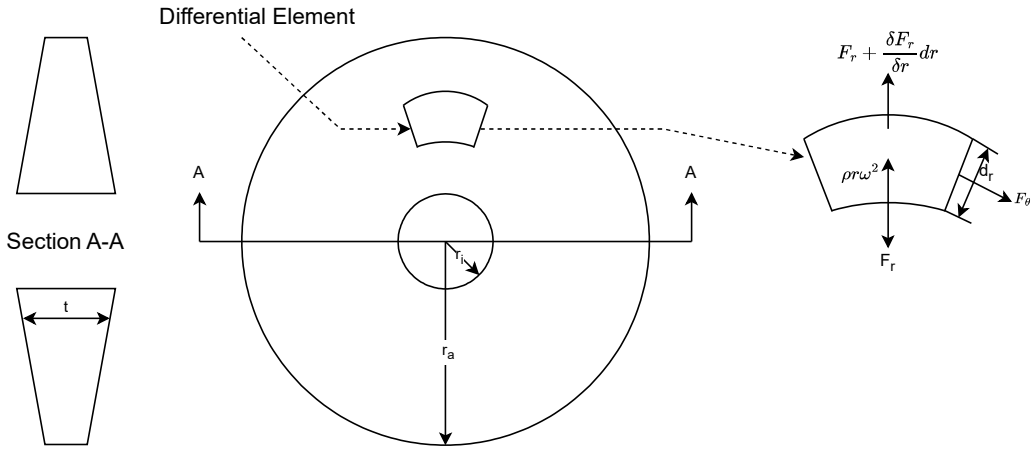


Figure 2.5: Finite differential element for disk dimensioning [27]

sioning of disks for axial impellers was developed at NASA in 1995 [27]. In contrast to earlier similar approaches by Manson [28] or Millenson [29], here, among other things, thermal stresses resulting from large temperature differences were integrated into the method, which is particularly necessary for disks for the high-pressure compressor. Analogous to the analytical solution, the formulation of the tangential and radial stresses according to the elasticity law is given by

$$\sigma_r = \frac{E}{1-\nu^2}[\epsilon_r + \nu\epsilon_\phi - (1+\nu)\alpha T] \tag{2.7}$$

$$\sigma_\phi = \frac{E}{1-\nu^2}[\epsilon_p hi + \nu\epsilon_r - (1+\nu)\alpha T] \tag{2.8}$$

But in contrast to the previous section, the thermal stresses are considered by the last term, with the coefficient of thermal expansion α and the temperature difference T. The required slice thickness is

calculated as a linear function of the radius with

$$t = mr + n \tag{2.9}$$

where n is a dimensional constant. In the cylindrical coordinate system, strains are defined as

$$\epsilon_\phi = \frac{u}{r} \tag{2.10}$$

$$\epsilon_r = \frac{du}{dr}, \tag{2.11}$$

where the desired profile thickness can now be formulated purely as a function of the radial displacement. Using the function (2.1) already known from the previous section and the temperature dependent formulations for the stresses (2.7) and (2.8), the differential equation to be solved can be expressed in the form

$$\frac{d^2u}{dr^2} + \left(\frac{2mr+n}{mr^2+nr}\right)\frac{du}{dr} + \left[\frac{\nu}{r^2} - \frac{1}{r^2} + \frac{\nu}{r}\left(\frac{2mr+n}{mr^2+nr}\right)\right]u = \alpha(1+\nu)\left[\frac{dT}{dr} + \left(\frac{2mr+n}{mr^2+nr} - 1\right)T\right] - \frac{\rho\omega^2(1-\nu^2)}{E}r \tag{2.12}$$

The equation (2.12) is a linear inhomogeneous differential equation of second order, which requires two boundary conditions for the solution. Here, analogous to the analytical solution, $\sigma_r = 0$ can be assumed at r=0, and the radial stress at the outer radius of the disk can be calculated due to the centrifugal force on the cross-section caused by the rotor blades. The difference quotient $\frac{du}{dr}$ to be solved can now be calculated at finitely many points for a quan-

titative solution. In addition, the method proposed by Armand [27]

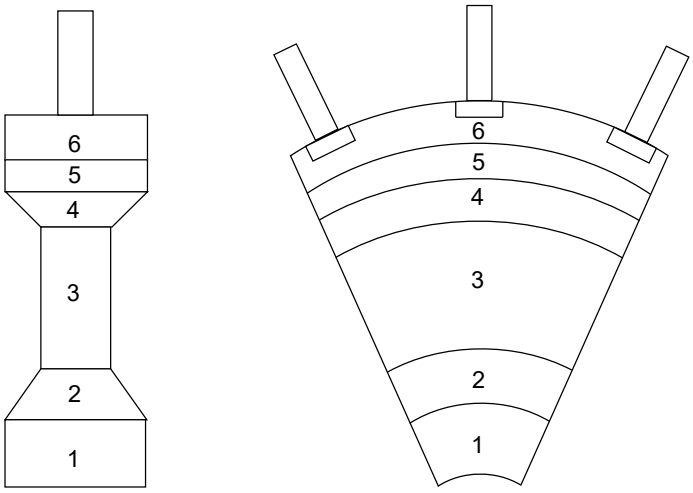


Figure 2.6: Disk sections of the finite difference method [27]

dimensions not only the disk contour, but also a parallel section at the disk interior, which is necessary for clamping during manufacturing, as well as the platform at the outer diameter for connection to the blade. For this purpose, the disc is divided into six sections in the radial direction, as shown in Figure 2.6. As can be seen, the method was not developed specifically for the design of blisks, but for disks with conventional blade connections. Section 6 contains the necessary parameterization of the blade-disk connection in the form of a dovetail blade root. The loads resulting from the rotation are only applied to the top of section 5, which allows the finite difference model to consider only sections 1-5. Thus, for blisk design, section 6 can be neglected, and the method can be applied without further modification. The theory described here has been implemented programmatically and is known as NASA's engine weight estimate computer code (WATE) [30]. The WATE algorithm has

been integrated into the GTLab framework used in this thesis and can be used for predesign calculations. (See section 3.1) During the development of the program, a classification of different slice parameterizations was defined, which is still used today. A distinction is made between the disk types shown in Figure 2.7. A more modern

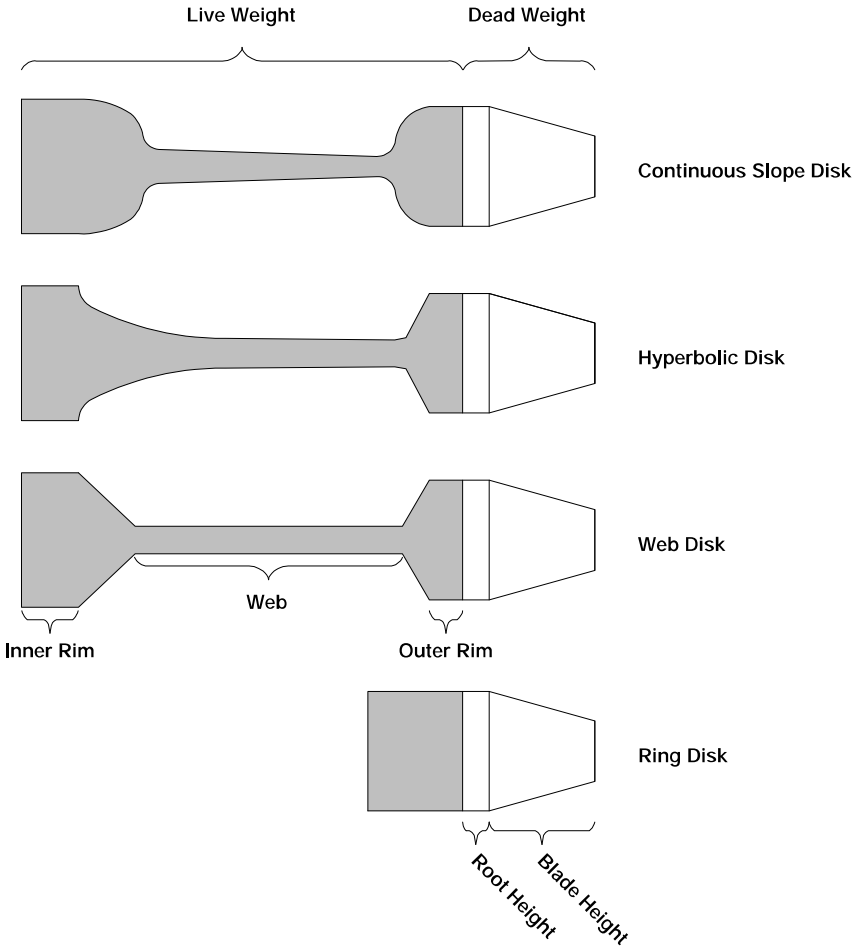


Figure 2.7: Different disk parameterization classes [31]

implementation of this method can be found at Gutzwiller [32] with

the program T-Axi Disk [31], which has a graphical user interface and converts geometries directly into a format readable by the FEM program ANSYS for validation. The finite difference method is very suitable for a preliminary weight estimation of compressor components and, due to the consideration of thermal stresses, also for turbine components. However, the calculated disks are only valid under the assumption of mirror symmetry. In modern engines, however, the flow channel often has very large gradients, which makes the disk geometry strongly asymmetric and the disks designed with the WATE code are no longer applicable without modification. An example of highly asymmetric disc geometries is shown in figure 2.8 in the next section.

Topology Optimization

As a special design method, topology optimization should be mentioned. This relatively new method has recently gained popularity mainly due to its very good suitability for the design of 3D printed components. The basic formulation of the theory can be found as early as 1904 by Michell [33] with the analytical solution for the best material utilization in steel beams from the point of view of structural analysis. Structures designed according to this principle are known today as Michell structures after their inventor [33]. However, the commercial application of the method to general problems became relevant only with sufficient computing power. A milestone in this respect was the release of the structure solver Optistruct by Altair in 1994 [34], which allowed topology optimization with lit-

tle effort and for general problems. Since then, much progress has been made in this area and new approaches to topology optimization have been found. Eschenauer and Olhoff [35] distinguish the following classes in their review on topology optimization:

- Homogenization method
- Perimeter method
- Macrostructure approach

For the design of disks, however, only the classical homogenization method can be considered due to the manufacturing boundary conditions resulting from section 2.1.1, since the newer methods always lead to undercuts and are strongly designed for the application of additive manufacturing [35]. The theory of topology optimization, especially all its subclasses, is very complex and will only be described in a basic way here. A more detailed explanation can be found in Bendsoe and Sigmung's compendium on topology optimization [36]. Similar to the previous sections, linear elasticity theory forms the basis of the method:

$$\epsilon_{ij} = E_{ijkl}\sigma^{kl} \tag{2.13}$$

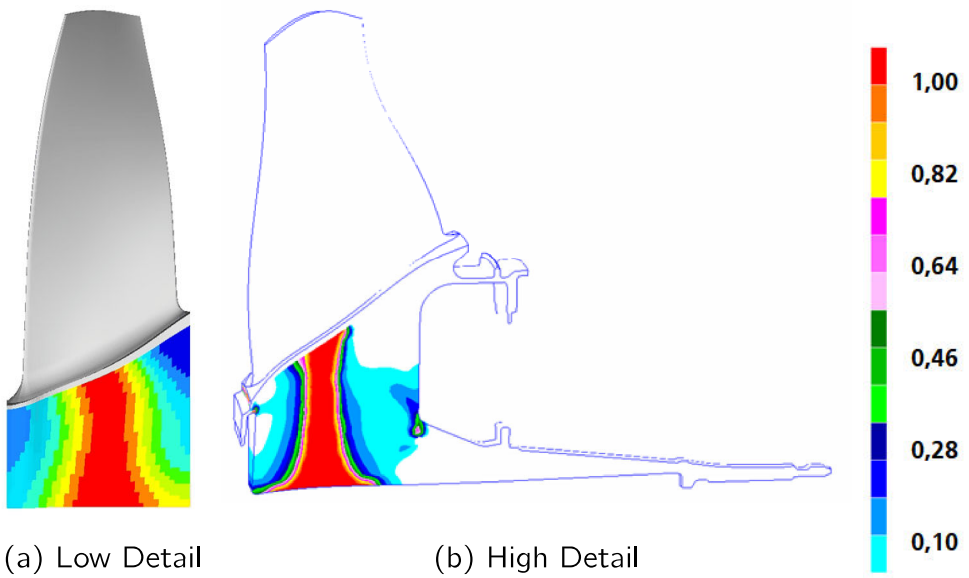
with the elastic tensor E. The optimization task is now the optimal choice of this elastic tensor, which is a variable of the design space. If this space is now divided into a finite number of elements, analogous to the FEM, the stiffness matrix K can be given as the sum of the stiffness of all elements $e = 1, .., N$ as follows:

$$K = \sum_{e=1}^{N} K_e(E_e) \tag{2.14}$$

The topology optimization now determines where in the design space material is needed and where no material is needed, i.e. the stiffness is high or zero. A possible formulation of this problem is the Solid Isotropic Microstructure with Penalty (SIMP) method

$$E_{ijkl} = \rho^p E^0_{ijkl} \tag{2.15}$$

where E^0_{ijkl} is the stiffness tensor of a given isotropic material and thus the maximum value of an element. The application of this method to the problem of disk sizing is shown in Figure 2.8a using the example of two different low-pressure compressor blisks. Al-



(a) Low Detail (b) High Detail

Figure 2.8: Topology optimization of an LPC blisk

though this method does not require disk symmetry, the transition to the blade platform is performed unrealistically. The runtime of the topology optimization of Figure 2.8a was 3 hours and 12 min-

utes[1] which is too high to compete with the computation times of FDM, if no significant gain in detail is achieved in the process. The creation of the model is also not trivial, since the manufacturing constraints must be met, i.e., undercuts must be avoided. Due to its formulation, the topology optimization always aims at a structural design that would only be feasible for wafers from additive manufacturing and would have to be suppressed for conventional processes by setting appropriate boundary conditions. Additive manufacturing processes for blisks will not be applicable in the short term future due to the safety-critical nature of the component, which is why topology optimization is not an appropriate means of solving the disk dimensioning problem, at least at the time of this work, and will be neglected in the following chapters.

Finite Element Method

The Finite Element Method (FEM) is the most promising method for solving the dimensioning of complex disk geometries with a high level of detail. The FEM is based on a similar principle to the Finite Difference Method and is essentially a method for solving differential equations. However, instead of dividing a physical space into finitely many difference quotients, it is divided into finitely many elements. The theory of FEM has already been sufficiently described in the literature and can be deepened for the interested reader either theoretically in Zienkiewic's fundamental work [37] or application-oriented in Klein [38]. The pure dimensioning of blisks using FEM

[1] All computational time measurements reported in this thesis were conducted on identical hardware: DELL Latitude Windows 10 I7 16 GB RAM

is simplified compared to the conventional blade-disk design from a static point of view, since the contact modelling of a positive blade root connection is omitted. However, the integral connection of blade and disk has almost no structural damping, so the isolated consideration of blade vibrations in the dynamic design is no longer permissible. Typically, the blade and disk are meshed incompatibly and connected through mathematical correlations with solver specific names like multipoint constraints (MPC) or ties. Numerous examples of blisk design and sizing using FEM can be found in the literature. However, to the best of the author's knowledge, there is a lack of highly automated and robust process chains that would be necessary to generate a surrogate model. In the conclusion of the paper [24] on which the analytical solution is based, the following statement can be found

> *Es erfordert lediglich einige Rechengeduld, beliebig viele weitere Lösungen[...]herzuleiten, um so einen möglichst umfassenden Katalog von fertigen Lösungen der technischen Praxis zur Verfügung zu stellen. - Grammel, 1936 [24, p.11]*

This quote resembles the current situation, where we have the technical possibilities for a general solution, but the implementation still requires a lot of manual effort. The method described in this thesis picks up here and develops a solution in the following chapters.

2.2 Data Science

After describing the prototype process component in detail, the following sections focus on the second key technology of the method to be developed, machine learning (ML) methods. ML methods are a subset of Artificial Intelligence (AI). Here, ML methods are the programmatic implementation that enables the AI to perform its task. This is done with specialized algorithms such as deep learning. An overview of the relationships is given in the figure 2.9. The

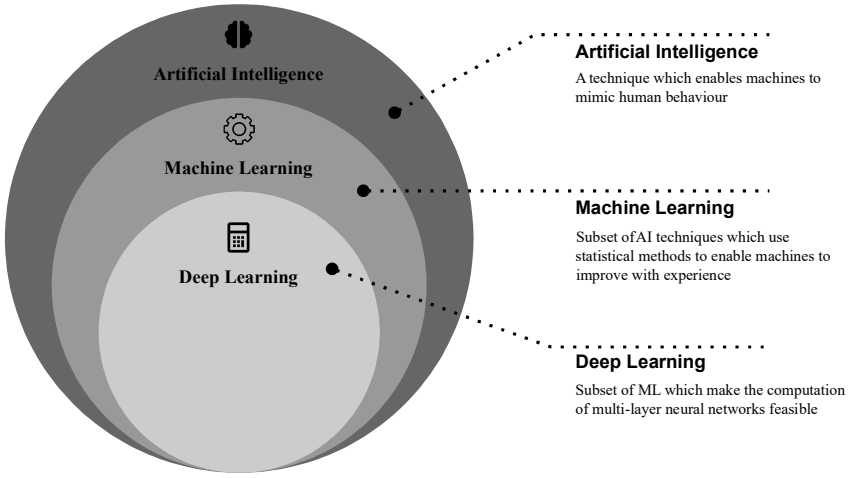


Figure 2.9: AI definition (derived from [39])

following sections first give a general overview of the state of the art and then describe the methods specialized for design processes.

2.2.1 Machine Learning

The term ML methods has evolved over time to encompass a wide range of different methods, some of which are highly specialized.

Roughly speaking, all ML methods use algorithms that analyse input data and predict different output values depending on the input data. The classification of methods depends mainly on how the data is prepared, how the method is processed, and how the program learns its behaviour. There are three main types of ML:

Unsupervised Learning

The input data for unsupervised learning (UL) requires relatively little preparation, since no classification or labelling of the data is required. Rather, the algorithm searches for patterns that are possibly unknown to the user. The engine determines correlations and relationships by analysing the available data. Most UL algorithms solve the problem by sorting or clustering the data. The main representatives of this genre are

- Adaptive resonance theory (ART)
- Self organizing maps (SOM)
- Different clustering algorithms

Applications include big data problems in statistics.

Supervised Learning

In supervised learning (SL), the user knows what patterns or classes to look for in the data provided. The data set used for training has been labelled so that the method can learn to classify unlabelled data in the future. SL methods distinguish between linear and non-linear

classification as shown in Figure 2.10. In linear approaches, the input and output data can be represented by linear relations, in non-linear ones by functions of arbitrary complexity. Disk sizing falls into the category of SL and nonlinear methods because different levels of abstraction, such as discretization, mean that linear relationships between input and output are not expected.

Reinforcement Learning

Reinforcement Learning (RL) methods, in contrast to UL and SL methods, do not require a training data set, but the so-called agent learns independently based on reward or punishment functions and thus gradually generates the database itself. The method works by trial and error and is most similar to human learning. Applications for RL methods include the control of complex systems such as the navigation of drones [40]. An incomplete overview of the algorithms of the individual ML methods can be found in 2.10.

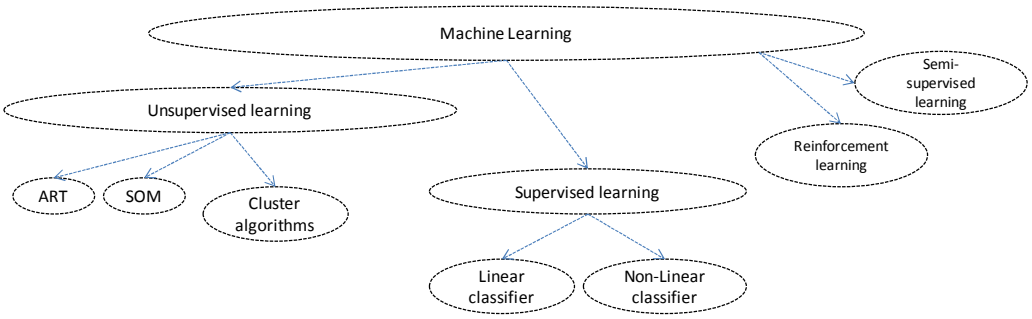


Figure 2.10: Taxonomy of ML methods [39]

2.2.2 AI Design Methods

The application of artificial intelligence in structural mechanics design is not new and can be dated back to 1989 where Adeli and Yeh [41] used a machine learning concept for the design of steel beams. The development of evolutionary optimization methods [42] and the use of logic-based surrogate functions such as kriging [43] can also be seen as a subset of the generic term AI. With the generation of ever larger amounts of data and newer training algorithms, a number of promising methods have emerged in the recent past that have the potential to revolutionize classical design methods. Predictive and generative design are two examples of the spectrum of data-driven design methods, with the latter already finding application in many areas due to advances in additive manufacturing processes.

Generative Design

The Generative Design (GD) subset falls mainly into the UL category, although a recent study [44] RL methods also show promising results. Based on the input data, the GD algorithm attempts to automatically find correlations and patterns that can be used to find valid solutions to the design problem according to the constraints. Generative Adversarial Networks (GAN) are particularly well suited for this purpose. Examples of applications can be found in architecture ([45], [46]), aerospace [47], and automotive [48]. The solutions developed using this method are very similar to the topology optimization results described in the 2.1.2 section, since similar formulations are used to find the load path. Figure 2.11

shows a generative design of a seat bracket developed by General Motors. According to [49], the 150 designs developed for the 3D

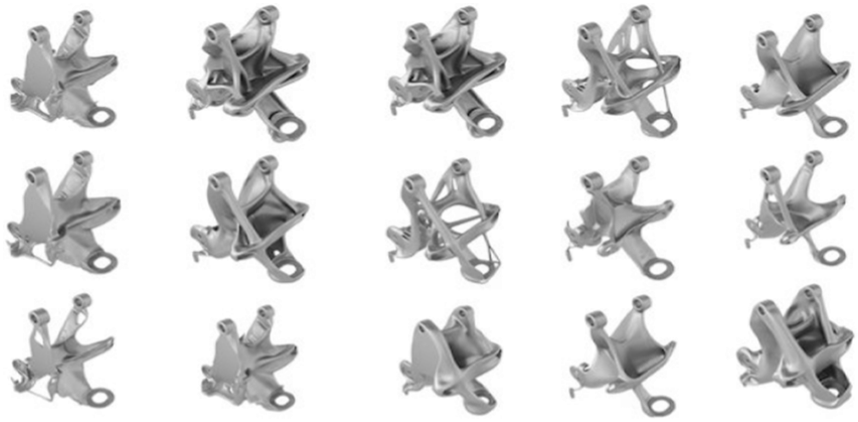


Figure 2.11: Generative design example [49]

printed component are 40 percent lighter and withstands 20 percent more stress than the design used previously. In addition to the topology optimization method of iterative material removal, recent research has explored methods such as biomimecry [50] or morphogenesis [51], which follow the example of growth to iteratively add material. GD, as with UL methods, gives good results when the desired design is unknown or when different approaches to the solution space are to be explored. An application to the disk dimension problem is therefore not very promising.

Predictive Design

Predictive Design (PD), on the other hand, can be classified as an SL method because the user already has data that contains optimal designs for certain parameter sets. The ML model is supposed to learn from this data to develop designs input parameters that are

not present in the training data set. A method comparison for SL design methods at Sharpe [52] shows good results for classification and regression problems using the following ML methods:

- Convolutional neural betworks (CNN)
- Bayesian networks (BNN)
- Random forests (RF)
- Support vector machines (SVM)

In addition to the ones mentioned above, the procedure of Kriging is investigated in this thesis. The detailed procedure can be found in the chapter 6.

2.3 Method selection

Since milling technology offers a high degree of automation, this thesis will focus on this manufacturing method. To achieve the necessary geometric detail in disk simulation, finite element analysis (FEA) appears to be the most promising approach. Coupled with optimization techniques, the optimal disk shape can be determined in a fully automated manner, generating the data needed to train an efficient model. As with most expert systems, a supervised approach is most promising in training the machine learning model, which will be explored to enhance design speed while ensuring a lightweight design.

3 Development Framework

The ML methods described above require large amounts of data for training, which can only be generated through automated processes. To develop robust and repeatable processes, a professional software implementation is required, which in turn requires appropriate user environments and frameworks. The following sections describe the software infrastructure used, as well as data formats and standards.

