

The Artificial Engineer: A Smart Holistic Framework for the Automated Transfer of Geometry to Analysis Models

Von der Fakultät für Maschinenbau, Elektro- und Energiesysteme
der Brandenburgischen Technischen Universität Cottbus–Senftenberg
zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

vorgelegt von
Dipl.-Ing.

Benjamin Spieß
geboren am 27.08.1991 in Weiden i. d. Opf.

Vorsitzender: Prof. Dr.-Ing. habil. Hon. Prof. (NUST) Dieter Bestle

1. Gutachter: Prof. Dr.-Ing. Klaus Höschler

2. Gutachter: Prof. Dr.-Ing. Marius Swoboda

Tag der mündlichen Prüfung: 18.09.2023

Acknowledgement

The completion of my doctoral dissertation marks a significant milestone of my academic career which I would have never achieved without the support and encouragement of many people. It is therefore my sincere pleasure to thank all those who have been part of this project. I am deeply thankful for the opportunity to bridge the worlds of engineering and computer science in my research. This interdisciplinary exploration has been a truly enriching experience.

First and foremost, I would like to express my deepest gratitude to my supervisors, Maren Fanter and Prof. Klaus Höschler. Their professional expertise, patience, and unwavering support have been invaluable to the successful completion of my thesis. Your inspiring advice helped me to rise above myself, and I am deeply grateful to you. Likewise, I would like to thank Prof. Marius Swoboda, for his valuable insights and constructive criticism as my second supervisor.

I also want to acknowledge the support and camaraderie of my colleagues, Chen-Xiang, Alexander, and Bennet. Your collaboration has been inspiring and motivating. Together we have overcome challenges and celebrated successes, and your support has been crucial for the success of my work.

To my family and my girlfriend, Annabell, I am really grateful for your continuous support, encouragement, and understanding. You picked me up during the intense phases of my work, gave me courage and were my strongest support. Your love and assistance gave me the energy I needed to persevere.

This thesis stands as a testament to the collective contributions of all those mentioned and countless others who have played a part in my academic and personal development.

Thank you for being a vital part of this transformative experience.

Benjamin Spieß

October, 2023

Abstract

The creation of adequate simulation models for complex assemblies is an extensive process that requires a lot of experience, and on the other hand involves a multitude of manual, tedious tasks. These are significant obstacles for improving the process performance and capabilities. The objective of this research is to develop methods which digitally imitate the way of thoughts of the engineer in the design process towards a digital system understanding and which support the automation of the involved manual workflow.

This thesis presents a strategy to translate engineering reasoning and actions to an equivalent in the computer domain. A cardinal step is to gain understanding of system arrangements, boundary conditions and its components. Based on this evaluation, the identification of assembly parts is forming the foundation for optimized process chains for the transfer to the analysis environment. Model complexity relates to computational effort, which in turn affects model capabilities and manageability. To achieve a satisfactory compromise of model quality and complexity, this transfer process is strongly dependent on the visual analysis, reasoning and manual implementation of skilled engineers.

The principle of translating engineering logics is pursued from the assembly system to its smallest parts. Component segmentation methods allow subdividing regions of interest into substructures which are assigned with a feature vector. This vector comprises metrics describing the substructures with regard to specific aspects and is the key decision point for subsequent steps as idealization, suitable Finite-Element modeling and ultimately building an analysis model. The created system database is continuously maintained and supports these process chains as well as the final setup of the assembly simulation model.

An automated workflow like this implies advantages for efficiency, but also creates opportunities for further use cases. This workflow has been exploited for generating a training data set from the different simulation variants as a basis to a knowledge representation imitating engineering experience. An algorithm from the graph neural network field is applied to this data set as a conceptual approach. The intention pursued in this concept is to model the learning progress about estimating the influence of modeling decisions on simulation results and quality.

This research proposes a holistic strategy and describes methods to achieve the objectives of decreasing manual effort, introducing an automated and geometry-based process and digitally replicating engineering experience by introducing a knowledge database.

Table of contents

List of Figures	iv
List of Tables	viii
Abbreviations	ix
Nomenclature	xii
1. Introduction	1
1.1. Motivation	1
1.2. Objectives and scope	3
1.3. Thesis structure	4
2. State-of-the-art	6
2.1. Aero engines	6
2.2. Design process	7
2.3. CAD	9
2.4. FEM	12
2.4.1. Overview	12
2.4.2. Element types	12
2.4.3. Modal analyses	15
2.5. CAE model generation	18
2.5.1. Simplification	18
2.5.2. Dimensional Reduction	27
2.5.3. Information integration	32
2.5.4. Assembly analysis	34
2.6. Knowledge representation	36
2.6.1. Knowledge base	36
2.6.2. Machine learning	38

Table of contents

3. Process overview	45
3.1. Scientific gap	45
3.2. Strategy	47
4. Recognition framework	49
4.1. Interfaces	50
4.2. Recognition	53
4.2.1. Bolts	55
4.2.2. Flanges	56
4.2.3. Bearings	59
4.2.4. Rotary Groups	61
4.2.5. Beam structures	63
4.2.6. Casings	64
4.3. Application	65
5. CAE transfer	71
5.1. Casings	72
5.1.1. Feature-based analysis	72
5.1.2. Silhouette-based analysis	74
5.1.3. Substructure analysis	83
5.1.4. Simplification	86
5.1.5. Simulation model setup	101
5.1.6. Application	103
5.2. Beam structures	115
5.2.1. Process	115
5.2.2. Application	116
5.3. Interface mapping	118
5.4. Assembly domain	121
5.4.1. Framework architecture	121
5.4.2. Application	124
6. Knowledge Representation	132
6.1. Objective	132
6.2. Data generation	133
6.2.1. Geometry	133
6.2.2. Simulation models	135
6.2.3. Data setup	139

Table of contents

6.3. Process development	140
6.3.1. Strategy	140
6.3.2. Data stream	142
6.4. Application	146
7. Conclusion	153
7.1. Summary	153
7.2. Outlook	155
A. Appendix	158
A.1. Algorithms	158
Bibliography	162

List of Figures

2.1. Scheme of the Brayton cycle	7
2.2. Modeling approaches (after [22])	10
2.3. Common continuum elements	13
2.4. Related fields	18
2.5. Different model representations	19
2.6. AAG process	21
2.7. Convex hull decomposition	23
2.8. Volume decomposition using concavity information	26
2.9. Duality of Voronoi diagram and Delaunay triangulation	28
2.10. Medial axis/object	29
2.11. Midsurface by face-pairs	29
2.12. Neural network