3.1 The GTlab Framework

The Gas Turbine Laboratory (GTlab) [53] is a design platform for engines and gas turbines used in both aviation and stationary power generation. It is being developed under the leadership of DLR's engine department (TWK) and is already being used as a multidisciplinary framework in numerous DLR institutes. GTlab is developed in C++ and uses the comprehensive Qt class library. The process control has a Python interface, which allows direct access to common Python libraries, e.g. for visualization. The interface is made possible by the Meta Object System (MOS) provided by Qt [54]. To realize the multidisciplinary design in GTlab, the application is modular and follows the plugin principle. It consists of a core framework

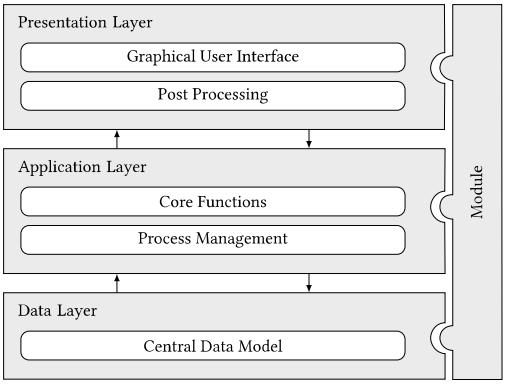


Figure 3.1: Modular software architecture [53]

that contains the basic functionalities of the application and has various interfaces. Modules can be attached to this framework to extend the functionality of the software (see figure 3.1). GTlab uses a central engine data model to store the data related to an engine or gas turbine system [55]. The data model is implemented using the well-established Unified Modelling Language (UML), which is suited for describing complex systems [56]. The formulation of the central data model enables different departments to work together at different levels of detail.

Using the example of an engine design, a thermodynamic model is first created using the performance module. This includes top-level requirements such as total thrust and installation space, as well as the various operating points derived from the mission profile. The general component structure can thus be largely defined and is shown for a generic Ultra High Bypass Ratio (UHBR) engine

in Figure 3.2. Based on this model, predesign calculations can be

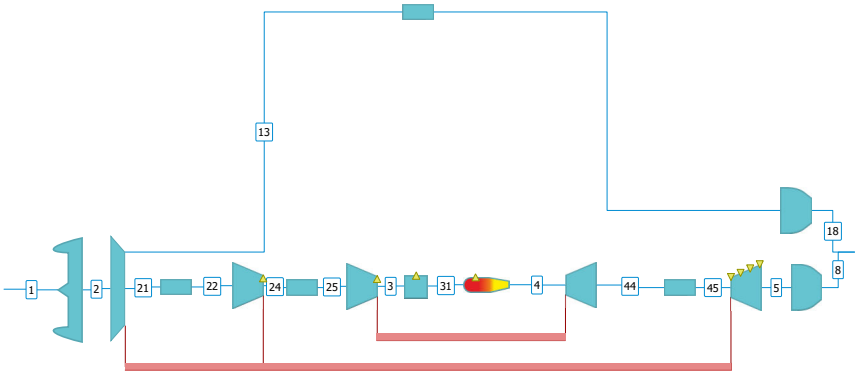


Figure 3.2: Generic UHBR Performance Model

started using the Predesign module or even detailed analyses. The predesign processes already generate 3D blade geometries using the blade parameterization described in section 4.1.2 via an interface to the BladeGenerator. The Predesign model of the generic UHBR model is shown in figure 3.3. The blade geometries can now be evaluated by various disciplines with their module functionalities and extended by detailed features such as cooling systems, fillets or, as in this case, compressor disks.

3.2 Structural Mechanic Module

The Structural Mechanics Module (SMM) is a novel GTlab module developed in the department for Design and Manufacture Technologies of the DLR, which provides processes for the structural mechanical design and evaluation of engine components via the

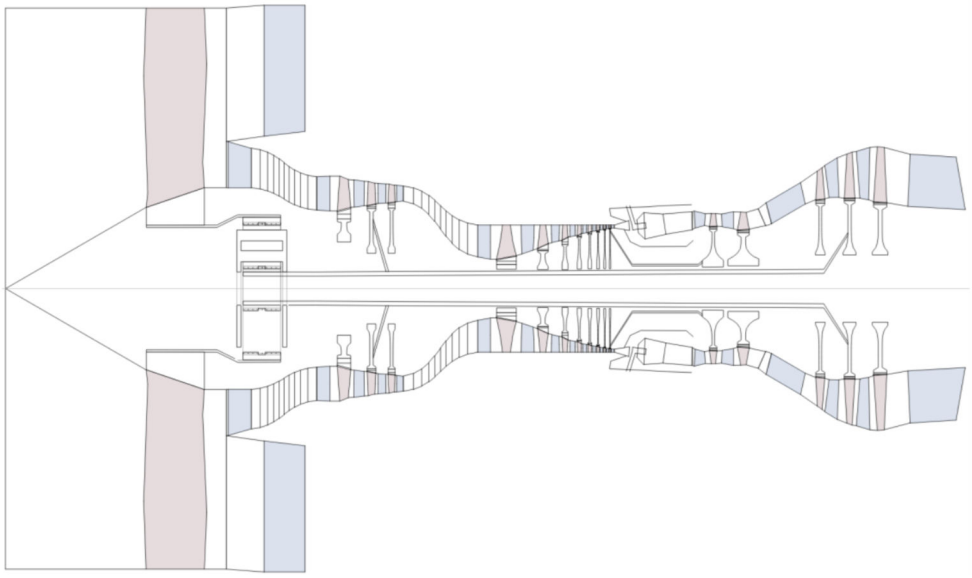


Figure 3.3: Generic UHBR PreDesign model

module interface (Figure 3.1) in GTlab. The processes described in this paper are a subset of the SMM process library. The SMM concept is intended to provide a long-term evaluation capability for engine components through standardized processes throughout the entire development process. The state of development at the time of writing, as well as the identification (blue arrows) of the processes developed in the context of this thesis, is shown in figure 3.4.

3.3 Data Standards

Standardization of data formats is an essential building block in the development of virtual engineering workflows. In other fields, such as aircraft design, the Commom Parametric Aircraft Configuration

Schema (CPACS) standard is already established, although the engine is idealized as a mass and suspension position [57]. In the field of engines, in the CFD world, the Computational Fluid Dynamic General Notation System (CGNS) standard is widely used for the storage and retrieval of CFD data. Newer CGNS files use the Hierarchical Data Format (HDF5) as base format, which can be seen as a data model template, library and file format. The hierarchical structure mimics a filesystem and can be viewed using hdf5 viewers. Since it is a binary file, read and write operation speeds are very fast and can even be increased with data chunking support. In the structural mechanic community, a standardized data format has yet been established. Coordinated by Fraunhofer SCAI and with the support of well-known industry representatives, the VMAP standard has been proposed [58]. VMAP is also based on the HDF5 format and extends it with definitions for material data, geometry and discretization standards. For GTlab, an HDF5 module has been developed that allows visualization of the file structure and provides file handling functions [59].

3.4 REBAR Framework

For the training of the replacement model, a framework developed at DLR is used, which was created in the context of the digitization initiative for artificial intelligence in cooperation with several DLR institutes. Under the working title Reducing Barriers for AI in (applied) Research (REBAR), the framework provides a toolbox for

the application of ML methods to simplify their scientific use. The framework works in a pipeline-oriented manner and has been designed for heterogeneous use cases based on four very different use cases, one of which is the method described in this paper. The ML pipeline is represented as a directed acyclic graph (DAG) and uses the following already proven applications in the core architecture [60]:

- Apache Airflow: Web server and scheduler
- Celery: Broker
- MLFlow: Web server and API
- Postgres and MinIO: Common components

In addition to the programming implementation of the framework, a web platform with training and best practices guidance will be made available to users. The framework will also be released as an open source application in the future.

With the framework available, the automated blisk design needs to be created. This will be the scope of the following chapter. Starting from the geometric description of a blisk, meshing and subsequent model building.

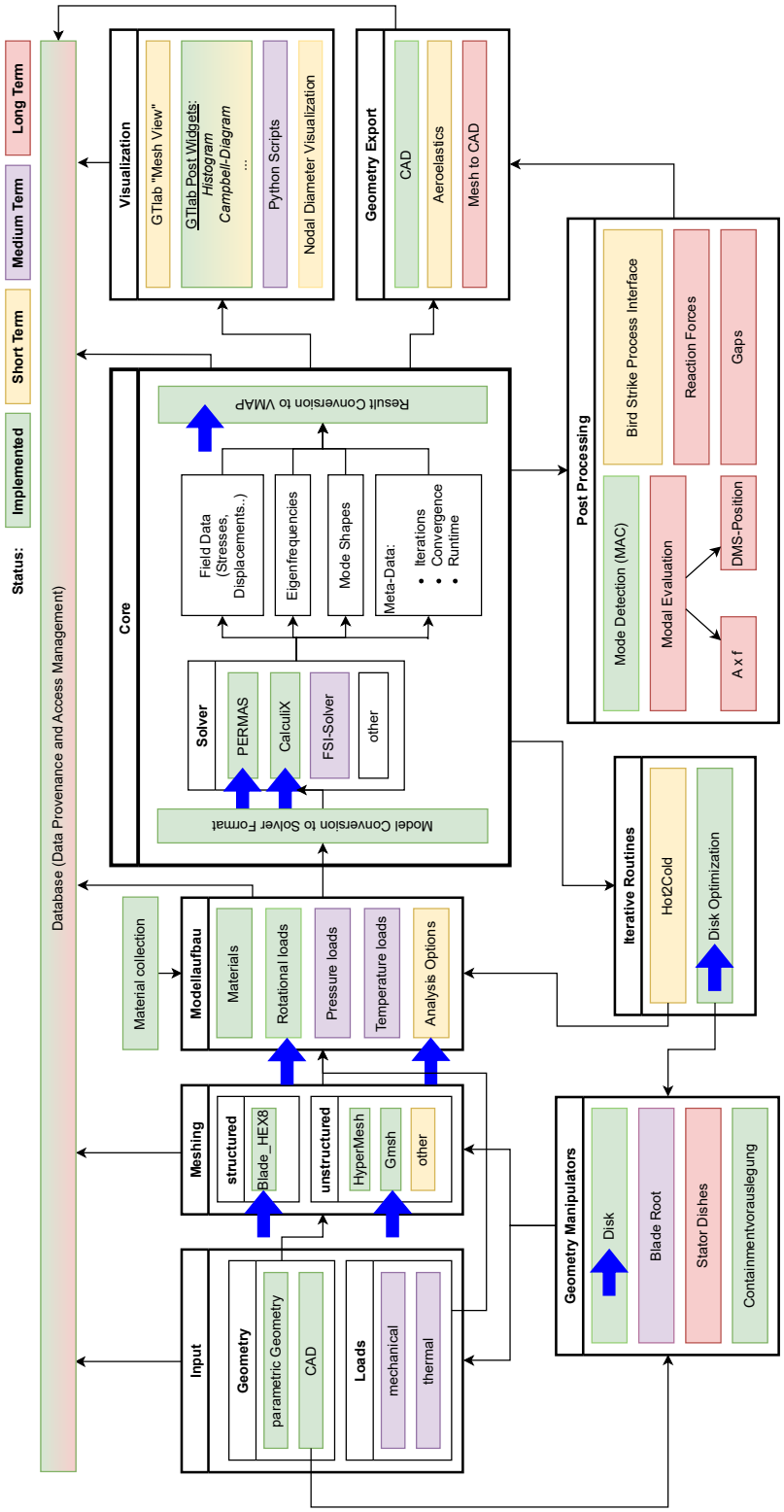
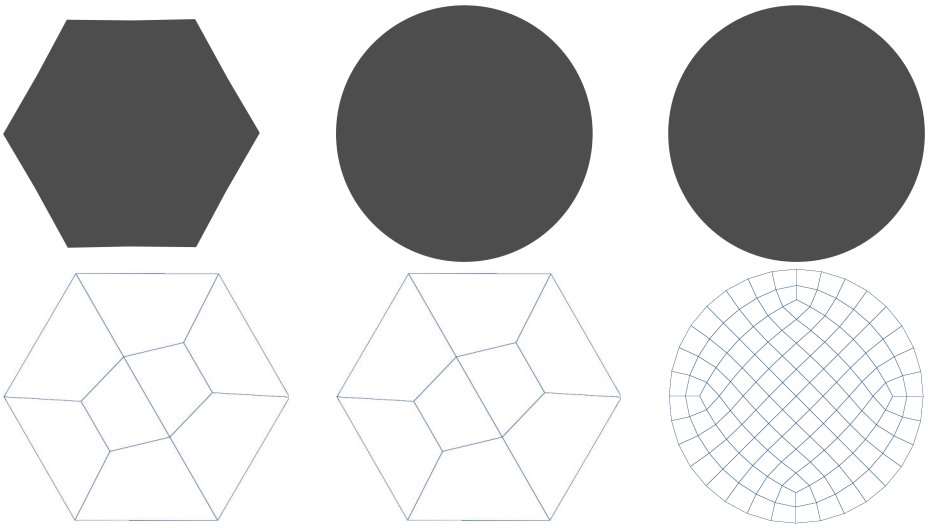


Figure 3.4: SMM development concept and status

4 Discretization

A basic prerequisite for an automated, robust and highly detailed disk design is the selection of a suitable geometry parameterization and an appropriate discretization for the FEM calculation. The number of free geometry parameters usually has a quadratic effect on the computational time of the optimization, and the fineness of the discretization on the computational time of the FEM. The parameterization of a geometry is not necessarily synonymous with its simplification, but if the number of parameters is greatly reduced, this is often unavoidable. The initial state can then no longer be restored with the same accuracy. To illustrate this, figure 4.1 shows an example geometry parameterized and discretized at different levels of detail. It becomes clear that the accuracy of the description depends on the geometry description as well as on the mesh resolution. If figures a and c are good representations of the geometry and its discretization, information from figure b will be lost. In the following sections, a solution is worked out to describe the geometry sufficiently accurately with the smallest possible number of parameters and to provide all necessary information to solve the dimensioning problem.



(a) Coarse-Coarse (b) Dense-Coarse (c) Dense-Dense

Figure 4.1: Different geometry and mesh parameterization details

4.1 Geometry

The description of the geometry is always the starting point of the FEM simulation, but as shown in Figure 4.1 it has a direct impact on the required level of detail of the mesh. The general parameterization of different geometries has a long history of development and, due to its wide range of applications, occupies a whole branch of industry under the collective term Computer Aided Design (CAD). Geometries are described by parametric curves and surfaces. While basic geometric shapes such as rectangles and circles are sufficient for simple applications, splines and spline surfaces derived from them become necessary for more complex geometries [61]. A single definition of the coordinate system is used for all the processes and applications described below. For Cartesian coordinates, the X axis

is always on the machine's rotational axis and points downstream. For cylindrical coordinate systems, the same is true for the Z axis.

4.1.1 Splines

The term spline is first found in 1946 as a name for a smooth harmonic composite curve of third degree [62], but is now used for any piecewise polynomial curve. In general, splines are capable of approximating arbitrary curves, although a distinction must be made between interpolating and approximating splines. Over time, many spline formulations have been developed, but almost all of them are based on the formulation of the so-called basic spline, or B-spline. The B-spline is a piecewise polynomial, where each curve can be of different degree p. The curve defined in this way always lies in the convex hull of the control points, where the change of the control points only affects the interval $[u_i, u_{i+p+1}]$. The surface analogous to the B-spline is the B-spline surface. This formulation is particularly robust in application because changes in the coefficients have only a local effect. For the approximation of a general curve C at point u, the following applies

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \tag{4.1}$$

with the control points P_i, also called de-Boor points, the degree p, the B-spline basis functions $N_{i,p}$ and the purely increasing node vector $U = (u_1, u_2, .., u_m)$. The length of the node vector depends on the degree p and the number of nodes as $m = n + p + 1$. The

basis functions depend only on the node vectors and are defined recursively via

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

at $p = 0$ and

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{4.3}$$

are defined for $p > 1$. The formulation is valid for both 2D and 3D and is adjusted by the dimension of the control points. The resulting basis functions are shown in the figure 4.2. A special form of this

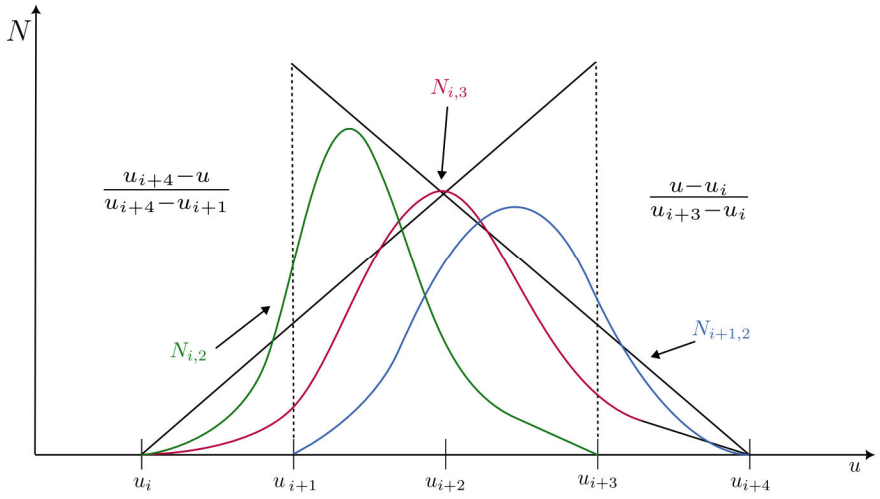


Figure 4.2: B-spline basis functions [39]

B-spline formulation are Non-Uniform Rational B-Splines (NURBS), where the control points get an additional parameter, the weight.

The relation (4.3) thus becomes the rational basis function

$$R_{i,k}(u) = \frac{N_{i,p}(u)w_i}{\sum_{j=0}^{n} N_{i,p}(u)w_j} \tag{4.4}$$

with weights w. These control the strength of the attraction, making the NURBS a rational curve. When all weights become one, the NURBS curve becomes identical to the B-spline formulation. The Euler diagram in Figure 4.3 illustrates the relationships between the different spline classes. NURBS were originally developed for the parameterization of free surfaces in the automotive industry and are now an integral part of CAD and manufacturing software (CAM), as well as part of numerous format specifications such as the Initial Graphics Exchange Specification (IGES) or the Standard for the Exchange of Product Model Data (STEP). Smooth surfaces require NURBS surfaces with at least G1 continuity, also known as tangent continuity, where the tangent at the end of each spline segment matches the tangent of the next spline segments start. For even smoother surfaces, G2 or curvature continuity can be required. This refers to the second derivative and implies that the rate of change of the tangents matches at the connection of two spline segments [63]. In general, the interactive editing of B-splines and NURBS curves and surfaces via their control points is very intuitive and predictable. The blade and disk geometry parameterizations described in the next sections use a B-Spline formulation for their 2D parameterization. When 3D properties, like volume or centre of gravity are called, GTlab's built-in CAD kernel converts the surfaces to solids using NURBS.

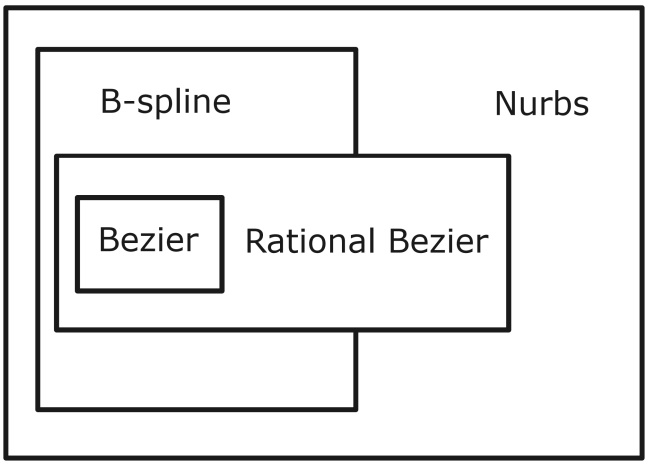


Figure 4.3: Spline classes

4.1.2 Blade Parameterization

As described in section 1.2, the starting point for the calculations is the blade geometry, which is primarily dimensioned according to aerodynamic properties such as aerodynamic efficiency or pumping distance. The geometric description of compressor and turbine blades is a prerequisite for blade optimization and has a long history of development. Numerous publications can be found from universities and industry representatives, such as Rolls Royce [64] or Shi [65]. These methods all use a similar approach. In the present work, a method developed at the DLR Institute of Propulsion Technology in the Fan and Compressor Department (AT-FUV) is used for blade parameterization, the BladeGenerator [66]. The parameterization is based on individual radial profile sections of the blade. A profile is divided into 4 zones, leading edge (LE), trailing edge (TE), suction side (SS) and pressure side (PS). The B-spline formulation

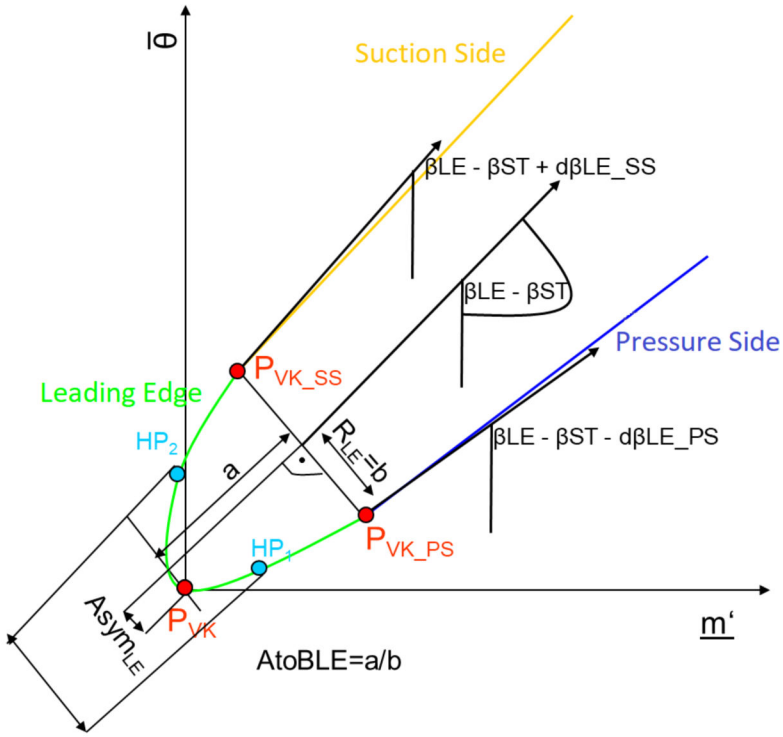


Figure 4.4: Blade profile parameterization[66]

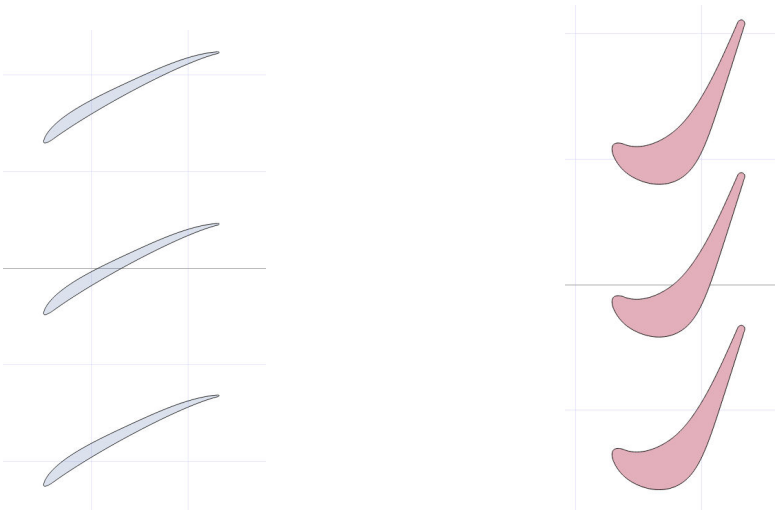
described in the previous section is used for the construction lines of these zones and connected to a closed curve with G2 connectivity. To realize the description with as few parameters as possible, this is done in a specially defined $\underline{m'}, \underline{\theta}$ coordinate system. Here $\underline{m'} = 0$ defines the LE point, $\underline{m'} = 1$ the TE point and the angle $\underline{\theta}$ the coordinate in radians surrounding the profile. The conversion first to m', θ and then to x, y, z coordinates is angle fidelity and can be done by a linear transformation. The LE and TE points and their transitions (P_{LE-SS}, P_{LE-PS}..) to the PS and SS zones are now defined by β angle specifications. Then the suction side is defined using the spline properties described in the previous section, i.e. the

location of control points, knot vectors and weights. The degree used is always 3. There are now 3 ways to describe the pressure side:

- Control points analogous to suction side
- Thickness distribution relative to suction side with respect to control points
- Analytical thickness distribution relative to suction side

Method 2 was found to be the most robust design variant and could be successfully used in optimizations [67]. After the SS- and PS-spline have been completely described, the edge splines can be generated by specifying a few parameters and continuity boundary conditions, and the 4 individual curves can then be combined into one curve. The parameterization of the profiles is extremely robust and flexible, allowing both compressor and uncooled turbine blade profiles to be easily described. (Figure 4.5) The profile description can be repeated on any number of radial sections to achieve the desired accuracy. Usually, only 3 sections are used in the pre-design process, since a high degree of freedom can already be achieved in the design. The 2D profiles are then transformed into 3D space so that they lie on the surface of revolution of their transformation curve. The profiles are then stacked in the r-direction using defined construction lines to create a B-spline surface. This is done by the so-called surface skinning method, a detailed description of which can be found in Woodward [68]. The profiles can be scaled, rotated or moved arbitrarily, which also provides a high degree of flexibility

in generating the 3D plate geometry. After the B-spline surface is generated, the blade can be blended at the hub and/or tip to create complex flow channel contours or even gap variations. The design of the fillet between the blade and the disk is structurally relevant. This can also be set in the BladeGenerator via various parameters. For this purpose, the profiles described above are transformed into 3 control curves. The curve shown in red (BladeCurve) in the figure 4.6 marks the beginning of the fillet, the green curve the outline of the fillet at the hub (BlowCurve). The blue profile must be extended to the green curve as shown in the figure 4.6. In addition, the shape of the fillet can be arbitrarily shaped using various parameters, which is also possible using a B-spline called ShapeCurve.



(a) UHBR LPC S2 (b) UHBR HPT R1

Figure 4.5: Diversity in profile geometries

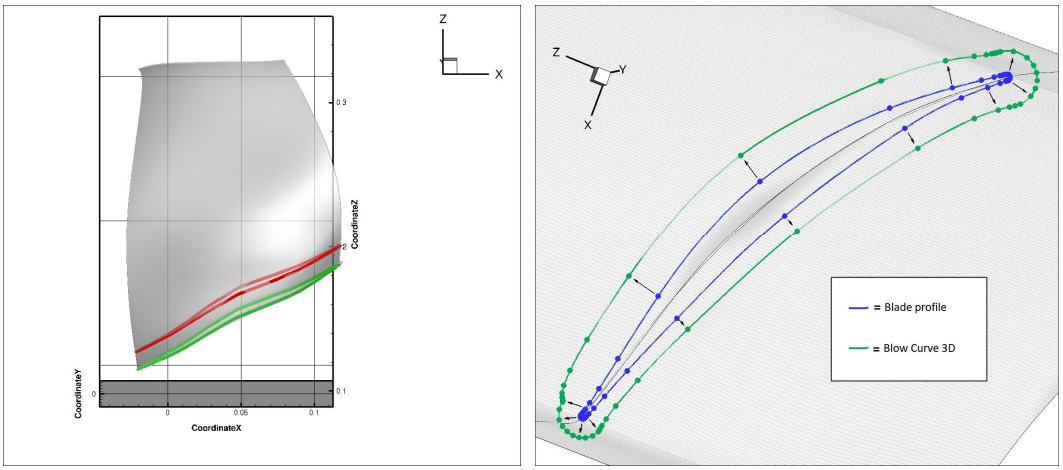


Figure 4.6: Fillet construction curves [66]

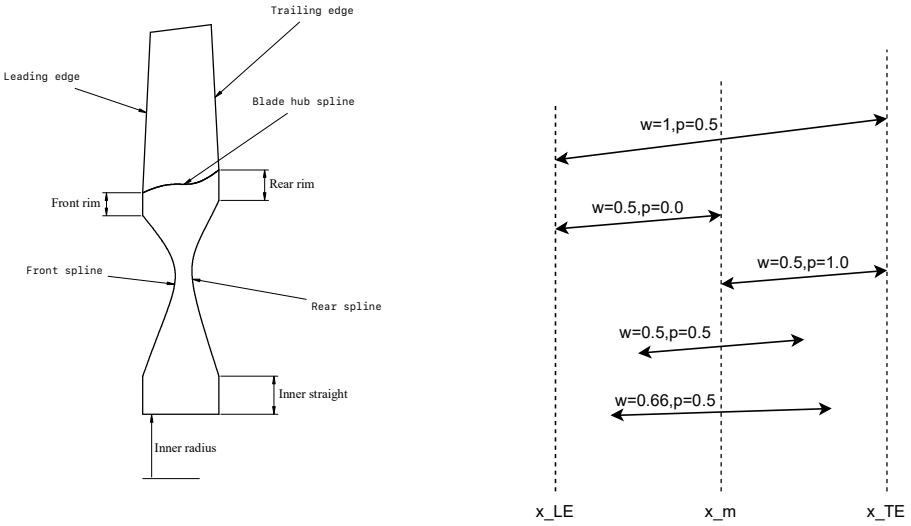
4.1.3 Disk Parameterization

The parameterization of the disk geometry has a large impact on the complexity of the optimization problem. The challenge is to achieve a high level of detail with as few parameters as possible. The goal is to eliminate the shortcomings described in the 2.1.2 section. These were mainly the axial symmetry of the disk and the straight line pattern. With these requirements, the following basic conditions can be imposed on the disk parameterization:

- Rotation Symmetry
- Full contour with no holes or inner segments
- No undercuts
- No holes or other elements required for assembly

Specifying rotational symmetry allows to describe the disk contour in the 2D coordinate system r, z. The contour must always be a

closed curve, but can be composed of any number of segments. Two contour classes are defined to allow comparison with the Predesign method described in section 2.1.2. The polygon class is described by connected polygon segments, and the spline class uses NURBS for the disk leading edge (DLE) and the disk trailing edge (DTE) described in section 4.1.1. The disk consists of the sections shown in the figure 4.7. The blade-disk transition for both classes can be



(a) Segment definition (b) Pairwise point examples

Figure 4.7: Disk parameterization

described as a straight line, polygon or spline and is derived from the radial outline of the lowest blade profile. At the same time, the first and last points of the blade in the flow direction are calculated and the axial dimensions of the disk are predefined. It should be noted that the axial extreme points are not necessarily identical with the definition of LE and TE of the blade in the previous section.

Following the hub line, each disk has a platform whose thickness can be varied up to half the radial height of the disk. Optionally, analogous to the description in section 2.1.2, a parallel section on the inner diameter is provided for manufacturing purposes. The geometrically unambiguous description of this contour must now be done with as few parameters as possible. The control points of the DLE and DTE splines are therefore described in pairs. This has two advantages: first, overlapping of the splines is avoided, since the formulation requires an axial distance of the control points of $\delta x < 0$, and second, two points in 2D space can be described with only two parameters. For each pair of control points numbered i, the parameters ring width w_i and ring position p_i must be specified. From these, the 2D position of the control point can be calculated using the relations

$$x_i = x_{LE} + (1 - w_i) * x_m * p_i \tag{4.5}$$

and

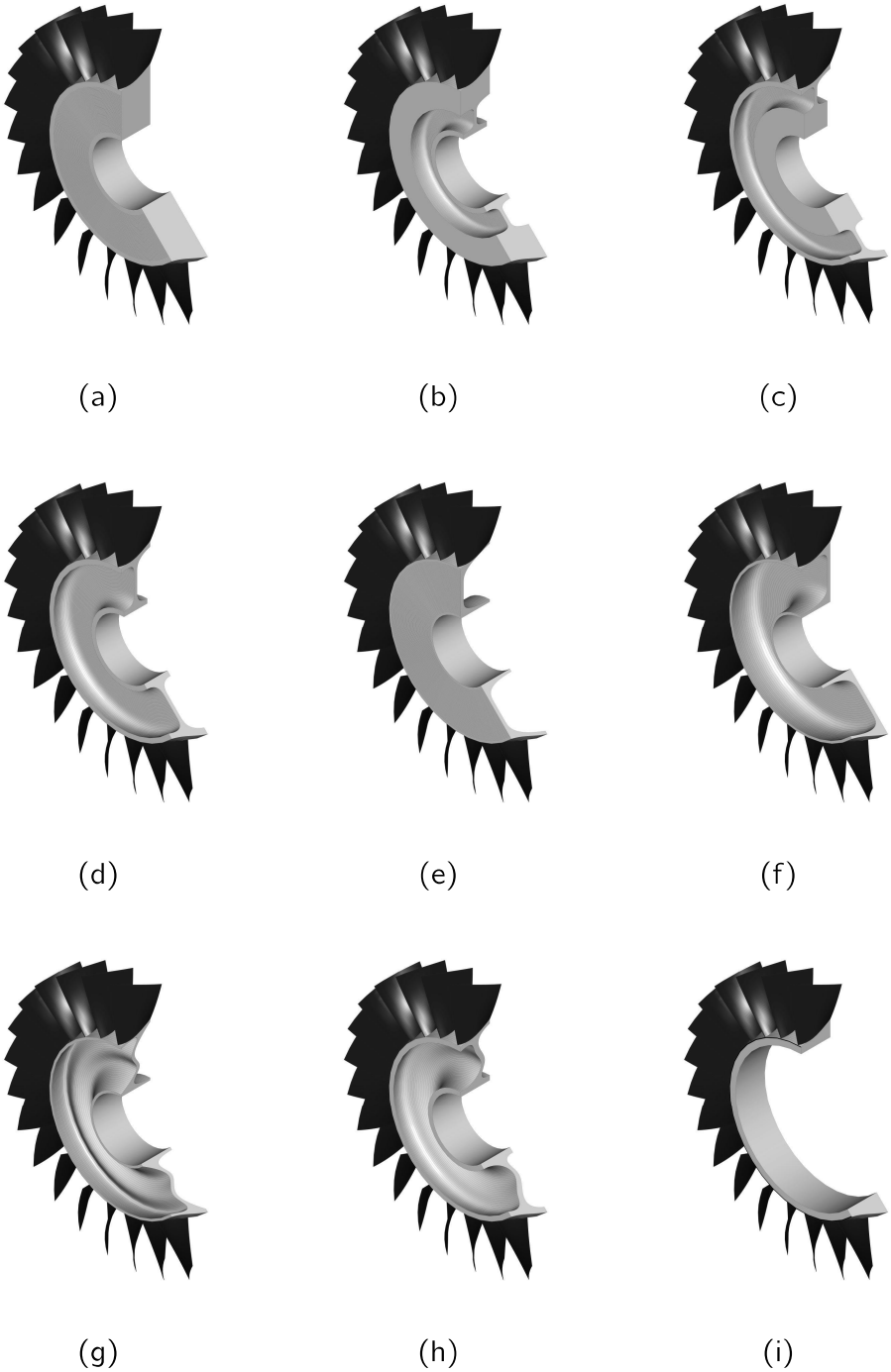
$$r_i = r_{max} + (1 - w_i) * r_m * p_i \tag{4.6}$$

with the mean axial disk coordinate x_m and the mean radial disk coordinate r_m. The necessary parameters for the complete description of the disk are summarized in the table 4.1. You can see that only the r_{Bore} parameter is dimension dependent. All other parameters scale from this parameter and the dimensions given by the plate. Despite the small number of parameters, the variation of these parameters allows a high degree of freedom in the design of the plate. A selection of possible parameter combinations and the resulting

Parameter	Description	Range	Unit
w_i	Width of ring i	0.0-1.2	-
p_i	Position of ring i	0.0-1.0	-
h_{RimDLE}	Height of DLE rim	0.0-0.5	-
h_{RimDTE}	Height of DTE rim	0.0-0.5	-
$h_{\|\|}$	Height of inner straight part	0.0-0.5	-
r_{Bore}	Bore radius	0.0-r_{max}	m

Table 4.1: Disk design parameters

disk shapes is shown in Figure 4.8 on the following page. In the extreme case, the platform extends to the inner radius (Figure 4.8 a) and fills the entire installation space. The inner radius can range between the extreme values of 0.0 to approximate the hub contour from below; the former corresponds to a solid disk without recess, the latter to a special type of disk called a bladed ring (BLING) (Figure 4.8 i). When the spline pitch parameters and the inner disk length are close to 0, the minimum installation space is reached. An example of the extensive design freedom can be achieved by varying the length and position parameters (Figure 4.8 b-h).



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.8: Design freedom with the underlying parameterization

4.2 Meshing

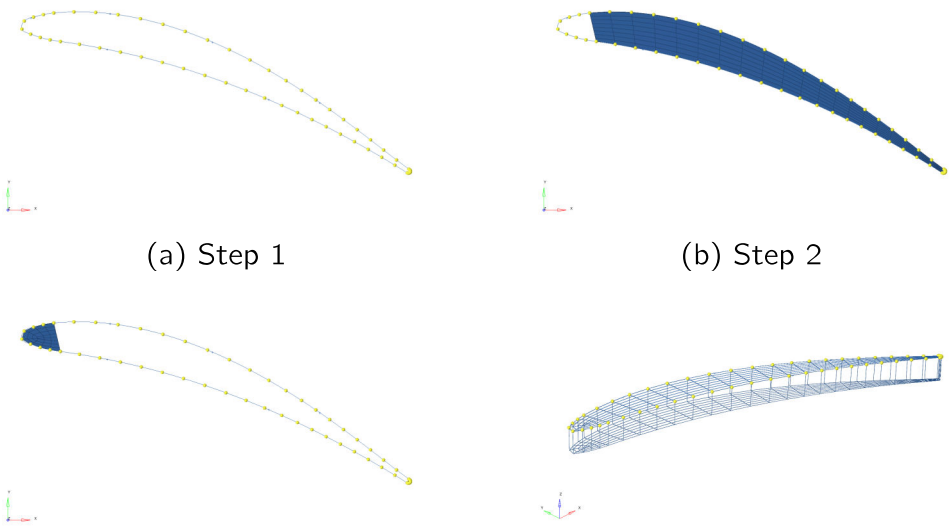
In the introduction of this chapter, the effects of unsuitable geometry and discretization accuracy were already pointed out. In order to achieve a robust process, the meshing methods are adapted to the geometry. Eight-node hexahedron elements (HEX8) are used as FE elements, since the small number of elements required allows for very fast computations. The advantages are explained in section 5.2 The following sections describe the specifics of the meshing methods for each geometry.

4.2.1 Structured Blade Mesh

The generation of the blade mesh directly accesses the geometric parameterization. The meshing process can be described in four steps as follows and is illustrated in the figure 4.9:

1. 1D Profile-mesh

The first step is to generate a line mesh on the spline curves of the profile parameterization described in section 4.1.2. The number of generated points can be chosen freely in pairs for the LE and TE area as well as for the PS and SS area. Since the profile section is located on a surface of revolution with a certain radial height, the points are output in the cylindrical coordinate system.



(a) Step 1 (b) Step 2

(c) Step 3 (d) Step 4

Figure 4.9: Blade meshing process

2. H-Grid

Next, 2D meshing of the centre H-block is performed. The opposite points of PS and SS are linearly connected in the cylindrical coordinate system and intermediate nodes are created according to the desired mesh resolution in thickness direction. The formulation in the cylindrical coordinate system ensures that all points lie within the surface of revolution of the profile section.

3. C-Grid

The meshing of the LE and TE regions is much more complex. The geometry can be idealized as a distorted semicircle, as different types of meshing have been developed for this geometric shape. The type of mesh topology used here is called butterfly mesh [69] and

must be done differently depending on the ratio of mesh resolution parameters. The determination of a reasonable point distribution is solved here by the method of transfinite interpolation (TFI), which is described in great detail in Nguyen [70] or Eisenmann[71]. Usually, the edges of the domain to be meshed in the 2D coordinate system are transformed exactly to the edges of the unit square in the η, ξ coordinate system. The construction of the linear 1D interpolation can be done with

$$\Pi_x R = (1 - \xi)R(0, \eta) + \xi R(1, \eta) \tag{4.7}$$

$$\Pi_\eta R = (1 - \xi)R(\xi, 0) + \eta R(\xi, 1) \tag{4.8}$$

The 2D interpolate is then calculated via the Boolean sum

$$\hat{R}(\xi, \eta) = (\Pi_\xi \oplus \Pi_\eta) = (\Pi_\xi + \Pi_\eta - \Pi_\xi \Pi_\eta)R \tag{4.9}$$

In the case of the C-block meshing required here, the leading and trailing edges are each divided into upper and lower domains based on the blade chord, resulting in 3-sided domains. The special case of TFI for 3-sided domains is described in Baart [72] and more generally in Salvi [73] and is called Coons patch. Here, one side of the unit square is degenerated and considered as a single point. The nodes created in 2D must then be moved to the correct radius using a profile-dependent r, z function.

4. 3D-Mesh

Steps 1-3 are repeated at different radial heights depending on the desired number of nodes, and the nodes of each profile are connected linearly. The identification of the required node sets can be calculated analytically from the mesh parameters in the case of structured meshing. Relevant NSets for the process are

- N_{hub}: Node of the blade hub
- N_{surf}: Surface node of the blade
- N_{tipLE}, N_{tipTE}: Leading and trailing tips of the blade

When meshing a blade with a fillet, the uniform r-distribution of the profiles can lead to poor mesh quality. To counteract this, the distribution in the r-direction can be varied using different BIAS settings. The implementation follows a methodology developed by Altair [34] for the Hypermesh program. Three bias modes are provided: Linear, Exponential and Bell Curve. For linear biasing, the element placement is controlled by the slope of a line with interval $[0, 1]$. With element density n and $s \in 0, \frac{1}{n}, \frac{2}{n}, .., \frac{n}{n}$, a node position determination function can be formulated as follows

$$x(s) = s\frac{ms + 2b}{m + 2b} \tag{4.10}$$

The intensity m can be used to control the strength of the bias, with a reasonable range defined by the constant $b = 1.5$ to $[0, 20]$. A stronger effect can be obtained by using an exponential distribution

function. The interval lengths for the exponential distribution are

$$\phi(0) = 1, \phi(\frac{1}{n}) = c, \phi(\frac{2}{n}) = c^2, .., \phi(\frac{n}{n}) = c^n \tag{4.11}$$

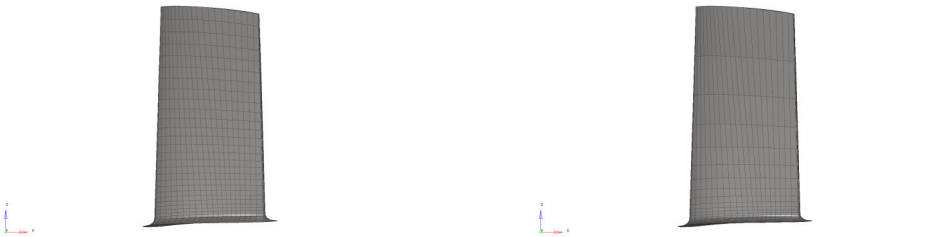
where c is defined as a function of the bias intensity m.

$$c = 1.0 + 0.1m \tag{4.12}$$

The node position can thus be determined by the function

$$x(s) = \frac{C^{ns} - 1}{C^n - 1} \tag{4.13}$$

Bell-curve biasing has been implemented as a third variant, but is in this context only relevant for stators and will not be described here. The effects of biasing and the positive effects of using fillets are shown in figure 4.10. The developed process behaves very robustly



(a) Linear Biasing m=20 (b) Exponential Biasing m=3

Figure 4.10: Biasing Examples

in tests and generates meshes in an average process time of 450ms, making it suitable for optimization processes. A high quality blade mesh can be generated by an appropriate choice of mesh parameters. The measurement of mesh quality is based on mesh quality

parameters. In the case of HEX8 elements, the Jacobi determinant, the maximum corner angle, and the aspect ratio of the shortest to the longest side are particularly meaningful. The optimal HEX8 element is a cube, and therefore the typical values for good element quality should be within the limits given in the table 4.2. The Ja-

Criteria	**Valid Range**	**Optimum**
Jacobian Ratio	0.6-1	1
Aspect Ratio	1-50	1
Max. Corner Angle	90°-150°	90°

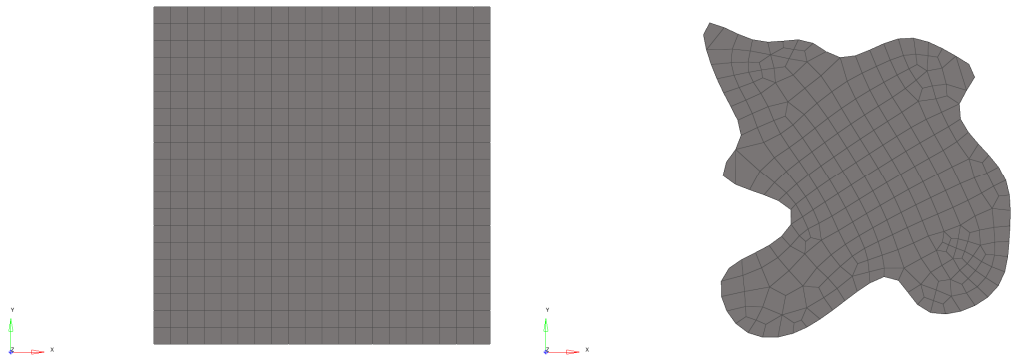
Table 4.2: HEX8 element quality ranges

cobi ratio is calculated from the determinant of the Jacobian matrix. For 3D elements, this is a 3x3 matrix. The determinant depends on the point at which the calculation is performed and is performed differently in different FE software, but usually the Gauss points of the element are used. For HEX8 elements with eight gauss points, the Jacobian determinant has eight values, of which the ratio of the smallest to the largest is calculated. Thus, the Jacobian ratio is always in the range 0-1 and becomes 1 in the case of a cube. This definition of element quality also applies to the following section.

4.2.2 Unstructured Disk Mesh

For the meshing of the disk geometry, the structured meshing method of the blade is not applicable, because the geometry was parameterized very flexibly. The large changes in the cross-section (see figure 4.8) would lead to strongly compressed elements and thus

to poor element quality. The different design of the contour description with splines, polygons and variable transition to the blade leads to many curve combinations and curve numbers in the contour description. The crucial feature is the rotational symmetry, which allows a 2D meshing of a r, x section, which can be rotated afterwards to get 3D elements. To obtain 3D HEX8 elements, the slice must be meshed as a pure unstructured quad mesh. The difference between structured and unstructured meshing is shown in figure 4.11. While the position and number of a node can be



(a) Structured mesh (b) Unstructured mesh

Figure 4.11: Quad mesh examples

uniquely calculated from the input parameters in structured networks, this is not possible in unstructured networks. The criteria for selecting a suitable quad meshing method are robustness, speed, and open source availability. The development of efficient quad meshing methods has been a long-standing research topic, beginning with the development of FEM in the 1960s and 70s, which provided many algorithms. With the rapidly growing computer graphics industry, this research has come back into focus, as fast and robust

algorithms are also needed. According to Rushdi [74], unstructured quad meshing can be divided into two categories, indirect and direct approaches. Indirect approaches start with the simpler generation of tria meshes, which are subsequently transformed into quad meshes by optimization [75], simplification [76], or refinement/coarsening [77]. Direct approaches, on the other hand, produce quad meshes without detours, but are usually less robust for complex geometries. Therefore, an indirect method was used in the implementation of the optimization process. For the quad meshing, the open source 3D finite element mesh generator Gmsh [78] in version 4.11.1 is used. The meshing process can also be divided into 4 steps and is illustrated in the figure 4.13.

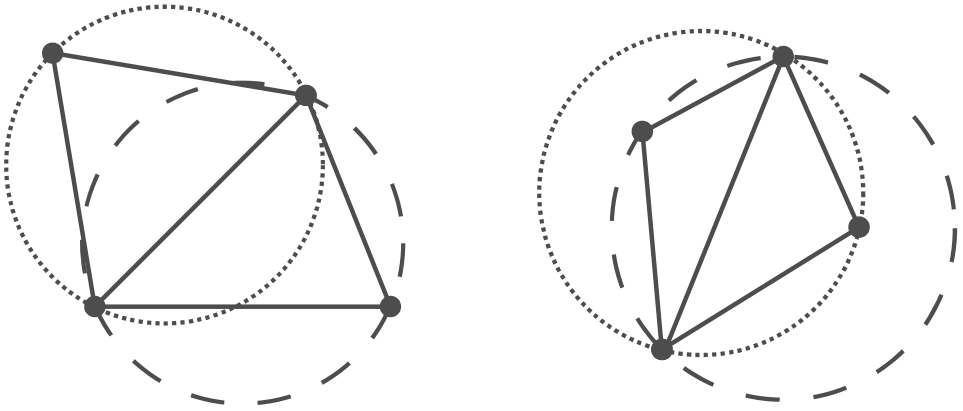
1. 2D Quad Meshing

The description of contours in Gmsh is very similar to the formulation described in section 4.1.3. Various geometric primitives, such as polygons or splines, are associated with a closed curve. Gmsh has an integrated CAD engine that internally generates a surface from the curve, which can now be used for 2D meshing. An indirect meshing method, the Frontal-Delaunay for Quads algorithm, is used to generate the 2D mesh. Regardless of the choice of meshing algorithm, Gmsh always first generates a 1D Delaunay mesh on the edges using a divide-and-conquer algorithm [79] developed by Dwyer. The frontal Delaunay algorithm combines the advancing front method with Delaunay triangulation and is described in detail by Rebay [80]. The Delaunay triangulation is a special trian-

gulation method developed by Delaunay in 1934, which was already implemented algorithmically in 1967 by Bernal and Finney [81]. The Delaunay criterion is said to be satisfied if the following conditions are met

- All Triangles produce a circumcircle which does not contain any other vertices but their own
- There are no degenerate Triangles with collinear points

The condition is illustrated by two examples in Figure 4.12. As can



(a) Delaunay (b) Non-Delaunay

Figure 4.12: Delaunay triangulation

be seen in figure 4.12.a, this criterion leads to the maximization of the smallest interior angle of a triangle and therefore tends to produce more uniform triangles and avoids sliver triangles, i.e. very thin triangles with two small interior angles, since these produce very large circumferences. To obtain a pure quad mesh, a special formulation of this method is used that preferentially produces right-angled triangles, which are easy to recombine into quad elements

[82]. The recombination of the tria elements into quad-elements is done with the so-called Blossom algorithm [83], which uses a minimum cost optimal match algorithm to generate quad meshes from triangulation. Using the full-quad option guarantees a pure quad mesh.

2. 3D Mesh

The 3D mesh generation is also done with Gmsh. The 2D elements are rotated in theta direction by specifying the number of elements. To reduce the required simulation time, only a segment of the disk is calculated, yet with periodic boundary conditions the model mathematically resemble a full blisk. The rotation angle results from the number of blades n_{blade} over $\delta\theta = \frac{2\Pi}{n_{\text{blade}}}$.

3. Nodeset Definition

Since this is an unstructured mesh, the required nodesets cannot be computed as in the case of the blade meshing, but must be created using specially implemented search functions. The following NSets are required for modelling:

- N_{RB}: NSet for setting boundary conditions
- N_{RIMF}, N_{RIMR}: NSet of leading and trailing edges
- N_{DTOP}: NSet of the top of the disc
- N_{LEFT}, N_{RIGHT}: NSets of the edges for periodic boundary conditions.

It is important that the nodes of the dependent edge are not contained in other NSets. The identification of the nodes is made possible by the fact that the first and the second point of the slice contour are always identical to the sheets LE and TE. Gmsh allows filtering of nodes based on geometric curves and surfaces. The identification of the N_{RB} set uses the axial symmetry of the number of edges to find the curve of the DLE and DTE, and then searches for two rows of nodes at a given radial height. The N_{RIMF} and N_{RIMR} NSets can be identified using the following two simple searches:

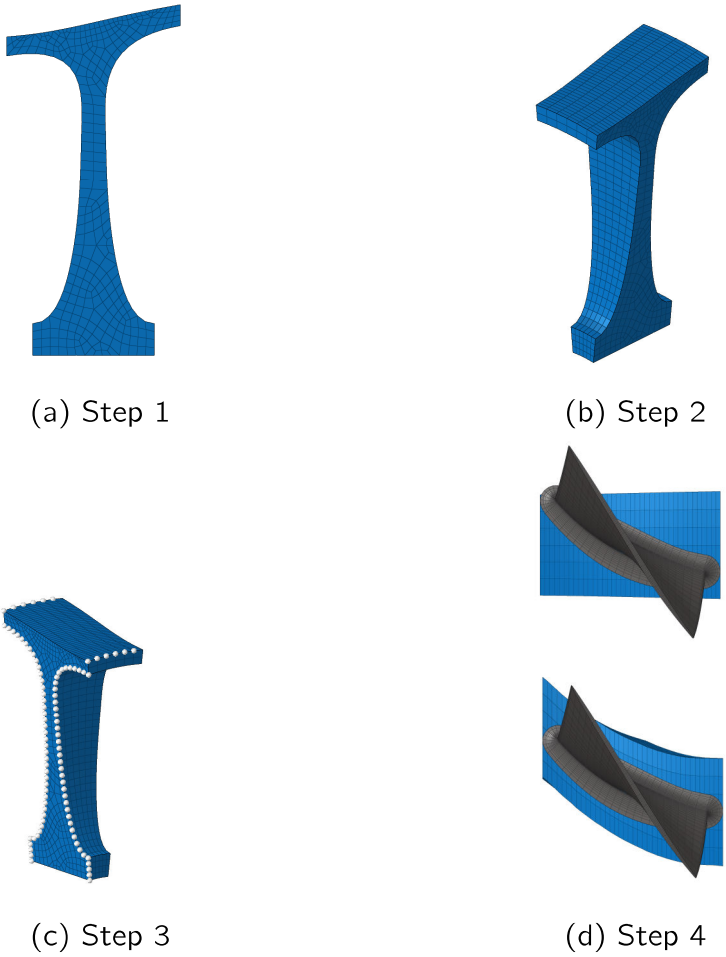
1. Find the 3 edges containing the blade LE or TE point
2. Find the edge whose vertices have constant radius

The NSET of the top of the disk N_{DTOP} can then be found using the surface containing the NSets N_{RIMF} and N_{RIMR}. The surface containing the original contour line has index 1 by default, so the opposite flank over nodes with surface 1 have identical x and r values and of these the maximum θ value, so N_{LEFT} and N_{RIGHT} are also determined.

4. Theta Distortion

The mesh is now a sector of the compressor disk with flanks of constant θ values. The rotor blade, on the other hand, has a more or less curved path and can therefore extend beyond the disk sector. If the calculation is performed considering fillets, this happens in almost all cases. The connection of the incompatible blade-disk mesh generated by Multi-Point-Constraint (MPC) would therefore

fail, which is why the disk must be distorted accordingly. Figure 4.13.d illustrates the method of theta distortion. The method used here generates a r, θ-spline based on the blade chord, which forms a surface with the z-axis to distort the disk.



(a) Step 1

(b) Step 2

(c) Step 3

(d) Step 4

Figure 4.13: Disk Meshing Process

5 Optimization

Optimization is the process of finding the objectively best solution according to specified criteria. The optimization problem always consists of three components: the objective, the design variables, and the constraints. Depending on the formulation of the optimization algorithm, one or more objective functions are possible, usually formulated as a minimization problem. In the case of multiple functions, there is a trade-off in the form of a Pareto front, which is similar to generative design, since a solution must now be selected from the solution space. The mathematical description of optimization methods goes back to the work of Newton, Lagrange, and Gauss and is summarized by Hancock [84]. Optimization methods could develop their real potential only with the use of powerful computers, which was demonstrated as early as 1984 by Karmarkar [85], with his linear programming algorithm. Since then, the theory has been continuously extended and is available, partly in freely accessible, user-friendly libraries. Regardless which optimization method is used, all formulations assume fully automated processes. In the following, the formulation of the optimization process for the design of disk geometries is described.

5.1 Aerodynamic Blade Design

The disk design should be based on a realistic engine configuration and therefore aerodynamically reasonable blade geometries. A full-scale design using 3D CFD is not feasible due to the computational time involved. To estimate realistic geometries, two prediction methods developed at the DLR Institute of Propulsion Technology are used.

Rubber Engine

Realistic compressor configurations are generated using the Rubber Engine tool described and validated by Häßy [86]. With this tool, the component configuration and overall dimensions of an engine can be estimated at the concept stage. This is made possible by a hybrid surrogate-based model, which is composed of the results of performance calculations and surrogate models from different disciplines. The conceptual structure of the method is shown in Figure 5.1. A parameter study on total engine weight and length, also

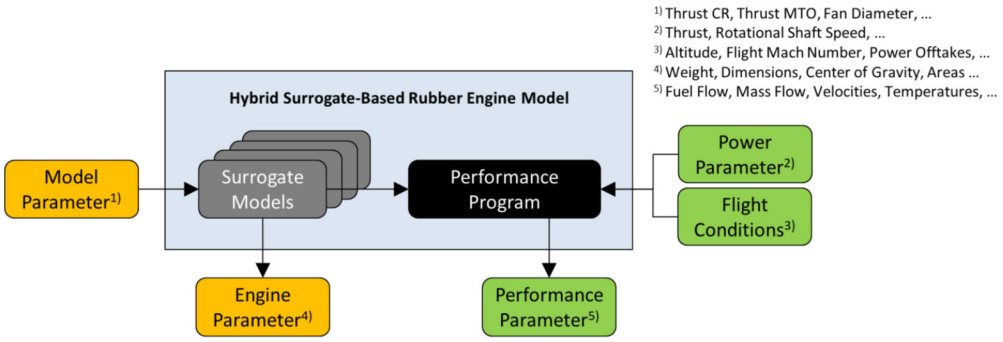


Figure 5.1: Rubber engine process architecture [87]

conducted by Häßy [87], showed a large influence on the engine dimensions of the parameters bypass ratio (BPR), thrust requirement at cruise (FN_{CR}) and thrust ratio takeoff to cruise (FN_{Ratio}). If BPR and FN_{Ratio} are kept constant, the weight of the engine rises sharply with increasing cruise thrust requirement due to increased mass flows and thus larger components. On the other hand, if BPR is increased while the thrust requirement remains constant, the fan weight increases, but the core components become smaller. Varying these parameters within the specified limits thus covers a very wide range of possible compressor configurations.

ACDC

After the conceptual design of the compressor with the help of the rubber engine tool, the number of stages, the flow path and the GT-lab station definitions (LE and TE) of the blades are available. This is where the Advanced Compressor Design Code (ACDC), also developed at the Institute of Propulsion Technology by Schnös, comes in [88]. The method uses a procedure similar to this work to create a surrogate model for aerodynamically favorable 2D blade profiles. For this purpose, a database of optimized 2D airfoils was created using the airfoil parameters described in 4.1.2 and [89]. The blade profiles were evaluated based on their loss characteristics using the MISES code [90]. A special formulation of co-rigging was then used to train the surrogate model, which reduces the number of evaluations required [91]. The airfoil geometries are sized using aerodynamic parameters such as downstream angle and inlet Mach number

according to the results of the performance and rubber engine calculations. The conversion to 3D geometry is then performed using the BladeGenerator, where a blade is constructed from 5 2D profiles per blade. The process is the same as shown in [91] and is illustrated in figure 5.2. The results of the profile dimensioning were compared

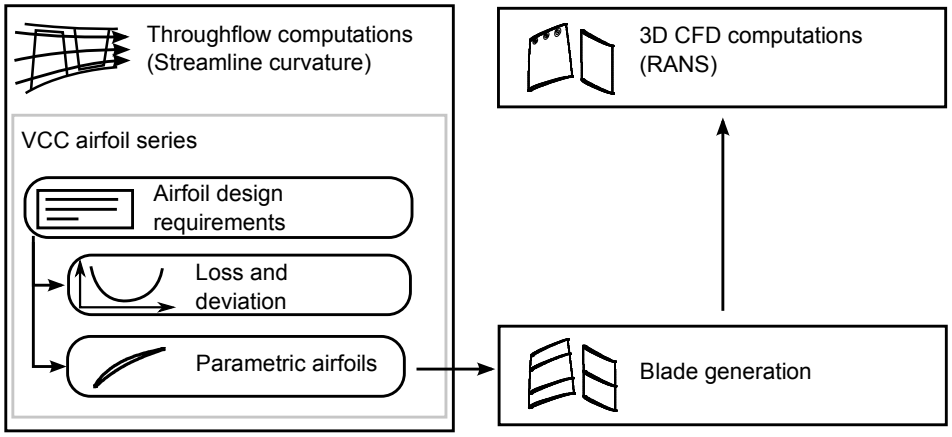


Figure 5.2: ACDC process chart [88]

with extensive 3D-CFD calculations of the DLR code TRACE [92]. According to Schnös [88], the profiles designed with ACDC show slightly lower losses, but higher downstream angles. For the airfoil area, which is directly linked to the mass of the blade, the deviations are negligible as shown in figure 5.3 at around 0.18% with the best method, which makes the method suitable for blade dimension estimation used for disk design.

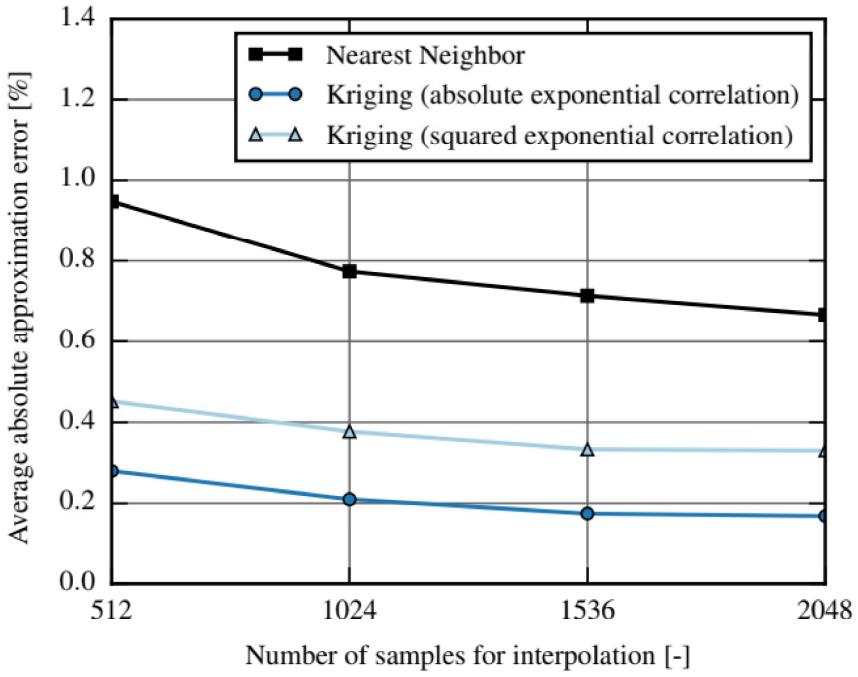


Figure 5.3: Predicted Profile approximation accuracy [91]

5.2 FE Solution

The commercial solver PERMAS developed by INTES GmbH and the open source solver CalculiX were implemented as FE solvers in GTlab. For the calculation, the blade and disk mesh, whose generation has already been described in detail in section 4.2.2, are connected by linear multipoint constraints (MPC). The displacements of the guiding nodes (u_g) are applied to the dependent nodes (u_d) with a distance-dependent weighting factor A, given by

$$u_d = \sum_{i=1}^{n} A_i u_{g,i} \tag{5.1}$$

With PERMAS, this is done through the ISURFACE function, where node sets can be directly connected. The distance over which nodes are connected applies in all three spatial directions and can be controlled by a tolerance parameter. CalculiX [93] has the TIE function, which computes a surface from a given nodeset and connects it to the second given NSet, but the tolerance parameter controls only the normal direction of the previously determined surface. Blade and disk meshes must therefore overlap in the surface normal direction, making the theta distortion shown in Figure 4.13.d mandatory. The coupling gives very good results for the resulting displacement, but the stress distribution in the MPC domain is often not trustworthy. However, since for the case of static stresses under rotational load, the maximum disk stresses are expected at the inner radius of the disk, the nodes involved in the MPC can easily be ignored in the stress evaluation. For a faster calculation time, instead of a complete disk, only a disk segment is calculated, whose size in circumferential direction is defined by the number of blades. For this purpose, all disk nodes are transformed into cylindrical coordinates according to

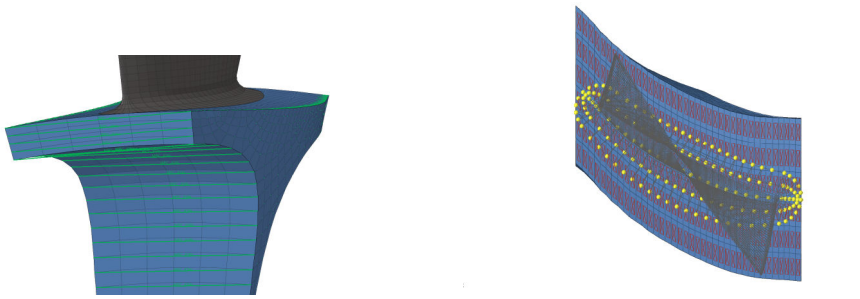
$$r = \sqrt{y^2 + z^2} \tag{5.2}$$

$$\theta = \text{atan2}(z, y) \tag{5.3}$$

$$z = x \tag{5.4}$$

The atan2 function, first implemented in Fortran in 1961 for exactly this application, always returns the correct θ value regardless of the y, z sign. With the application of periodic boundary condi-

tions (PBC) at the disk sector flanks, the model now mathematically corresponds to that of a full disk. The blade-disk connection and the periodic boundary conditions are shown in figure 5.4. Another



(a) Periodic boundaries (b) Blade disk interface

Figure 5.4: Multi point constraint definitions

advantage of the nodal transformation to cylindrical coordinates is the simplified application of displacement constraints. A solid disk is self-supporting in the radial direction and this is also true for the idealization with PBC's, which means that only the motion in the circumferential and axial direction has to be restricted. The position of the boundary conditions has been implemented as a variable parameter via the node set N_{RB} as described in section 4.2.2. The main requirement for the FE solver in the optimization is the minimum computation time while maintaining the required accuracy. The measure of accuracy is the validity of the stress and displacement results used to constrain the optimization. From the analytical disk calculation in section 2.1.2 it is already known that the dimensioning is defined by both σ_r and σ_θ. Since an isotropic material model is used, the multiaxial stress state can be transferred to a fictitious uniaxial state by applying appropriate hypotheses. For ductile

materials under static loading, Mises' shape change hypothesis has proven to be very reliable. The six stress components are converted according to

$$\sigma_{\text{Mises}} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2 - \sigma_x\sigma_y - \sigma_x\sigma_z - \sigma_y\sigma_z + 3(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{xz}^2)} \tag{5.5}$$

into an equivalent stress. The result can be directly compared with the material-dependent yield strength to obtain the material utilization. For all following calculations, the titanium alloy Ti-6Al-4V (TI64) is used with the material properties listed in the table 5.1. For the discretization, the resolution of the FE mesh has a direct

Parameter	**Description**	**Value**	**Unit**
E	Modulus of Elasticity	113.8	GPa
ρ	Density	4.43	$\frac{\text{g}}{\text{cm}^3}$
ν	Poisson's Ratio	0.342	-
$R_{\text{p0,2}}$	Tensile yield Strength	880.0	MPa

Table 5.1: TI64 Material Properties [94]

influence on the solution time; three levels of detail are considered for the blade: low with 240 elements, high with 8800 elements and idealization only as mass point. For the disk a low resolution with around 400 elements and a high resolution with around 6000 elements is evaluated. Since the disk uses an unstructured mesh method the amount of used elements differs for different geometries, so the given values are only approximations. As evaluation criteria, only the effects on the calculated disk stresses and the maximum displacement values in the disk are considered. The results of

the mesh study are shown in figure 5.5. The influence of detailed

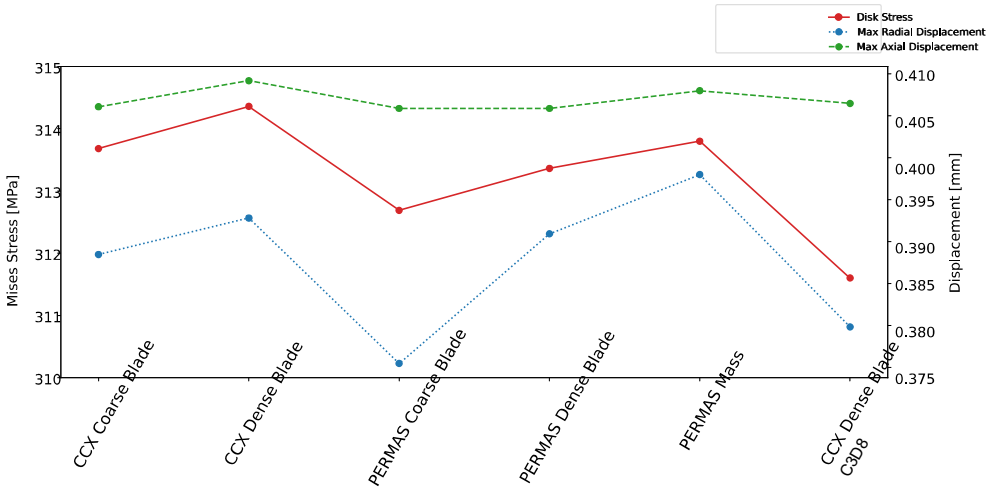
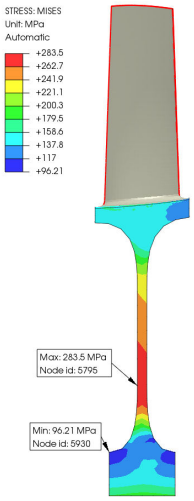


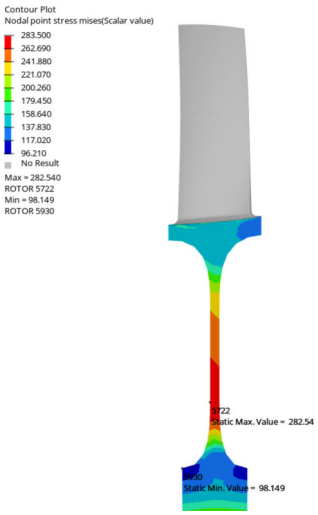
Figure 5.5: Comparing different solvers and mesh resolutions

meshes when using CalculiX is negligible and reaches almost identical values compared to PERMAS with fine meshes. An idealization of the blade as a mass point leads to almost no acceleration. The calculation with CalculiX with coarse blade and disk meshing proves to be a good compromise between accuracy and speed. The use of HEX8 elements is allowed when advanced strain functions, the so-called bubble modes, are taken into account. This formulation avoids Possoin ratio locking, which leads to excessively stiff element behavior [95], as seen by the low stress and displacement values when using the standard eight-node hexahedron element C3D8 in CalculiX. PERMAS uses the advanced elements by default, in CalculiX the element type C3D8I must be explicitly selected. Another option to accelerate the calculation is the level of detail of the analysis. Both FE solvers have geometric linear and nonlinear static

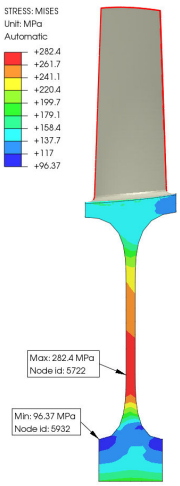
analysis methods. In linear static analysis, only one solution of the structural response with respect to the initial orientation is calculated, which leads to errors, especially for large displacements. The geometrically non-linear method calculates the solution iteratively, reorienting the load, in this case the centrifugal field, on the model after each step until the calculation falls below a convergence limit. The results of the different analysis methods as well as the comparison between PERMAS and CalculiX are shown in Figure 5.6. A comparison of the results shows small deviations of 0.75% for PERMAS and 0.39% for CalculiX between the geometrically linear and non-linear analyses, yet a time saving with a factor of around 10. The negligible effects of the linear analysis can be explained by the very small displacements in the disk. When considering the rotor blades, the differences are much more significant with 14.83% for PERMAS, since large displacements occur there. In summary, the analyses show that the calculation of the stress and displacement distribution within the disk using the simple blade option in combination with a geometric linear static analysis provides sufficient accuracy at very fast computation times. CalculiX was chosen as solver because its open source licence allows it to be used in parallel on multiple computers and provides more easily repeatable results for the simple blade option.



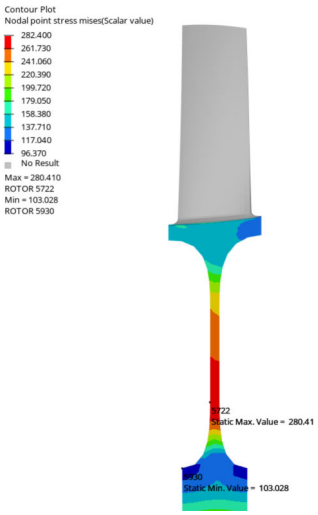
(a) CalculiX Linear



(b) PERMAS Linear



(c) CalculiX non-linear



(d) PERMAS non-linear

Figure 5.6: Geometric linear and non-linear analysis with PERMAS and CalculiX

5.3 Constraints and Aim Functions

The results of the FE solution have to be transformed into an optimization problem. By default, the objective function is the minimization of the weight due to the lightweight requirements in aerospace. To control the optimizer, it is sufficient to specify the volume, since the density is a constant parameter during the optimization. The formulation of the constraints h has a great influence on the disk dimensioning. Most available optimizers support the formulation of nonlinear inequality constraints of the form

$$h_i(x) \leq 0 \tag{5.6}$$

which are sometimes referred to as nonlinear programming problems. Using equality constraints

$$h_i(x) = 0 \tag{5.7}$$

is only supported by advanced optimization algorithms. In principle, any equality constraint could be formulated by two inequality constraints $h_i(x) \leq 0$ and $h_i(x) \geq 0$. Yet in practice, such formulations usually lead to convergence problems [96]. The constraints of the present problem require both formulations. The disks are among the critical components of an aircraft engine, since their failure can lead to the loss of the aircraft, and are listed in the Failure Mode Effects Analysis (FMEA) performed for aerospace components in the component group A1 [97], whose failure must be absolutely excluded. The design load case under consideration is the disk burst

at 110 percent of the maximum rotational speed (Redline), which, to the best of the authors' knowledge, represents the critical load case. The process architecture has been deliberately constructed to enable the integration of more sophisticated failure criteria with minimal modifications, should this become necessary. The disk burst criterion can be calculated as a function of Mises stress by

$$S = \frac{R_{p0,2}}{\sigma_{\text{Mises}}} \tag{5.8}$$

multiplied by the safety factor S. An minimum safety factor for safety critical components is 2, which is increased to 2.25 in order to incorporate eventual unknown load requirements. This leads, according to (5.8) and table 5.1, to a σ_{Mises} limit of 391.11MPa. The method to be developed is of statistical nature and will most likely predict solutions distributed around the training data. To incorporate this behavior at an early stage, the statistical factor $S_{\text{prob}} = 1.1$ is introduced, reducing the stress limit further to $\sigma_{\text{MisesMax}} = 352\text{MPa}$. The maximum stress of the disk and the disk weight are loosely inverse proportional, the stress restriction can be formulated as an inequality constraint

$$h_1(x) \leq \sigma_{\text{Mises}} - 352 \tag{5.9}$$

In addition to the failure criterion, there are requirements for the deformation behavior. To prevent the rotor blades from rubbing against the outer wall, the disk must expand as uniformly as possible. In addition, the centrifugal load must not cause any axial movement.

Equal constraints are used to formulate these requirements. In the Cartesian coordinate system, they are expressed as follows:

$$h_2(x) = \delta_{rTE} - \delta_{rLE} = 0 \tag{5.10}$$

$$h_3(x) = \delta_{zTE} = 0 \tag{5.11}$$

5.4 Process Architecture

The optimization process is controlled by a global Python script that can be executed through the GTlab Python interface. The Python script can optionally be run in GTlab batch mode, which allows easy data generation on mainframes. Figure 5.7 shows the process flowchart. The process consists of the following steps. First, the

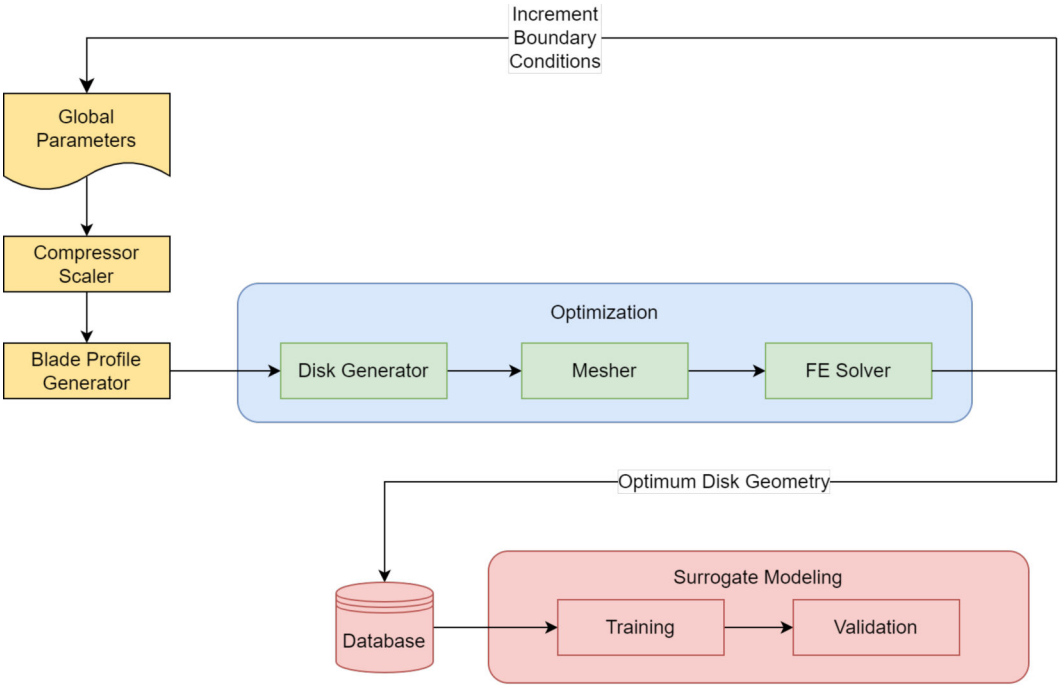


Figure 5.7: Global process description

Rubber Engine uses the thrust and bypass specifications to generate a compressor component with the optimum number of stages and the flow path (1). Then the 2D profiles of the rotor blades are dimensioned by ACDC (2). The 5 profiles per blade are converted into a 3D geometry by the BladeGenerator (3). According to the disk parameterization, an initial disk can be generated from blade LE and TE (4) with minimal mass (the reason for this is explained in the next section) and the ring position parameters located under blade center of gravity. This initial disk can be meshed and connected to the meshed blade (5) to form an FE model. The available FE solvers solve the displacements and stresses resulting from the loads (6). The results are stored in a member database (7). Steps 4-7 are now repeated with variation of the blade geometry parameters until at least a locally mass-optimal blade is found within the constraints. The convergence criteria for the optimization process are defined by two relative tolerances: one on the function value (FTol) and the other on the optimization parameter values (XTol). The process terminates upon meeting either of these tolerances, where a relative tolerance of 0.0001 has been established as a satisfactory balance, providing an efficient trade-off between optimization duration and the quality of the resultant optima. The result is stored in a global database of optimized members (8) and the python script continuous with a new set of bypass and thrust values. The loop is repeated until a sufficient amount of data is available.

5.5 Optimization Algorithms

The optimization is carried out with the open source package NLopt [96]. NLOpt has a C++ API, which makes it easy to integrate into the GTlab framework. The library contains several algorithms, but both equality and inequality constraints must be satisfied, which limits the choice. In test runs, the COBYLA algorithm showed robust and very fast solutions, yet struggles to find the global optimum reliably. Given the substantial volume of data anticipated to be produced, the velocity of the optimization process is a parameter of considerable significance, which may take precedence over the precision of the optimization outcomes. To assess the optimizer's efficacy in locating the global optimum, a comprehensive examination using 100 optimization trials for the same problem was executed, each with varying starting solutions and initial step sizes. The corresponding results are presented in Figure 5.8. This investigation reveals that starting from a lightweight solution in conjunction with smaller initial step sizes tends to yield superior outcomes. With good starting conditions, the COBYLA demonstrates a robust ability to identify the global optimum in approximately 90% of the trials conducted. The Optuna algorithm [98] used in the next chapter for hyperparameter optimization was also tested for the design optimization, yet could not produce satisfactory results in test runs, mainly due to the missing implementation of inequality constraints. The NLOpt COBYLA implementation is based on Powell's work described in [99] and reviewed in [100]. The formulation of the equality constraints is solved via the pairwise implementation of in-

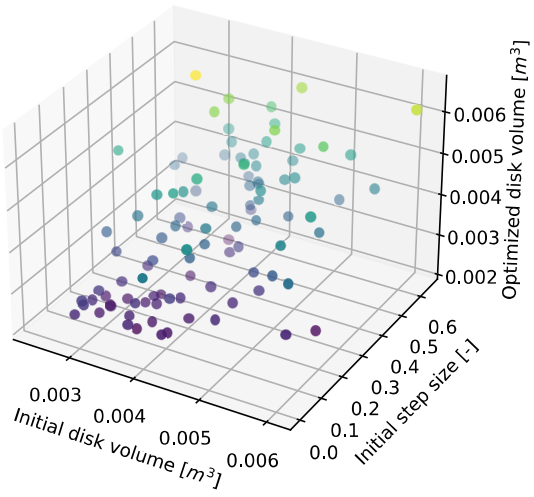


Figure 5.8: Results for 100 optimizations with varying starting conditions

equality constraints, which in this case does not cause convergence problems for the algorithm. The main challenge for the optimizer is to find the global optimum instead of just a local one. COBYLA constructs a sequence of solutions based on linear approximations of the objective and constrain functions. This sequence inherently has local characteristics, which the optimizer tries to minimize. A general solution would be the excessive sampling of the parameter space, yet this quickly leads to high computation times. To achieve fast solutions with a high probability to find the global optimum, the best strategy at hand is to choose an already very good initial solution to initialize the parameter space in the neighborhood of the global optimum. Based on a sampling of the overall parameter space, a set of starting parameter was chosen. The data analysis in

the next chapter will determine quality of the generated data.

6 Data Driven Design

Over time different data science methods and standards were developed on how data should be processed, with the two most prominent being CRISP-DM and OSEMN [101]. While CRISP-DM aims mostly for web data scraping processes and large scale business applications, the latter is formulated more general and therefore suites the application of a predictive design prototype better. First introduced by Mason and Wiggins [102] the OSEMN method divides the data life cycle into the five name giving sections displayed in figure 6.1. The underscored letters form the acronym OSEMN which

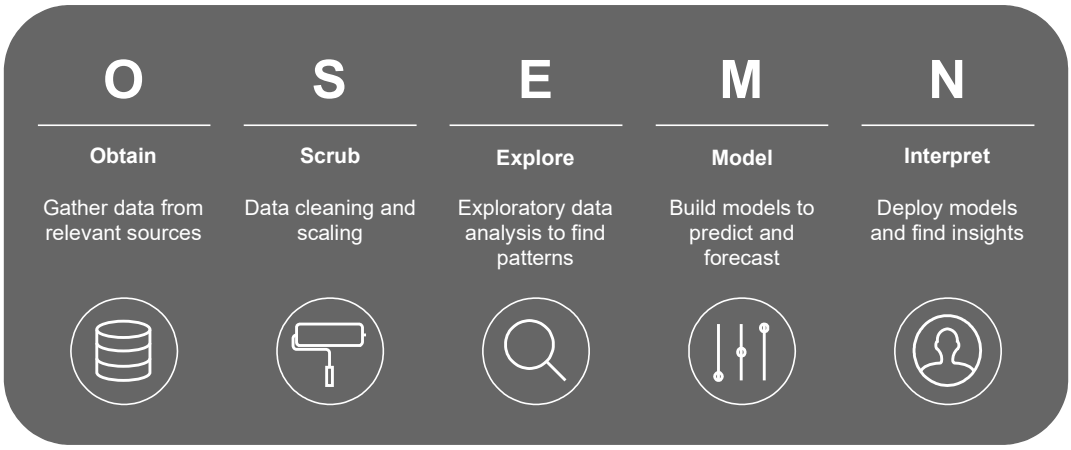


Figure 6.1: OSEMN data cycle method (redrawn from [101])

is also a distant homophone to the word "awesome". This chap-

ter explores the OSEMN approach as a guiding framework for the training process of a machine learning model to predict optimal disk geometries.

6.1 Obtain

The process of data extraction has already been described in detail in the section 5 from the structural mechanical point of view. This section will focus on the information technical aspects. Starting from an eleven dimensional input parameter space, an optimization is carried out for a selected parameter combination. The generated results, both of the individual iterations and the determined weight optimal disk geometry for the specific input data selection, are stored in csv databases. The schematic data flow of the generation process is shown in figure 6.2. The overlying rectangles stand for individual optimizations with different blade geometries and loads, where one rectangle is an iteration with different disk geometries. When the optimization is finished the optimal result is stored in the global database while every intermediate step is stored in a numbered optimization database. The input parameter space is covered as evenly as possible within the defined limits. The even distribution of the input parameters is realized by using the python library "random". The function 'random seed' allows the generation of random parameters for each run within the limits given in the table 6.2, which leads to a uniform distribution if the sample number is high enough without having to specify discrete step sizes.

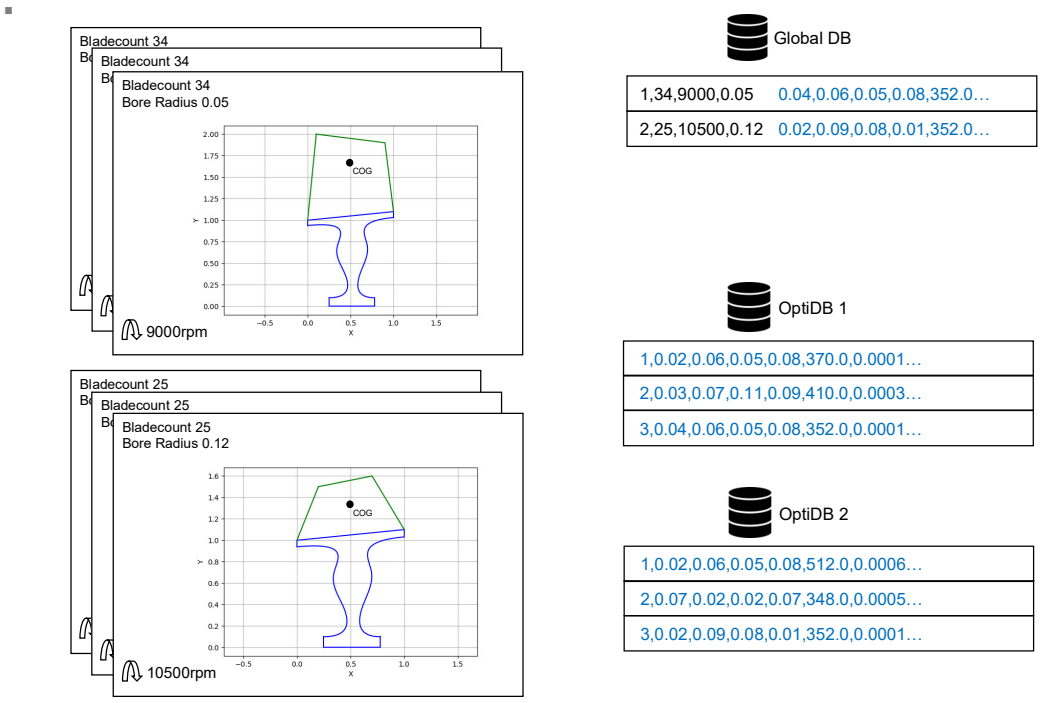


Figure 6.2: Schematic data acquisition process

The user tends to select rounded values as input, for example 0.1 will be used more often than 0.08 for r_{bore}. To account for this behavior and increase the prediction accuracy at these query points, additional training data is generated at certain discrete input parameter values, which can be seen in figure 6.5. The selection of the free parameters and their ranges go back to the sensitivity analyses performed in chapter 5. The parameter ranges represent many possible compressor blades. Only very small engines and turbine power simulators (TPS) or very large rotor blades for fan stages are currently not covered. Through parallelization and using several computers, a total of 52956 optimizations with a sum of 9498499 FE calculations were executed. Depending on the complexity of the

problem the optimizer required different amounts of iterations, averaging at 179 to meet the convergence criteria already described in 5.5. The wide parameter space increases the complexity of the optimization problem. High speed load in combination with small available disk space does not allow the optimizer to find physically valid solutions, while the opposite leads to an under-definition of the problem and therefore several valid but not necessarily optimal results. Yet, the advantages of a broad database outweigh the associated disadvantages, as targeted data cleaning can eliminate much of the aforementioned problems. The measures for this cleaning process are described in the next section.

6.2 Scrub

The next step in the data life cycle is the data cleaning. During the optimization process some errors occur which lead to the invalidity of data points. Depending on the type of error, these are marked with a corresponding error code and the calculated stresses and displacements are set to the value 9999.99. The used error codes have the meaning given in table 6.1. The column "use" specifies the handling of the different error types. Error code 0 to 3 and 6 are critical errors, whose data points must be removed from the database, since the is corrupted due to process failures. Meshing can occur due to complex geometries and FE Solver errors due to corrupt elements. Process errors are infrequent, occurring in approximately 1 out of every 1000 cases, as considerable care has been devoted to

Error code	Description	Use
0	Blade meshing failed	remove
1	Disk meshing failed	remove
2	FE Solver failed	remove
3	NLOpt failure	remove
4	NLOpt FTol reached	filter
5	NLOpt XTol reached	filter
6	NLOpt max evaluations reached	remove

Table 6.1: Meaning and use of different error codes

ensuring the robustness of all tools. Error codes 4 or 5 can be used as filters to improve the data quality and achieve better prediction accuracies, since they represent different convergence criteria. The effects of different filters will be discussed in more detail in section 6.3. The implementation of removing and filtering is made possible by the functionalities of the python library "pandas". The csv-files can be read directly into a Pandas dataframe using the built-in read CSV function. Dataframes can be understood as dynamic tables, which can be used to perform various data operations. After outlier removal and filtering, the data needs to be normalized. Due to the different dimensions and units of the individual parameters, for example the rotational speed has the order of magnitude 10E5 while r_{rb} is dimensionless in the order of magnitude 10E-1, the formulation of suitable substitute models is made difficult. The data can be normalized in different ways, established methods are Min-Max and MaxAbs Scaling, which scale the individual parameters according to their maximum value either to the range [0,1] or [-1,1] and thus remove dimension effects from the dataset. MaxAbs Scaling mainly

has advantages in the robustness against outliers and when used with sparse data, which neither is applicable to the data at hand. The chosen method to scale the data is therefore the straight forward and good interpretable Min-Max scaling with:

$$X_{std} = (X - X_{min})/(X_{max} - X_{min}) \tag{6.1}$$

$$X_{scaled} = X_{std} * (b - a) + a \tag{6.2}$$

a and b are the desired lower and upper bounds of the scaled data, in the case of Min-Max scaling with $a = 0$ and $b = 1$. To inverse transform the scaled data the given equation must be rearranged to X, which results for the simplified case of Min-Max scaling to:

$$X = X_{std} * (X_{max} - X_{min}) + X_{min} \tag{6.3}$$

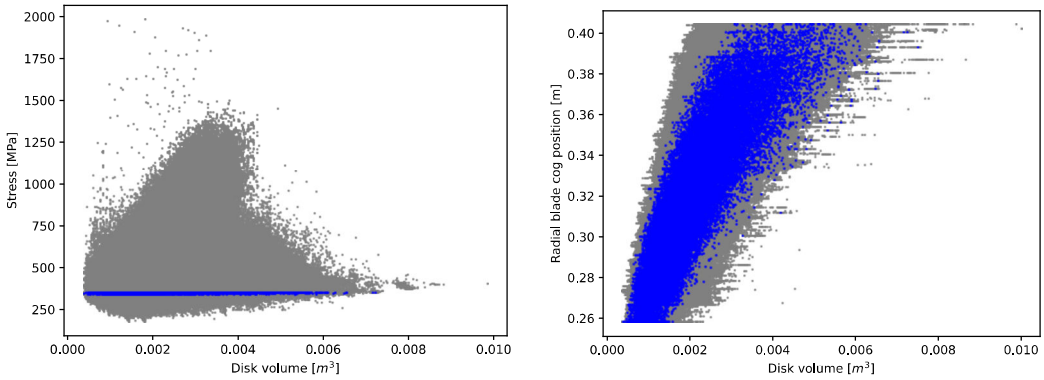
The maximum and minimum values must be stored for each parameter, for one to be able to undo the scaling after model building and to normalize further data sets using the same scaling rule. The scaling parameters can be exported using the joblib libraries dump function and are documented in table 6.2 Scaling has an influence on the quality assessment criteria of the trained model [103], since absolute and relative errors depend on the scale of the data. When there is significant difference in the magnitudes of the data dimensions, the impact becomes more pronounced. In this context, the revolutions exhibit dimensional values that are roughly 100,000 times greater than those of the axial length, underscoring the critical importance of appropriate data scaling.

	m_{blades}	cog_x	cog_r	L_x
Min	1.4505	0.3983	0.2584	0.0383
Max	5.3638	0.4646	0.4037	0.0649
	δ_r	$bore_r$	rb_r	rpm
Min	0.0029	0.0100	0.0100	6052.0
Max	0.0068	0.1897	0.9890	14987
	w_0	w_1	w_2	w_{hub}
Min	0.0200	0.0200	0.0200	0.3000
Max	0.1000	0.2996	0.2992	0.9900
	p_0	rim_{le}	rim_{te}	$bore_h$
Min	0.4296	0.0400	0.0400	0.1000
Max	0.5012	0.0789	0.0795	0.4869

Table 6.2: Minimal and maximum values for each parameter

6.3 Explore

The exploratory data analysis is the most important step in the data life cycle. Understanding the data to utilize different enhancement techniques can greatly increase the performance of the surrogate model. Due to the local optimization process, the exploration shows a high amount of noise in the data. Feature selection and data augmentation can be used to increase the model performance on the data side. The analysis is done on cleaned data but before the normalization in order to facilitate interpretability. The optimization process in general serves as a tool for controlled data generation. While searching for the global optimum the optimizer generates data more likely in the region of interest. The scatter plots of figure 6.3 show the influence of the optimizer, with the density of suboptimal data points (grey) being higher around the optimum solutions



(a) Maximum stress vs. disk volume (b) COG radius vs. disk volume

Figure 6.3: Distribution of optimal and suboptimal data points

(blue) and more scarce when the distance to the optimum weight increases. While stress and disk volume show a loosely inverse proportional connection, the blade radial center of gravity influences the disk weight nearly linear. To get a better understanding of the parameter relationships and their influence on the output parameters in the optimum database, a correlation heatmap can be used. The correlation coefficients indicate what correlation a pair of parameters have and are in the range of -1 to 1, where:

- 1 indicates a perfect positive correlation
- -1 indicates a perfect negative correlation
- 0 indicates no correlation

For the data analysis of large datasets, the Seaborn python library is used to plot the correlation heatmap in figure 6.4 directly from a pandas dataframe. Since the disk is rotational loaded, the mass and radial coordinate of the blade cog have a high influence on

Figure 6.4: Correlation heatmap of all parameters

the optimum disk mass, yet the influence of the rotational speed shows very low correlation with the disk mass at a value of only 0.04. Since the disk itself is rotationally loaded, it has to carry its own weight. At higher rotational speeds the disk still tends to be as lightweight as possible to reduce the maximum stress on the inner diameter. For the targets, the ring width w_1 to w_{hub} and the axial position of the disk center have the most influence on the disk weight. The strongest positive correlation is found between the radial center of gravity of the blade and the disk weight. Due to the centrifugal loading higher radial cog positions mean higher forces, which result in a heavier disk. The strongest negative correlation can be found between the displacement results in x- and r-direction. This is consistent, due to the constraints applied to the disk which prevent lateral rigid body motion of the disk. The disk can only bend or tilt, so the displacement values are linked together, but cannot be derived from each other. To get a better understanding on the parameter distribution, a histogram is plotted for all parameters, which is shown in figure 6.5. The first three plots (numbering from top left to bottom right) clearly show the concentration on the region of interest, where the majority of the data points are at and around the specified restriction values for the allowed stress and displacements. The following eight input parameters are evenly distributed, except for the additional data concentration on even and rounded discrete data values. The eight target parameters defining the disk geometry show random fluctuations and irregularities, which in data science is called noisy data.

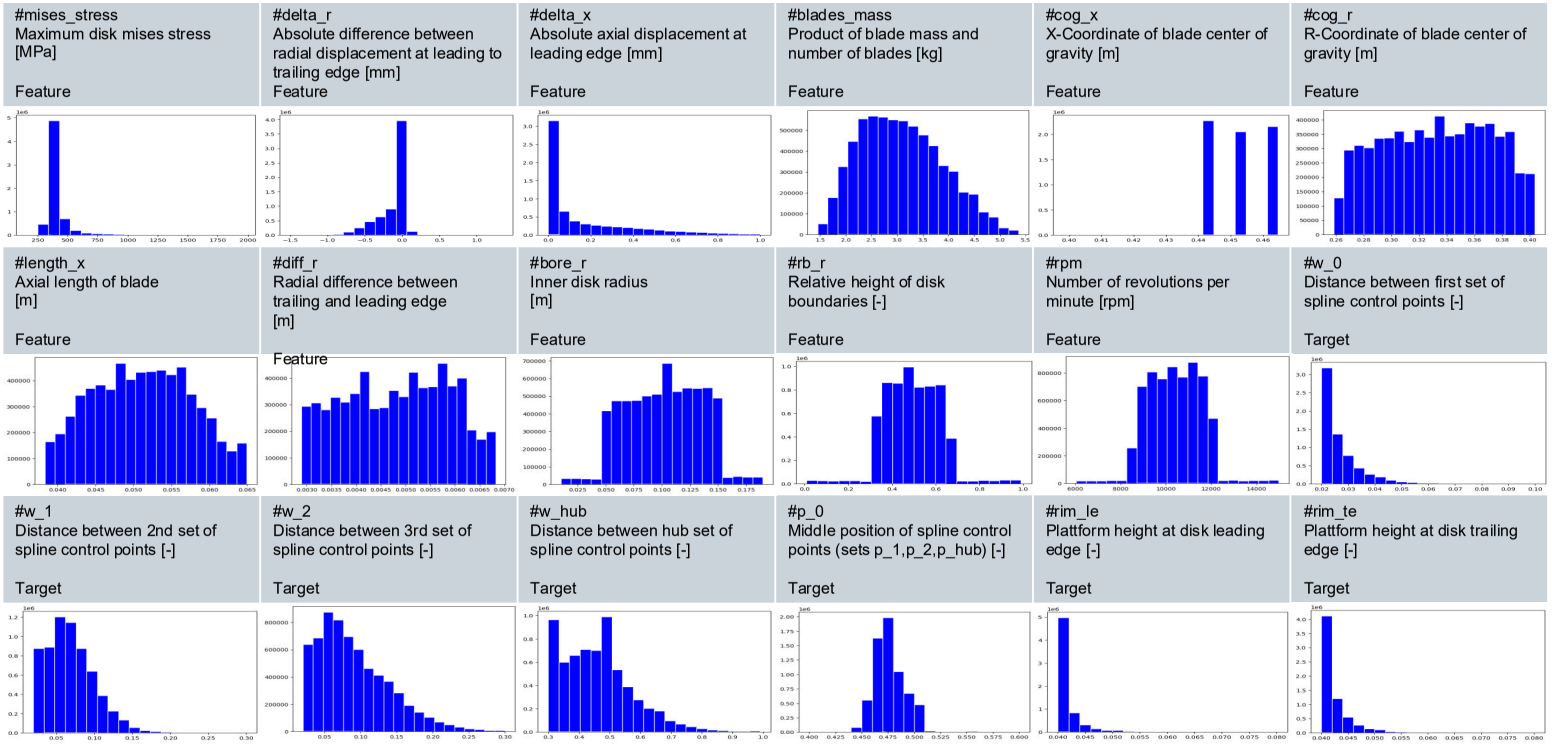


Figure 6.5: Range and distribution overview for all parameters. All plots have the absolute occurancies on the y-axis and their specific values as stated above the diagram as x-axis

6.4 Model

The exploratory data analysis showed high noise in the dataset, which calls for special model architectures. A simple experiment with an out-of-the-box model can quickly confirm this assumption. For the model generation the python library Keras [104] and Tensorflow [105] are used, with the Scikit-learn [106] library for pre- and post-processing tasks. Using a DNN with the standard layer architecture shown in figure 6.6 on the optimization database with a train/test split of 80% yields and 20 epochs yields unsatisfactory results. The loss value is reduced from 1.21 to 0.62 while the vali-

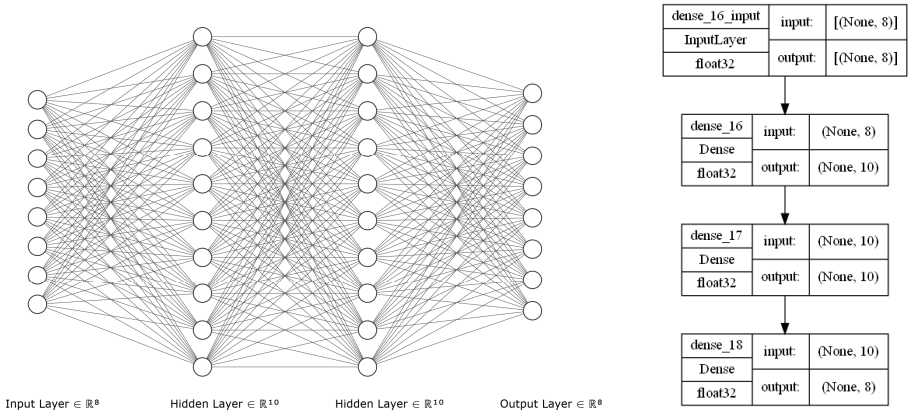


Figure 6.6: Standard neural network architecture

dation loss stays nearly unchanged at 0.98 over the training period. This behavior is a clear indicator for model overfitting and can be countered with a range of different measures. The most promising models for noisy data are: Random Forests, Nearest Neighbors Methods and Neural Networks with measures against overfitting. These models have various fine-tuning parameters to improve their

prediction accuracy. To determine the optimal parameters, a hyperparameter optimization is carried out using the python library Optuna [98]. Optuna automates the process for finding the best hyperparameters and offers various optimization algorithms and visualization options. Unlike conventional optimizers such as NLOpt from the 5 section, Optuna is suitable for optimizing discrete values such as integers or string lists. The following subsections describe the specificities of the models applied and the optimization of their control parameters.

6.4.1 Random Forest

Generally ensemble methods like Random Forests perform well on high noise datasets. Also called decision trees, random forests build an ensemble of decision trees and train each of them only on a subset of the data. During the prediction each decision tree casts a vote with the final output being the average value for the regression problem. The averaging process is the main reason why random forests perform well on high noise datasets since the effect of outliers and noise are mitigated. The Optuna hyperparameter optimization is used to find the optimal parameter set for the following control parameters of the Scikit-learn random forest regressor algorithm: The number of trees used can be specified and is usually between 50 and 500. Generally a higher number improves the models' performance but increases the computational cost. How the data is split can be controlled by the number of features considered at a node and the criterion used to measure the quality of the data split. The fea-

Param	Description	Range
n_est	Tree count	50-500
max_d	Max depth	1-35
s_split	Min samples to split	0-1
s_leaf	Min samples at leaf	0-0.5
feat	Max features	"sqrt", "log2", 0.2-0.8
crit	Split quality	"mse", "poisson","friedmann"

Table 6.3: Hyperparameter search space for the random forest regressor

ture threshold is needed to divide the data into homogeneous sets, which enables more accurate predictions. For regression problems the most common criteria are the mean square error or the mean absolute error, similar to neural networks. The result of the optimization is shown in figure 6.7. The optimal split criterion is MSE.

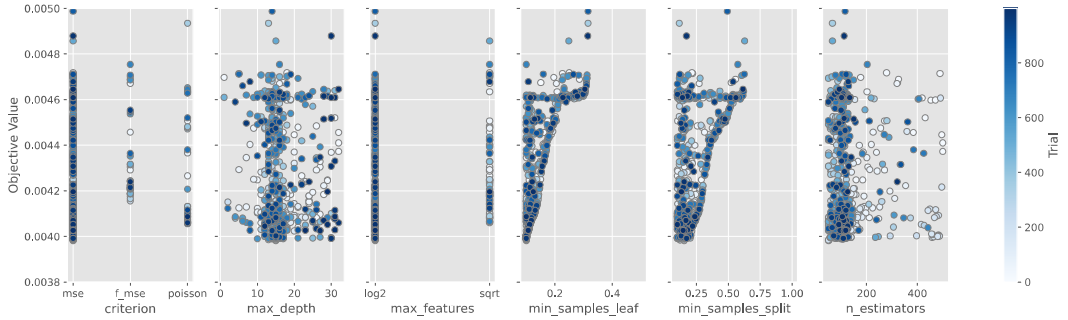


Figure 6.7: Trial results of the random forest parameters

The max depth has not much influence on the objective function and is set to 15. For the max features selection log2 is used with minimal samples to split at 0.15 and minimal samples at each leaf at 0.1. The parameter with the highest influence is the amount of estimators, which showed an optimum at 51.

6.4.2 Nearest Neighbor Search

Nearest Neighbors methods rely on local information and perform well on datasets with high noise and relatively small dimensions. Even with the only eleven dimensional input space, finding the nearest neighbor with brute force is ineffective for large datasets. Advanced mathematical models like k-d trees or ball trees are able to solve this problem a lot faster. For the dataset at hand, a k-d tree is used since there are robust and fast algorithms available in python through the Scikit-learn library as well as a header only C++ library called Nanoflann [107] already integrated into the GT-Lab framework. With only a few parameters to be tuned, the k-d tree hyperparameter search can be done on the complete dataset with the following parameters 6.4 In k-d trees the parameter space

Parameter	Description	Range
n_neighbors	Neighbor count	1-20
leaf_size	Leaf size	1-60
p	Minkowski p	0-2

Table 6.4: Hyperparameter search space for the nearest neighbor search

is recursively divided. Each node represents an axis aligned rectangle containing a subset of the data similar to the random forest technique. To build the tree, the dataset is split along one dimensions median value into two subsets, one containing all points above and one containing the points below the median. This process is repeated with alternating dimensions until the specified leaf size is reached. To find the nearest neighbor of a query point, we start at

the root and move up the tree until a leaf node is reached which becomes the first best guess for the nearest neighbor. The other half space must then be checked if it contains a closer point and if so the nearest neighbor is updated. Building the tree can take some time depending on the dimensionality and size of the dataset, but once built, queries can be processed very fast in a matter of milliseconds. The importance of the parameters as well as the scatter plots for the objective function are shown in figure 6.8 From the optimized

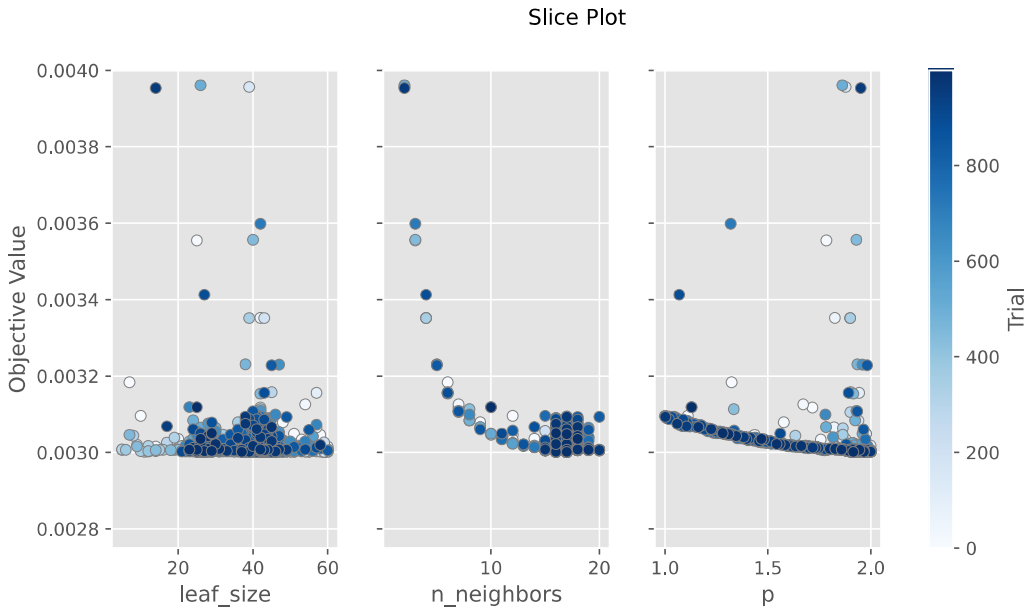


Figure 6.8: Nearest neighbor hyperparameter optimization results

parameters only the number of neighbors used for the prediction has an influence on the prediction accuracy and converges to an optimum at 17. Also, the influence of the Minkowski parameter p is small, a value of 2 gives the best results. The leaf size has no notable effect and is set to 30.

6.4.3 Neural Networks

Neural networks can be formulated with a range of measures to increase performance on high noise datasets. Dropout layers randomly set a specified fraction of neurons outputs to zero during the training process which is an effective measure against overfitting. The loss function measures the difference between the models prediction and the actual data. Common functions are the mean square error (MSE) or mean absolute error (MAE) similar to the random forest. While the mean square error produces good results on low noise data and usually results in high accuracies, the mean absolute error performs well with high noise data and outliers [108]. To combine both advantages, custom loss functions can be used with the Keras library. The Huber loss function [109] uses quadratic loss below a certain threshold and becomes linear above, which allows the handling of both, small and large errors, without penalizing the model too much. The loss functions are implemented as follows, with the δ threshold called Huber delta:

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta, \\ \delta(|a| - \frac{1}{2}\delta) & \text{otherwise.} \end{cases} \tag{6.4}$$

Additionally to the custom loss function, regularization techniques can be applied to add penalties to the loss function and further reduce the influence of outliers. Most established are the L1 and L2 regularization, with the first using the sum of the parameter values and the second using the sum of squared parameter values. A selection of the most important control parameters for a neural network

and their ranges is shown in table 6.5. Apart from noise resistance,

Parameter	Description	Tested Ranges
hidden_layer_number	Hidden layers	1-5
hidden_layer_sizes	Hidden layer neurons	16-256
dropout_rate	Dropout rate	0.2-0.5
epochs	Epochs	10-200
regularization	Regularization	0.0001, 0.1
huber_delta	Huber delta	0.1-2.0

Table 6.5: Hyperparameter search space for the neural network

the number of neurons per layer, the number of hidden layers itself and the epochs used for training influence the model accuracy and depend highly on the specific task. An Optuna hyperparameter optimization is carried out using the parameter space specified in table 6.5 and 25% of the dataset to speed up the optimization. The hyperparameter optimization showed the layer count as the most important parameter, followed by the number of neurons per layer. An overview of the parameter importance on the validation loss is given in figure 6.9. For the neural network the most important parameters are the Huber delta and the number of units. To get a better understanding of the parameter influence on the objective function, slice plots of every parameter are shown in figure 6.10. The Huber loss function described earlier provides good results for the high noise dataset as expected. Since the most important parameters, the number of layers and neurons per layer, are influenced by the size of the dataset, a second hyperparameter optimization is conducted for these two parameters. The final control parameters

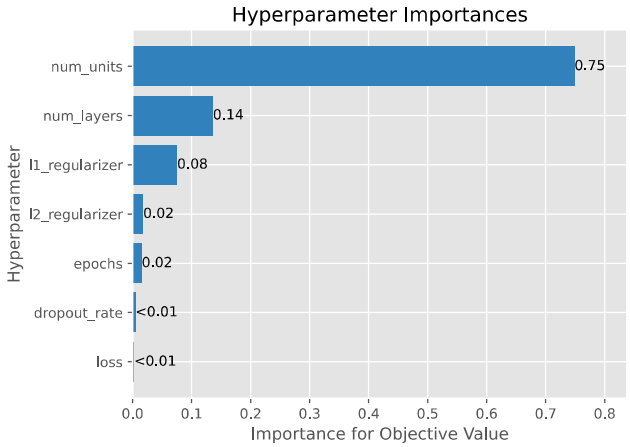


Figure 6.9: Parameter importance

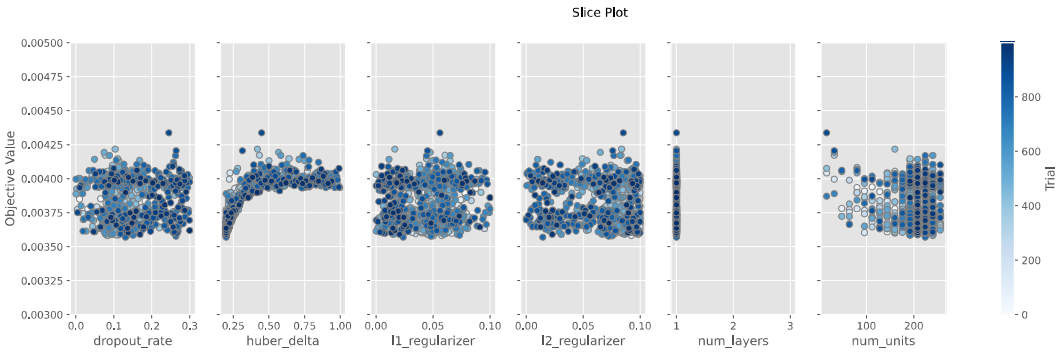


Figure 6.10: Scatter plots of the optimized parameters

for the neural network architecture are listed in table 6.6.

6.4.4 Prediction accuracy

After appropriate models have been selected and their optimal parameters have been determined, the available data must be divided into training, test and validation data sets in order to be able to independently evaluate the prediction accuracy of the individual models.

	layers	units	L1	L2	dropout	Huber δ
value	1	224	0.0239	0.0145	0.1306	0.2002

Table 6.6: Hyperparameter results for the neural network

The validation set contains 5000 data points equally distributed over the input parameter domain and is not used in the training or testing of the models. The models are compared over the mean square error (MSE) and the mean absolute error (MAE)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_{\mathrm{val}} - Y_{\mathrm{pred}})^2 \tag{6.5}$$

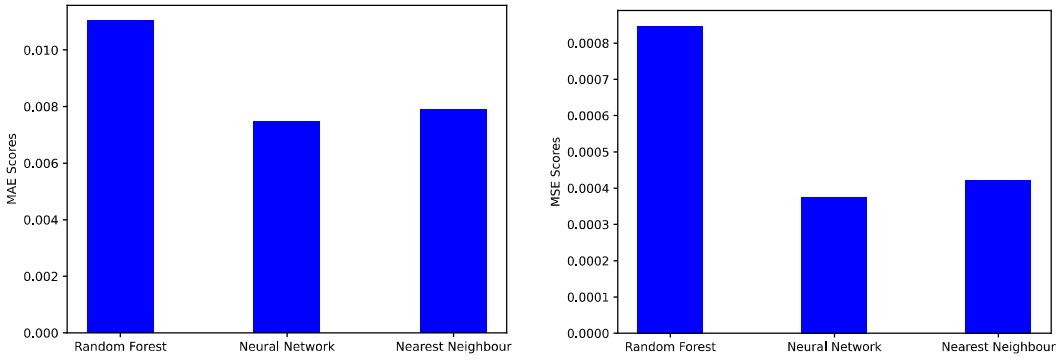
$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|Y_{\mathrm{val}} - Y_{\mathrm{pred}}| \tag{6.6}$$

of their predictions Y_{pred} to the validation data set Y_{val}. The available data generally enables two approaches going forward.

Method 1: Only optimized data points

One way to train the models is to use only the smaller dataset which contains an optimal disk geometry for each of the input data points. This reduces the input parameter space to 8 dimensions, since stress and displacement restrictions are already fulfilled for every entry in the database. The models are trained with the optimum parameters from the hyperparameter optimization. The neural network is trained using a test/train split of 25%, for the k-d tree and random forest no data splitting is needed. To compare the accuracy with the same metric for the three models, the mean square error and the

mean absolute error is evaluated on the validation dataset. Figure 6.11 shows the achieved accuracies of the validation set predictions. The nearest neighbor method and the neural network achieve similar



(a) Mean absolute error (b) Mean square error

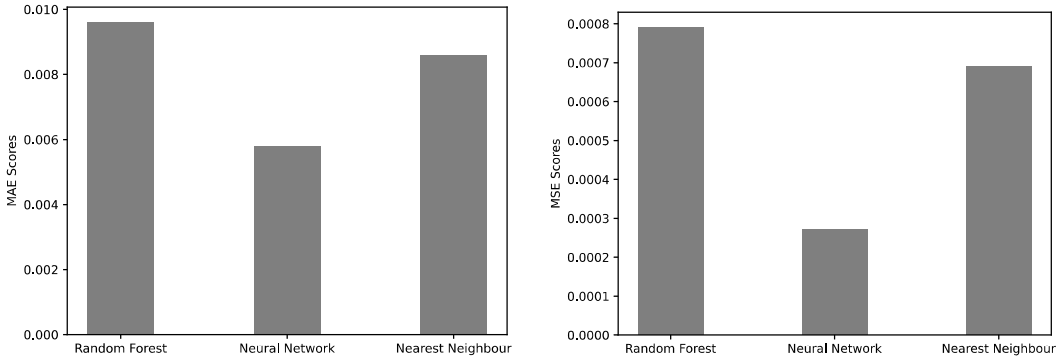
Figure 6.11: Prediction accuracies for the different models

MSE values, yet the error of the random forest model is significantly higher for the MSE as well as the MAE. This indicates that on average both models predictions are equally accurate, but for some predictions the nearest neighbor method produces predictions which are significantly off. The larger errors of the nearest neighbor method could be filtered by additional checks for example the distance to the next data point, but in general the neural network shows to be the preferred surrogate model.

Method 2: All data points

The second approach is including the non-optimal data points in the training data, along using the optimal dataset from method one as target values. While the complexity is increased, this can have

beneficial influence on the model accuracy. This approach is known as adversarial training or negative data augmentation. By exposing the model to a more diverse range of inputs, the robustness and generalization can be increased. The model is exposed to data points not representing optimal solution, due to physical limits of the optimization problem or a failure to find the global optimum, through which it is able to learn failure modes and the boundaries of the dataset. The achieved accuracies for the same models used for method one are shown in figure 6.12 The outcomes indicate that



(a) Mean absolute error (b) Mean square error

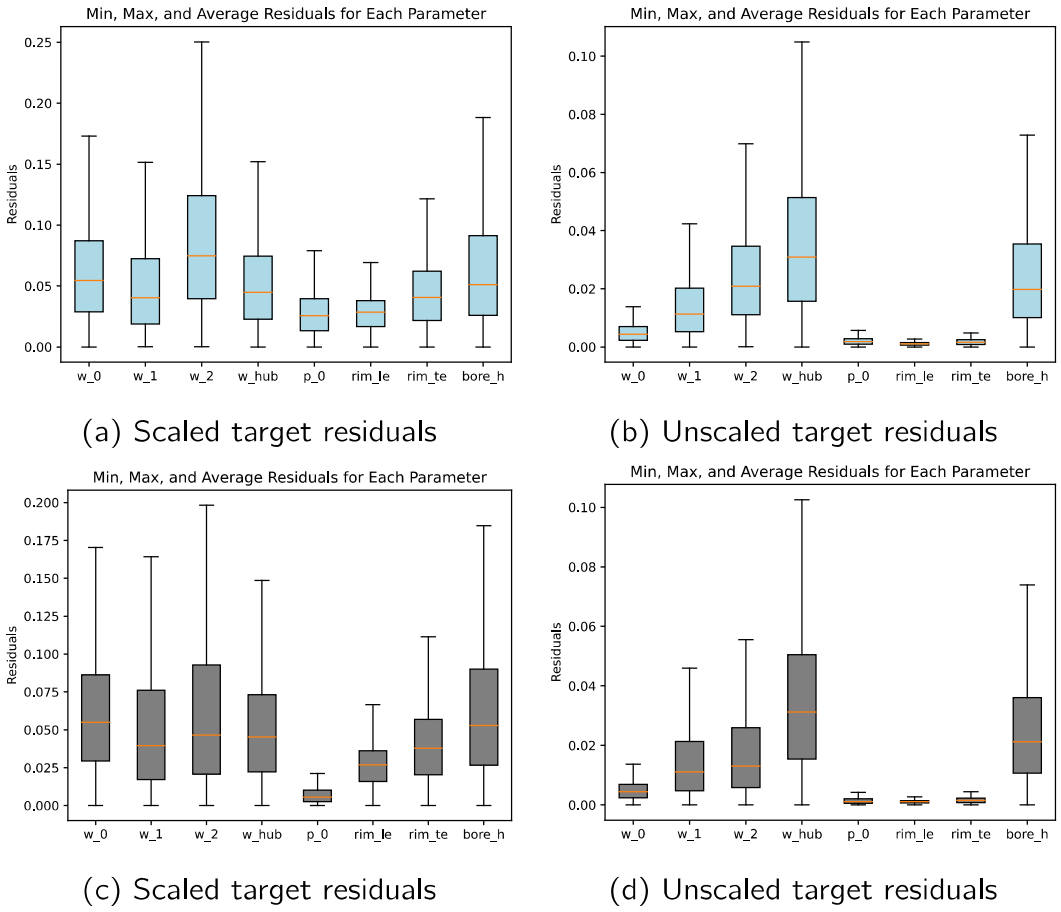
Figure 6.12: Prediction accuracies for the different models

the neural network model consistently outperforms others in terms of accuracy. However, unlike with the first method, the nearest neighbor approach yields suboptimal results. This can be attributed to the model's inherent architecture, which tends to identify numerous non-optimal data points present within the dataset. The prediction accuracy can not be compared directly with method one, because the normalization limits are different for the two datasets, which changes the error metric. A detailed comparison of the best

model results, the neural networks, for both methods is conducted throughout the next section.

6.5 Interpret

The interpretation of the predicted results and the deployment of the model are the final steps in the data life cycle. The predictions in this section are performed with the best scoring model, the neural network with measures for noisy data. To get a better understanding of the accuracies of the surrogate models, the target parameter predictions can be analysed regarding their minimal, mean and maximum deviations from the validation dataset. The deviations can be visualized with box plots as shown in figure 6.13. For every parameter the box contains 50% of the data points and the bar indicates the 25th and 75th percentile, with the orange line being the median value. The box plot with scaled data shows the efforts of the neural networks to produce as small errors as possible, showing comparable absolute values for each parameter. When unscaled to the initial parameter range, the width of the hub ring shows the highest deviation from the desired value. This could be expected since the absolute value range for the hub ring width is the highest for the target values as seen in figure 6.5. The results for method two show a slightly higher accuracy for the scaled residuals with 0.00592 vs 0.00750 for method one. Since the overall range of the parameters is higher than for method one, yet for the unscaled accuracy, both methods achieve nearly identical results as shown in figure 6.13 b



(a) Scaled target residuals

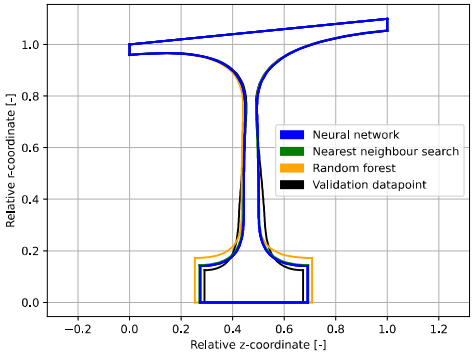
(b) Unscaled target residuals

(c) Scaled target residuals

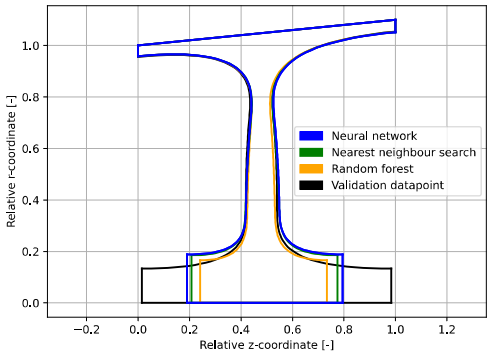
(d) Unscaled target residuals

Figure 6.13: Residuals for the neural network with method 1 (blue) and method 2 (grey)

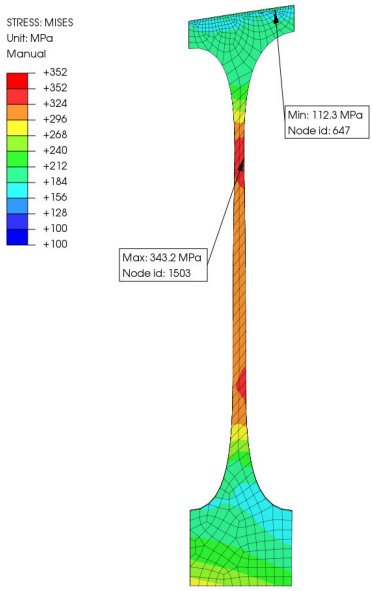
and c. Since the improvements using method two are very small, but result in highly increased calculation times (training time increases with a factor of 21), the detailed inspection of the result is carried out using only method one. The target values represent the geometrical parametrization of the disk as described in section 4. To visualize the actual deviations for the disk geometry, single data points can be plotted as 2D contour plots. The optical comparison in figure 6.14 shows the actual geometric deviations



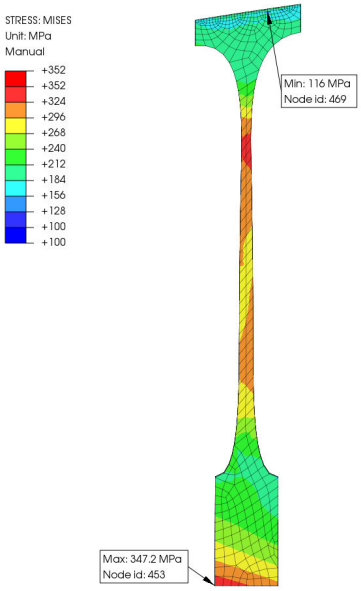
(a) Average deviation (relative scale)



(b) Worst case deviation (relative scale)



(c) Worst case reference Mises stress

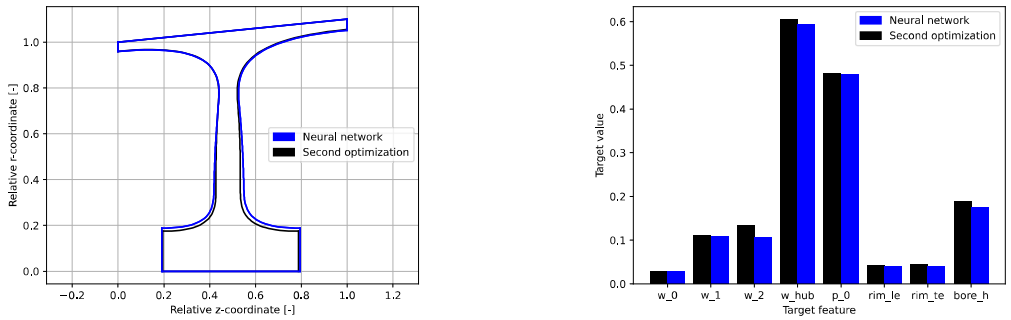


(d) Worst case prediction Mises stress

Figure 6.14: Predicted vs benchmark disk shape

for a data point with an average deviation on the left and the worst case from the validation dataset on the right. At first glance the deviation of the worst case contour plot seems way off the validation data point, yet the predicted disk shape itself is physically feasible,

while the validation disk shape is untypical. This could indicate a valid solution for the prediction instead of the optimized validation data point. To evaluate the prediction quality, a FE calculation is performed for the neural network prediction and the reference point from the validation set. The Mises stress distribution of both disk geometries is also displayed in figure 6.14 c and d. As suspected both disks are valid solutions and respect the given stress limits of 352MPa, yet the predicted shape shows lower disk volume than the reference. This indicates that the cause of high deviations in the predictions is not necessarily an uncertainty in the used model, but results from the underlying optimization process which generates the training and validation data. As mentioned in chapter 5, NLOpt struggles to reliably find the global optimum. When training the neural network on the imperfect dataset, the model seemingly finds a connection between global optima and treats local optima as noise and outliers. To further prove this assumption, the optimization of the worst case data point is repeated with a different starting solution and initial step sizes to aid the optimizer in finding the global optimum. The second optimization result plotted against the former worst case prediction as well as their absolute deviations in figure 6.15 shows now very good consistency.

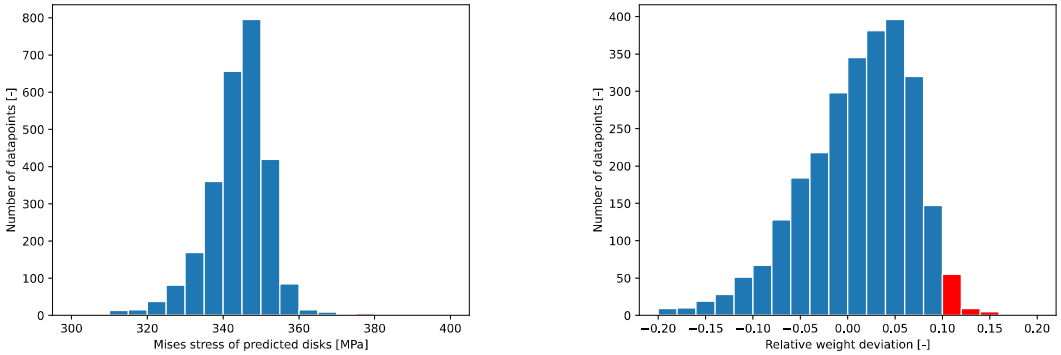


(a) Disk shape comparison

(b) Target parameter comparison

Figure 6.15: Second optimization reference analysis

A quantitive evaluation of the overall prediction accuracy has to involve a FE calculation of all predicted geometries, since the relevant parameters for the disk are the resulting stresses and disk volumes and not the deviations to outliers. Using a python calculator in GTLab and the process elements developed in chapter 4, FE calculations are performed for all predicted disk shapes for the validation dataset and are stored in a csv database. The stresses and disk volumes of the predictions can now be directly compared to the validation data with the histograms in figure 6.16. With the Mises stress on the left and a comparison of the disk volumes on the right. The real prediction accuracy can now be calculated with two conditions. For a valid prediction, two conditions must be fulfilled. The predicted Mises stress must not exceed the defined stress limit. With the uncertainty factor introduced in chapter 5 the resulting absolute stress limit is 391.11MPa. Second, the volume of the predicted disk must not exceed the corresponding validation data points disk volume with a tolerance of 10%. Using this metric



(a) Mises stresses and outliers

(b) Deviation in disk volume

Figure 6.16: Prediction accuracy conditions

the predicted disk shapes for the validation dataset are in 96.7% of the cases within the specified limits. Despite this high accuracy, erroneous predictions need to be intercepted. The predicted disks can be meshed and simulated with the same process used for the optimization. Since the FE calculation is very fast, every prediction can be instantly checked against the actual static Mises stresses and if the check shows stresses exceeding the stress limit, a warning is generated, effectively adding around 0.5s to the dimensioning task.

To apply the developed design method on other use cases in the future, the necessary amount of data points is an important factor. To quantify the size of the data points necessary for each model, the training is repeated with fractions of the complete dataset and evaluated on the achieved mean square error. The results are shown in figure 6.17. The random forest performs relatively well even on small dataset sizes, yet shows no performance increase with growing data amounts. The neural network and nearest neighbour method

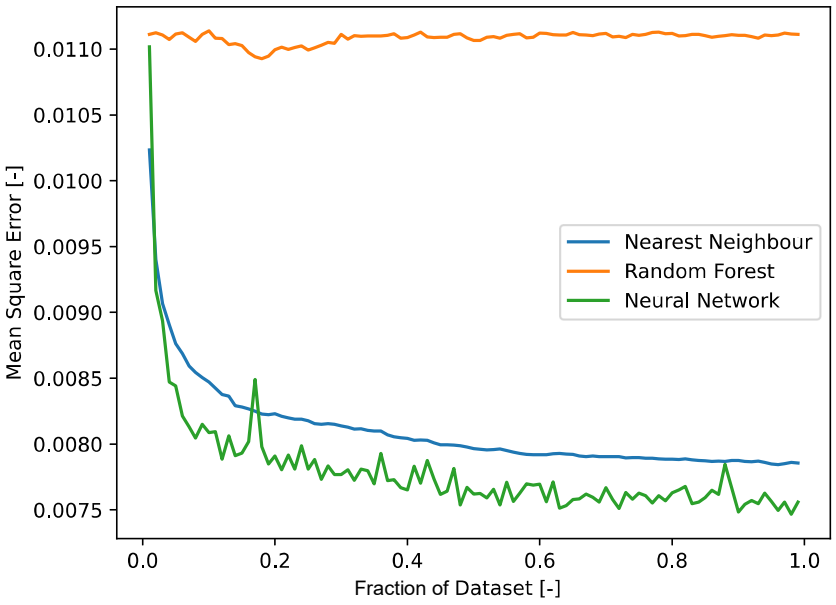


Figure 6.17: Prediction accuracy depending on the dataset size

show similar behavior with a sharp increase of prediction accuracy up to 20% of the available data. The accuracy increase slows down at around 70% indicating that a further increase of training data will not have a significant influence on the model prediction accuracy.

As a final evaluation factor, the robustness and boundaries of the model are analysed. Using a database consisting of input parameters which exceed the ranges of the training data, the extrapolation performance of the model can be evaluated. The influence between the distance of the selected parameter to the training data points and the mean square error is visualized in figure 6.18. In general the error is proportional to the distance from the next trained data point. For the tested outliers, the model accuracy did not degrade

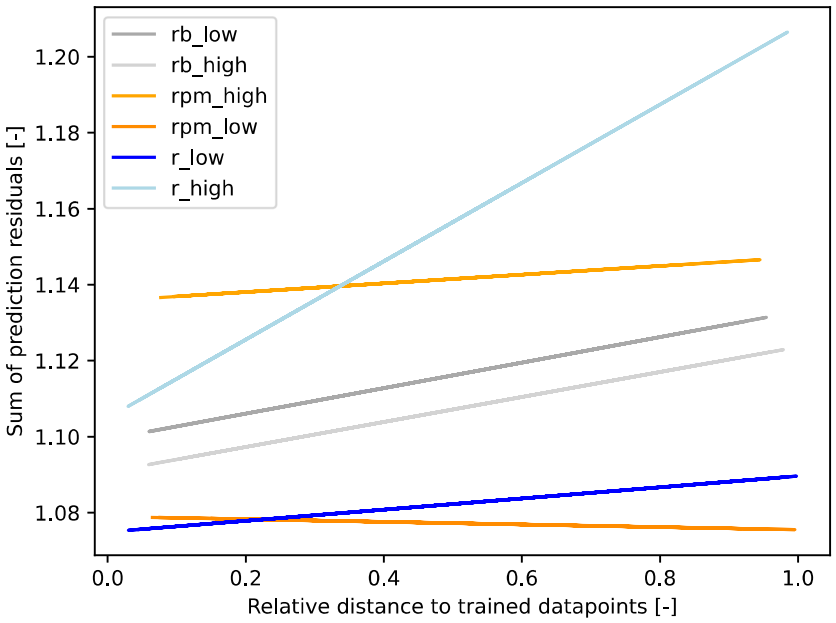


Figure 6.18: Model sensitivity against selected input parameter outliers

significantly and proves to be resistant to outliers, at least in the tested ranges. The highest influence shows the inner diameter of the disk design space, which increases the prediction error with approximately 10% when doubling the highest value present in the training dataset. For this purpose a warning system is implemented when the query point is outside the training dataset to indicate possible prediction inaccuracies. For the distance calculation, the nearest neighbor search is used.

7 Conclusion

The final chapter will give a summary of the development steps and give a quantitative evaluation of the developed design method, followed by an outlook regarding planned improvements and the application possibilities.

7.1 Summary

This thesis addresses the structural sizing of compressor components using machine learning (ML) methods demonstrated with an application for compressor blisks. A geometric parameterization for compressor disks was created, characterized by only eight independent parameters while maintaining a high degree of design flexibility. Advanced meshing processes were then developed to prepare for structural analysis, using the finite element method to assess stress and displacements. Among different solvers, CalculiX emerged as the most efficient for its balance of speed and accuracy. The methodology established was used as a foundation to optimize the disk geometry with the local optimizing algorithm COBYLA. The design goal was to minimize mass while adhering to design constraints like stress and displacement ratios. The optimization was

fully automated within the GTlab framework. The resulting synthetic datasets containing optimal disk shapes for the corresponding input condition were used to train and test various surrogate models. As model architecture random forests, a nearest neighbor method and a feed forward neural network with measures against noise were investigated. The neural network showed the best results regarding robustness and prediction accuracy. After validation, the model was deployed as part of the structural mechanic model in the GTlab framework.

The design process made use of the following open-source or freeware tools and libraries:

- Calculix: Open-source FEM Solver
- Gmsh: Open-source meshing tool
- GTLab: Framework for jet engine development, DLR property
- ReBAR: ML pipeline framework, open-sourcing planned
- NLOpt: Open-source optimization library
- KERAS: ML library, freeware
- Scikit-learn: ML library, freeware
- Optuna: Hyperparameter optimizer library, freeware

The developed method showed high accuracy with an overall value of 96.7% against the validation database. To intercept bad predictions for the remaining 3.3%, a warning system was implemented

using a subsequent FE analysis for every prediction. For a quantitative evaluation of the developed design methods, the criteria accuracy and calculation time are compared with two state-of-the-art methods, the preliminary design method from the WATE code described in 2.1.2 implemented in GTlab and a detailed global optimization. A comparison over the necessary calcualtion time over the course of the thesis development is shown in figure 7.1 The

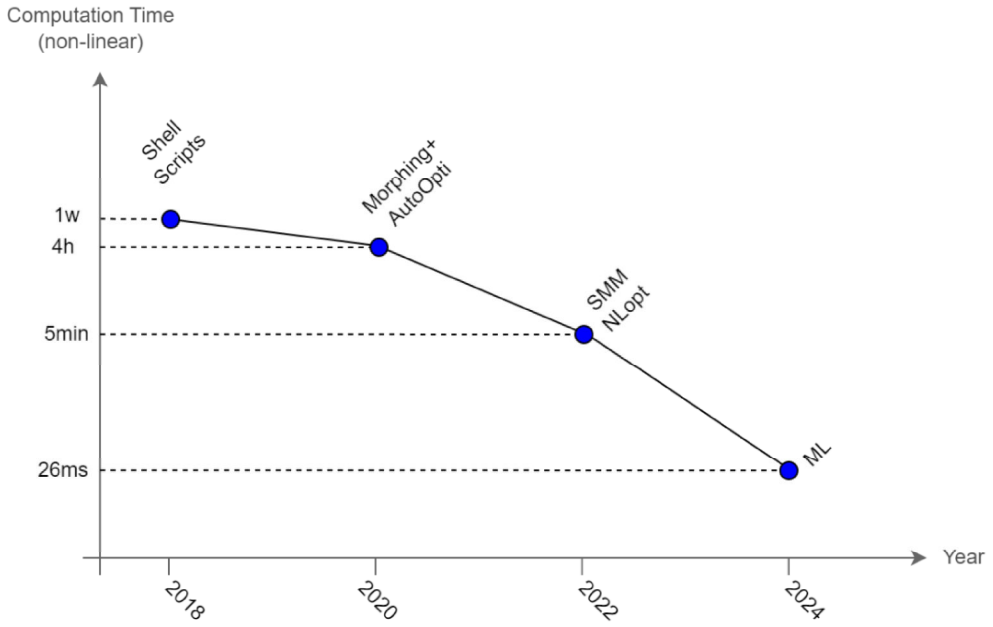


Figure 7.1: Calculation time decrease

novel method allows for real time design of compressor blisks. The medium calculation time is 12ms for the prediction itself and 0.5s including the subsequent FE check. This beats even the preliminary method which takes on average 3.2s to compute. In order to compare the accuracy against the preliminary design method, the validation dataset can be used. A preliminary design is performed

for every data point and checked with a subsequent FE calculation. Using the same metric already used for the predictive method, an overall accuracy of 61.0% can be stated. In most cases, the low accuracy can be attributed to breached stress and displacement restrictions. When using only the weight as metric, the preliminary design method performs reasonably well with an accuracy of 89.4%, which is the main task the method was intended for, but still falls short against the surrogate.

A comparison with a detailed design method is objectively difficult because there are no public global optimization methods available. The method is already significantly faster than the local optimization process described in this work, which takes on average 3 minutes and 12 seconds. Since the development of the optimization process is part of this thesis, the calculation time should be compared against the state of the art before the work described here. Speaking only for the processes available at the department of design and manufacturing technology of the German Aerospace Center, a global disk optimization had to be set up for every single rotor manually. Although single processes, for example meshing tools, were available and could be reused, generating an optimization process would take days, which sets the developed design method on a new level calculation timewise. With a suitable global optimization algorithm and no limit on calculation time, the most weight efficient disk shape should be found in 100% of the cases, which beats the prediction methods accuracy of 96.7%. The best solution here is to use the predictive design method with subsequent FE checks and in case of

bad predictions an optimization can be performed.

With this integrated approach of a fast, accurate FEM simulation, the creation of an extensive database of thousands of simulation runs, data cleaning and the construction of an ML tool trained with the database, a unique simulation tool was created that can generate the corresponding disk design in near real time and with high accuracy when the aerodynamic profile of a blisk blade changes.

7.2 Outlook

The developed design method is already used in the DLR project Assessment and Digitalization of Forthcoming Propulsion Technologies (ADAPT) and the DLR project Future Fighter Engine (FFE) to give fast and accurate predictions for the overall compressor weight at very early design stages. With the achieved high accuracy and safety measures, the method is also used for detailed design tasks in late design stages. Although the method performs reasonably well on low pressure compressors, given that the input parameters remain confined to the constraints of the training dataset, there is notable depreciation in the prediction accuracy when the method is applied on blade geometries or loading conditions that were not represented in the initial training set. To overcome this limitation and expand the method's utility to high pressure compressor and turbine rotors, it is essential to augment the training dataset. This expansion should include a broader spectrum of blade geometries, a

wider array of loading conditions, and varied material types, thereby enhancing the algorithm's capability for generalization. While the generalization does not necessitate modification of the underlying process, an improvement of the method would lead to a reduced required dataset and therefore computational time. Two promising strategies to increase the efficiency of the method are the refinement of data quality and the improvement of the data pipelines. While the method achieved high accuracy using the local optimizer COBYLA, the accuracy would benefit from higher data quality. The use of advanced optimization algorithms will be investigated. Promising approaches are surrogate optimization and evolutionary algorithms as for example provided by the open-source framework for multi-disciplinary design, analysis and optimization (MDAO) [110]. Introducing a surrogate into the optimization would allow to transfer the knowledge gained about the design task between the individual optimization runs, which could speed up the data generation while maintaining high data quality. The investigation of the mentioned aspects is currently addressed in the DLR project ADAPT. To allow the application of the method for other use cases, the automation of the data pipelines is an important aspect for the effectivity and repeatability. The process described in chapter 6 needs to be fully automized, while still allowing flexibility in the data structure. The data pipeline automatization is currently addressed in the DLR project ReBAR.

The generated database can be used for further studies regarding the sensitivity of disk properties to geometric changes. In order to

extend the method for different compontents the database can be used as a baseline to create a physical understanding of the problem. If geometric parameters are changed or added, a completely new training database needs to be generated since there are now transfer methods implemented at the moment.

In future work mult criteria decision making is a promising approach to increase the usability of the model. Method B described in chapter 6.4 is already a step in this direction. When training not on only weight optimal disks, but the whole database, the model can be enabled to predict disks with certain other desired features like for example a disk with very low radial displacement.

Both mentioned points will benefit from more powerful computation hardware, provided through the DLR investment program CUBE, which was not finished at the time of writing. For the mid and long-term strategy, the method could be applied for example to simulations for the dimensioning of fan blades with respect to bird strike or aeroelasticity simulations, which both have very long calculation times and can at the moment not be used in multidisciplinary optimization runs.

Acronyms

ACDC Advanced Compressor Design Code.

AI Artificial Intelligence.

API Application Programming Interface.

ART Adaptive Resonance Theory.

AT Institute of Propulsion Technology.

BC Boundary Condition.

BGF Design and Manufacturing Technology Department.

BLING Bladed Ring.

BLISK Bladed Disk.

BNN Bayesian Neural Network.

BPR Bypass Ratio.

BT Institute of Structures and Design.

CAD Computer-Aided Design.

CFD Computational Fluid Dynamics.

CNC Computer Numerical Control.

CNN Convolutional Neural Network.

CO2 Carbon Dioxide.

CPACS Common Parametric Aircraft Configuration Schema.

DAG Directed Acyclic Graph.

DLE Disk Leading Edge.

DLR German Aerospace Center.

DNN Deep Neural Network.

DTE Disk Trailing Edge.

FEM Finite Element Method.

FMEA Failure Mode and Effects Analysis.

FUV Fan and Compressor Department.

GAN Generative Adversarial Network.

GD Gradient Descent.

GTlab Gas Turbine Laboratory.

HDF Hierarchical Data Format.

HEX Hexahedron Element.

IGES Initial Graphics Exchange Specification.

IPCC Intergovernmental Panel on Climate Change.

LE Leading Edge.

Ltd Limited.

MAE Mean Absolute Error.

ML Machine Learning.

MOS Meta Object System.

MPC Multi Point Constraint.

MSE Mean Squared Error.

MTU Motoren- und Turbinenunion.

NLR Netherlands Aerospace Center.

NoX Nitrogen Oxides.

NURBS Non-Uniform Rational B-Spline.

OSEMN Obtain Scrub Explore Model Interpret.

PD Predictive Design.

PLC Public Limited Company.

PS Pressure Side.

RF Random Forest.

RL Reinforcement Learning.

SL Supervised Learning.

SMM Structural Mechanic Module.

SOM Self-Organizing Map.

SS Suction Side.

STEP Standard for the Exchange of Product Data.

SVM Support Vector Machine.

TE Trailing Edge.

TFI Transfinite Interpolation.

TPS Turbine Propulsion Simulator.

TWK Engine Department.

UHBR Ultra High Bypass Ratio.

UML Unified Modeling Language.

List of Figures

List of Tables

Bibliography

[1] IEA. "Aviation Report". In: (2022). URL: https://www.iea.org/reports/aviation.

[2] D.S. Lee et al. "The contribution of global aviation to anthropogenic climate forcing for 2000 to 2018". In: *Atmospheric Environment* 244 (2021), p. 117834. ISSN: 1352-2310. DOI: https://doi.org/10.1016/j.atmosenv.2020.117834. URL: https://www.sciencedirect.com/science/article/pii/S1352231020305689.

[3] Hannah Ritchie, Max Roser, and Pablo Rosado. "CO2 and Greenhouse Gas Emissions". In: *Our World in Data* (2020). URL: https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions.

[4] IEA. "World Energy Outlook 2019". In: (2019). URL: https://www.iea.org/reports/world-energy-outlook-2019.

[5] Marc C. Gelhausen, Peter Berster, and Dieter Wilken. *Part III - Forecasting future air traffic development up to 2040 and assessing mitigation strategies*. Ed. by Marc C. Gelhausen, Peter Berster, and Dieter Wilken. Academic Press, 2020, p. 223. ISBN: 978-0-12-812657-8. DOI: https://doi.org/10.1016/B978-0-12-812657-8.00019-1. URL: https://www.sciencedirect.com/science/article/pii/B9780128126578000191.

[6] IPCC. *Global Warming of 1.5°C: IPCC Special Report on Impacts of Global Warming of 1.5°C above Pre-industrial Levels in Context of Strengthening Response to Climate Change, Sustainable Development, and Efforts to Eradicate Poverty*. Cambridge University Press, 2022. DOI: 10.1017/9781009157940.

[7] Wikipedia. *Technology Readiness Level — Wikipedia, die freie Enzyklopädie*. [Online; Stand 27. Januar 2023]. 2021. URL: https://de.wikipedia.org/w/index.php?title=Technology_Readiness_Level&oldid=210985433.

[8] Fuel Cells and Hydrogen 2 Joint Undertaking. *Hydrogen-powered aviation : a fact-based study of hydrogen technology, economics, and climate impact by 2050*. Publications Office, 2020. DOI: doi/10.2843/471510.

[9] Keisuke Hanaki and Joana Portugal-Pereira. "The Effect of Biofuel Production on Greenhouse Gas Emission Reductions". In: *Biofuels and Sustainability: Holistic Perspectives for Policy-making*. Ed. by Kazuhiko Takeuchi et al. Tokyo: Springer Japan, 2018, pp. 53–71. ISBN: 978-4-431-54895-9. DOI: 10.1007/978-4-431-54895-9_6. URL: https://doi.org/10.1007/978-4-431-54895-9_6.

[10] Samarth Jain et al. "Estimating the Reduction in Future Fleet-Level CO2 Emissions From Sustainable Aviation Fuel". In: *Frontiers in Energy Research* 9 (2021). ISSN: 2296-598X. DOI: 10.3389/fenrg.2021.771705. URL: https://www.frontiersin.org/articles/10.3389/fenrg.2021.771705.

[11] Jarosław Kozuba and Mateusz Ojciec. "Overview of historical and future trends of commercial aircraft fuel efficiency". In: *Acta*

Avionica Journal (June 2019), pp. 12–17. DOI: 10.35116/aa.2019.0003.

[12] A. Dik et al. "Conceptual Design of a 3-Shaft Turbofan Engine with Reduced Fuel Consumption for 2025". In: *Energy Procedia* 142 (2017). Proceedings of the 9th International Conference on Applied Energy, pp. 1728–1735. ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2017.12.556. URL: https://www.sciencedirect.com/science/article/pii/S1876610217363129.

[13] *Turbofan Engine Bypass Ratio as a Function of Thrust and Fuel Flow*. Mechanical Engineering and Materials Science. 2017.

[14] Arslon. *Moore's Law*. Ed. by The Strategist. 2021. URL: https://www.aspistrategist.org.au/graph-of-the-week-moores-law/.

[15] A. Mateo. "BLISK Fabrication by Linear Friction Welding". In: Nov. 2011. ISBN: 978-953-307-611-9. DOI: 10.5772/21278.

[16] W.A. Hunter, G.A. Grimmer, and GENERAL ELECTRIC CO LYNN MA AIRCRAFT ENGINE GROUP. *T700 Blisk and Impeller Manufacturing Process Development Program*. Defense Technical Information Center, 1979. URL: https://books.google.de/books?id=-dBJNwAACAAJ.

[17] D. Dilba. "Blisk development: How blade and disk became one". In: *AEROREPORT* (May 2019). URL: https://aeroreport.de/en/aviation/blisk-development-how-blade-and-disk-became-one.

[18] *Design Improvements of the EJ 200 HP Compressor: From Design Verification Engine to a Future All Blisk Version*. Vol. Volume 1: Aircraft Engine; Marine; Turbomachinery; Microturbines

and Small Turbomachinery. Turbo Expo: Power for Land, Sea, and Air. V001T01A001. June 2001. DOI: 10.1115/2001-GT-0283. eprint: https://asmedigitalcollection.asme.org/GT/proceedings-pdf/GT2001/78507/V001T01A001/2414598/v001t01a001-2001-gt-0283.pdf. URL: https://doi.org/10.1115/2001-GT-0283.

[19] Rolls Royce Press. *Rolls-Royce ships 10,000th blisk from Oberursel*. Apr. 2019. URL: https://www.rolls-royce.com/media/press-releases/2019/11-04-2019-ships-10000-blisk-from-oberursel.aspx.

[20] Paul Breeze. "Chapter 5 - Advanced Gas Turbine Design". In: *Gas-Turbine Power Generation*. Ed. by Paul Breeze. Academic Press, 2016, pp. 43–53. ISBN: 978-0-12-804005-8. DOI: https://doi.org/10.1016/B978-0-12-804005-8.00005-7. URL: https://www.sciencedirect.com/science/article/pii/B9780128040058000057.

[21] Thomas Klauke. "Schaufelschwingungen realer integraler Verdichterräder im Hinblick auf Verstimmung und Lokalisation". PhD thesis. Jan. 2008.

[22] S. Hansen. "Industry 4.0 in aviation". In: *AEROREPORT* (May 2016). URL: https://aeroreport.de/en/innovation/industry-4-0-in-aviation.

[23] Alexander Ernst. "Analyse und Modellierung der Einflüsse auf das Arbeitsergebnis der elektrochemischen Bearbeitung hochtemperaturfester Werkstoffe für Turbinenstrahltriebwerke". PhD thesis. 2020. DOI: http://dx.doi.org/10.22028/D291-33356.

[24] R. Grammel. "Neue Lösungen des Problems der rotierenden Scheiben". In: *Ingenieur-Archiv* VII.Band.3.Heft (1936).

[25] C. B. Biezeno and R. Grammel. "Rotierende Scheiben". In: *Technische Dynamik: Zweiter Band Dampfturbinen und Brennkraftmaschinen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1953. ISBN: 978-3-642-65189-2. DOI: 10.1007/978-3-642-65189-2_1. URL: https://doi.org/10.1007/978-3-642-65189-2_1.

[26] G.D. Smith. *Numerical Solution of Partial Differential Equations: With Exercises and Worked Solutions*. Oxford mathematical handbooks. Oxford University Press, 1965. ISBN: 9780198596110. URL: https://books.google.de/books?id=wnKNq67GYrEC.

[27] Sasan C. Armand. "Structural Optimization Methodology for Rotating Disks of Aircraft Engines". In: *NASA Technical Memorandum* 4693 (1995).

[28] S. Manson. "Determination of elastic stresses in gas-turbine disks". In: *NACA-TR-871* (1947).

[29] M. Millenson. "Determination of stresses in gas-turbine disks subjected to plastic flow and creep". In: *NACA-TR-906* (1948).

[30] Michael Tong, Ian Halliwell, and Louis Ghosn. "A Computer Code for Gas Turbine Engine Weight and Disk Life Estimation". In: *Journal of Engineering for Gas Turbines and Power-transactions of The Asme - J ENG GAS TURB POWER-T ASME* 126 (Apr. 2004). DOI: 10.1115/1.1691980.

[31] D. Gutzwiller. "T-Axi Disk V2.2 User's Guide and Tutorial". In: (2009).

[32] D. Gutzwiller. "Automated Design, Analysis, and Optimization of Turbomachinery Disks". In: (2009). URL: http://rave.ohiolink.edu/etdc/view?acc_num=ucin1258579409.

[33] A.G.M. Michell. "The Limits of Economy of Material in Frame Structures. Philosophical Magazine". In: *Philosophical Magazine* Series 6 (1904).

[34] Altair. *Element biasing*. URL: `https://2021.help.altair.com/2021/hwdesktop/hm/topics/pre_processing/meshing/meshing_element_biasing_r.htm`.

[35] Hans Eschenauer and N. Olhoff. "Topology optimization of continuum structures: A review". In: *Applied Mechanics Reviews - APPL MECH REV* 54 (July 2001). DOI: 10.1115/1.1388075.

[36] Martin Philip Bendsoe and Ole Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer, 2004. ISBN: 9783540429920.

[37] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals, Sixth Edition*. 6th ed. Butterworth-Heinemann, 2005. ISBN: 0750663200.

[38] B. Klein. *FEM: Grundlagen und Anwendungen der Finite-Element-Methode im Maschinen- und Fahrzeugbau*. Springer Vieweg Studium. Maschinenelemente und Konstruktion. Vieweg und Teubner Verlag, 2012. ISBN: 9783834821348. URL: `https://books.google.de/books?id=YisgBAAAQBAJ`.

[39] Ravil Mukhamediev. "Machine learning methods: An overview". In: *CMNT* 19 (Jan. 2015), pp. 14–29.

[40] Victoria Hodge, Richard Hawkins, and Rob Alexander. "Deep reinforcement learning for drone navigation using sensor data". In: *Neural Computing and Applications* 33 (Mar. 2021). DOI: 10.1007/s00521-020-05097-x.

[41] H. ADELI and C. YEH. "Perceptron Learning in Engineering Design". In: *Computer-Aided Civil and Infrastructure Engineering* 4.4 (1989), pp. 247–256. DOI: https://doi.org/10.1111/j.1467-8667.1989.tb00026.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8667.1989.tb00026.x.

[42] *Automated Multiobjective Optimisation in Axial Compressor Blade Design*. Vol. Volume 6: Turbomachinery, Parts A and B. Turbo Expo: Power for Land, Sea, and Air. May 2006, pp. 1289–1297. DOI: 10.1115/GT2006-90420. URL: https://doi.org/10.1115/GT2006-90420.

[43] *Application of Kriging Metamodels to the Automated Start Value Generation for Gas Turbine Performance Simulations*. Vol. Volume 1: Aircraft Engine; Fans and Blowers; Marine. Turbo Expo: Power for Land, Sea, and Air. V001T01A024. June 2018. DOI: 10.1115/GT2018-76153. eprint: https://asmedigitalcollection.asme.org/GT/proceedings-pdf/GT2018/50985/V001T01A024/2788089/v001t01a024-gt2018-76153.pdf. URL: https://doi.org/10.1115/GT2018-76153.

[44] Seowoo Jang, Soyoung Yoo, and Namwoo Kang. "Generative Design by Reinforcement Learning: Enhancing the Diversity of Topology Optimization Designs". In: *Computer-Aided Design* 146 (2022), p. 103225. ISSN: 0010-4485. DOI: https://doi.org/10.1016/j.cad.2022.103225. URL: https://www.sciencedirect.com/science/article/pii/S0010448522000239.

[45] Oded Amir and Emad Shakour. "Simultaneous shape and topology optimization of prestressed concrete beams". In: *Structural and Multidisciplinary Optimization* 57 (May 2018). DOI: 10.1007/s00158-017-1855-5.

[46] Fatima Alsakka et al. "Generative design for more economical and environmentally sustainable reinforced concrete structures". In: *Journal of Cleaner Production* 387 (2023), p. 135829. ISSN: 0959-6526. DOI: https://doi.org/10.1016/j.jclepro.2022.135829. URL: https://www.sciencedirect.com/science/article/pii/S0959652622054038.

[47] Shawn Joseph D'mello, S. Renold Elsen, and J. Ronald Aseer. "Generative design study of a remote-controlled plane's wing ribs". In: *AIP Conference Proceedings* 2283.1 (2020), p. 020046. DOI: 10.1063/5.0024955. URL: https://aip.scitation.org/doi/abs/10.1063/5.0024955.

[48] Seowoo Jang and Namwoo Kang. "Generative Design by Reinforcement Learning: Maximizing Diversity of Topology Optimized Designs". In: *ArXiv* abs/2008.07119 (2020).

[49] Tristan Briard, Frederic Segonds, and Nicolo Zamariola. "G-DfAM: a methodological proposal of generative design for additive manufacturing in the automotive industry". In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 14 (Sept. 2020). DOI: 10.1007/s12008-020-00669-6.

[50] Aleksandar Čučaković, Biljana Jovic, and Mirjana Komnenov. "Biomimetic Geometry Approach to Generative Design". In: *Periodica Polytechnica Architecture* 47 (Jan. 2016), pp. 70–74. DOI: 10.3311/PPar.10082.

[51] Domenico D'Uva. "Morphogenesis and panelling, the use of generative tools beyond academia. Case studies and limits of the method." In: Sept. 2013. DOI: 10.13140/2.1.4716.5446.

[52] Conner Sharpe et al. "A Comparative Evaluation of Supervised Machine Learning Classification Techniques for Engineering Design Applications". In: *Journal of Mechanical Design* 141.12 (Oct. 2019). 121404. ISSN: 1050-0472. DOI: 10.1115/1.4044524. URL: https://doi.org/10.1115/1.4044524.

[53] Stanislaus Reitenbach et al. "Collaborative Aircraft Engine Preliminary Design using a Virtual Engine Platform, Part A: Architecture and Methodology". In: Jan. 2020. DOI: 10.2514/6.2020-0867.

[54] Marvin Nöthen. "Integration eines Python-Interpreters innerhalb der Triebwerksvorentwurfsumgebung GTlab zur Steuerung des Programmablaufs". In: (2019). URL: https://elib.dlr.de/134142/.

[55] R.-G. Becker et al. "An Integrated Method for Propulsion System Conceptual Design". In: vol. Volume 1: Aircraft Engine; Fans and Blowers; Marine. Turbo Expo: Power for Land, Sea, and Air. June 2015. DOI: 10.1115/GT2015-43251. URL: https://doi.org/10.1115/GT2015-43251.

[56] Grady Booch, James Rumbaugh, and Ivar Jacobson. *Unified Modeling Language User Guide, The 2nd Edition*. Addison-Wesley Professional, 2005. ISBN: 0321267974.

[57] Matthias Schulze et al. "Parametric modelling of a long range aircraft under consideration of engine wing integration". In: *DLRK 2019 - Deutscher Luft- und Raumfahrtkongress*. 2019. URL: https://elib.dlr.de/128059/.

[58] et al. Gino Duffet Klaus Wolf. "A new Interface Standard for Integrated Virtual Material Modelling in Manufacturing Industry". In: *ITEA* (2020).

[59] Marius Broecker. "Implementierung und Integration einer GUI Testing-Methodik in Continuous Integration Prozesse". In: (Jan. 2020).

[60] Holger Weber. "REBAR, Reducing Barriers for AI in (applied) Research". In: *DLR WaW* (2022).

[61] A. Lastra. *Parametric Geometry of Curves and Surfaces: Architectural Form-Finding*. Mathematics and the Built Environment. Springer International Publishing, 2021. ISBN: 9783030813178. URL: https://books.google.de/books?id=05RBEAAAQBAJ.

[62] Carl de Boor. "Spline Basics". In: Jan. 2002, pp. 141–163. ISBN: 9780444511041. DOI: 10.1016/B978-044451104-1/50007-1.

[63] Les Piegl and Wayne Tiller. *The NURBS Book*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. ISBN: 978-3-642-97385-7. DOI: 10.1007/978-3-642-97385-7_1. URL: https://doi.org/10.1007/978-3-642-97385-7_1.

[64] Jürgen Gräsel et al. "A Full Parametric Model for Turbomachinery Blade Design and Optimisation". In: Jan. 2004. DOI: 10.1115/DETC2004-57467.

[65] Hengtao Shi. "A Parametric Blade Design Method for High-Speed Axial Compressor". In: *Aerospace* 8.9 (2021). ISSN: 2226-4310. DOI: 10.3390/aerospace8090271. URL: https://www.mdpi.com/2226-4310/8/9/271.

[66] Christian Voß. "BladeGen Manual". In: *DLR - Institute for Propulsion Technology* (Jan. 2023), p. 47.

[67] Christian Voß and Eberhard Nicke. "Automatische Optimierung von Verdichterstufen". In: *Technische Informationsbibliothek Hannover* (Sept. 2008).

[68] C.D. Woodward. "Skinning techniques for interactive B-spline surface interpolation". In: *Computer-Aided Design* 20.8 (1988), pp. 441–451. ISSN: 0010-4485. DOI: https://doi.org/10.1016/0010-

4485(88)90002-4. URL: https://www.sciencedirect.com/science/article/pii/0010448588900024.

[69] Jordan Ko. "Numerical Modelling of Highly Swirling Flows in a Cylindrical Through-Flow Hydrocyclone". In: (Jan. 2005).

[70] T. I. -P. Shih et al. "Algebraic grid generation for complex geometries". In: *International Journal for Numerical Methods in Fluids* 13 (1991), pp. 1–31. DOI: 10.1002/fld.1650130102.

[71] Peter Eiseman and Robert Smith. "Applications of algebraic grid generation". In: (Apr. 1990).

[72] Tamas Varady, Alyn Rockwood, and Péter Salvi. "Transfinite surface interpolation over irregular -sided domains". In: *Computer-Aided Design* 43 (Nov. 2011), pp. 1330–1340. DOI: 10.1016/j.cad.2011.08.028.

[73] Péter Salvi, Istv'an A. Kov'acs, and Tam'as V'arady. "Computationally efficient transfinite patches with fullness control". In: *ArXiv* abs/2002.11212 (2020).

[74] Ahmad Rushdi et al. "All-quad meshing without cleanup". In: *Computer-Aided Design* 85 (Aug. 2016). DOI: 10.1016/j.cad.2016.07.009.

[75] Rao V. Garimella, Mikhail J. Shashkov, and Patrick M. Knupp. "Triangular and quadrilateral surface mesh quality optimization using local parametrization". In: *Computer Methods in Applied Mechanics and Engineering* 193.9 (2004), pp. 913–928. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2003.08.004. URL: https://www.sciencedirect.com/science/article/pii/S0045782503005942.

[76] Joel Daniels et al. "Quadrilateral Mesh Simplification". In: *ACM Trans. Graph.* 27.5 (2008). ISSN: 0730-0301. DOI: 10.1145/1409060.1409101. URL: https://doi.org/10.1145/1409060.1409101.

[77] Bret Anderson, Steven Benzley, and Steven Owen. "Automatic All Quadrilateral Mesh Adaption through Refinement and Coarsening". In: Jan. 2009, pp. 557–574. ISBN: 978-3-642-04318-5. DOI: 10.1007/978-3-642-04319-2_32.

[78] Christophe Geuzaine and Jean-François Remacle. "Gmsh: A three-dimensional finite element mesh generator with built-in pre-and post-processing facilities". In: 79 (Nov. 2008).

[79] R A Dwyer. "A Simple Divide-and-Conquer Algorithm for Computing Delaunay Triangulations in O(n Log Log n) Expected Time". In: *Proceedings of the Second Annual Symposium on Computational Geometry*. Yorktown Heights, New York, USA: Association for Computing Machinery, 1986, pp. 276–284. ISBN: 0897911946. DOI: 10.1145/10515.10545. URL: https://doi.org/10.1145/10515.10545.

[80] S. Rebay. "Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer-Watson Algorithm". In: *Journal of Computational Physics* 106.1 (1993), pp. 125–138. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1993.1097. URL: https://www.sciencedirect.com/science/article/pii/S0021999183710971.

[81] J. D. Bernal and J. L. Finney. "Random close-packed hard-sphere model. II. Geometry of random packing of hard spheres". In: *Discuss. Faraday Soc.* 43 (0 1967), pp. 62–69. DOI: 10.1039/DF9674300062. URL: http://dx.doi.org/10.1039/DF9674300062.

[82] Jean-François Remacle et al. "A Frontal Delaunay Quad Mesh Generator Using the Linf Norm". In: vol. 94. Jan. 2012, pp. 455–472. ISBN: 978-3-642-24733-0. DOI: 10.1007/978-3-642-24734-7_25.

[83] Jean-François Remacle et al. "Blossom-Quad: A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm". In: *International Journal for Numerical Methods in Engineering* 89 (Feb. 2012), pp. 1102–1119. DOI: 10.1002/nme.3279.

[84] Theodore N. Edelbaum. "Theory of Maxima and Minima". In: *Optimization Techniques*. Ed. by George Leitmann. Vol. 5. Mathematics in Science and Engineering. Elsevier, 1962, pp. 1–32. DOI: https://doi.org/10.1016/S0076-5392(08)62089-5. URL: https://www.sciencedirect.com/science/article/pii/S0076539208620895.

[85] Narendra Karmarkar. "A new polynomial-time algorithm for linear programming". In: *Combinatorica* 4 (1984), pp. 373–395.

[86] Jannik Häßy and Jens Schmeink. "Knowledge-Based Conceptual Design Methods for Geometry and Mass Estimation of Rubber Aero Engines". In: *33th Congress of the International Council of the Aeronautical Sciences*. 2022. URL: https://elib.dlr.de/192376/.

[87] Jannik Häßy et al. "Hybrid Surrogate-Based Rubber Engine Model for Aircraft Multidisciplinary Design Optimization". In: *AIAA AVIATION Forum 2020*. 2020. URL: https://elib.dlr.de/135385/.

[88] Stanislaus Reitenbach et al. "Design and Application of a Multi-Disciplinary Pre-Design Process for Novel Engine Concepts". In: *Journal of Engineering for Gas Turbines and Power* 141.1 (2018). URL: https://elib.dlr.de/126225/.

[89] Markus Schnoes and Eberhard Nicke. "A Database of Optimal Airfoils for Axial Compressor Throughflow Design". In: *ASME Journal of Turbomachinery* 139.5 (2017). Ed. by Hall Kenneth, pp. 051008–051008. URL: https://elib.dlr.de/110879/.

[90] Harold Youngren Mark Drela. *A User's Guide to MISES 2.63*. 2008.

[91] Markus Schnoes et al. "Strategies for Multi-Fidelity Optimization of Multi-Stage Compressors with Throughflow and 3D CFD". In: *ISABE 2019*. 2019. URL: https://elib.dlr.de/129457/.

[92] Kai Becker, Kathrin Heitkamp, and Edmund Kügeler. "Recent Progress In A Hybrid-Grid CFD Solver For Turbomachinery Flows". In: *V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010*. Ed. by J. C. F. Pereira, A. Sequeira, and J. M. C. Pereira. 2010. URL: https://elib.dlr.de/68938/.

[93] Guido Dhondt. *CalculiX CrunchiX USER's MANUAL*. Version 2.20. 2022.

[94] E.W. Collings R. Boyer and G. Welsch. *Materials Properties Handbook: Titanium Alloys*. ASM International, 1994. ISBN: 978-0-87170-481-8. URL: https://doi.org/10.1007/978-3-030-64708-7_10.

[95] INTES GmbH. *PERMAS User's Reference Manual*. Version 18.00.121. 2020.

[96] Steven G. Johnson. *The NLopt nonlinear-optimization package*. 2023. URL: http://github.com/stevengj/nlopt.

[97] Jean-Pierre Signoret and Alain Leroy. "Failure Mode, Effects (and Criticality) Analysis, FME(C)A". In: *Reliability Assessment of Safety and Production Systems: Analysis, Modelling, Calculations and Case Studies*. Springer International Publishing, 2021, pp. 165–172. ISBN: 978-3-030-64708-7. DOI: 10.1007/978-3-030-64708-7_10. URL: https://doi.org/10.1007/978-3-030-64708-7_10.

[98] Takuya Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[99] M. J. D. Powell. "A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation". In: *Advances in Optimization and Numerical Analysis*. Ed. by Susana Gomez and Jean-Pierre Hennart. Dordrecht: Springer Netherlands, 1994, pp. 51–67. ISBN: 978-94-015-8330-5. DOI: 10.1007/978-94-015-8330-5_4. URL: https://doi.org/10.1007/978-94-015-8330-5_4.

[100] M. J. D. Powell. "Direct search algorithms for optimization calculations". In: *Acta Numerica* 7 (1998), pp. 287–336. DOI: 10.1017/S0962492900002841.

[101] J. Eric. *Data Science Life Cycle: CRISP-DM and OSEMN frameworks*. URL: https://datarundown.com/data-science-life-cycle/.

[102] C. Wiggins H. Mason. *A Taxonomy of Data Science*. URL: http://dataists.com/2010/09/a-taxonomy-of-data-science/.

[103] Vinod Sharma. "A Study on Data Scaling Methods for Machine Learning". In: *International Journal for Global Academic & Scientific Research* 1 (Feb. 2022). DOI: 10.55938/ijgasr.v1i1.4.

[104] François Chollet et al. *Keras*. https://keras.io. 2015.

[105] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[106] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[107] Jose Luis Blanco and Pranjal Kumar Rai. *nanoflann: a C++ header-only fork of FLANN, a library for Nearest Neighbor (NN) with KD-trees*. https://github.com/jlblancoc/nanoflann. 2014.

[108] T. O. Hodson. "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not". In: 15 (2022), pp. 5481–5487. URL: https://doi.org/10.5194/gmd-15-5481-2022.

[109] Peter J. Huber. "Robust Estimation of a Location Parameter". In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73–101. DOI: 10.1214/aoms/1177703732. URL: https://doi.org/10.1214/aoms/1177703732.

[110] Justin S. Gray et al. "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization". In: *Structural and Multidisciplinary Optimization* 59.4 (Apr. 2019), pp. 1075–1104. DOI: 10.1007/s00158-019-02211-z.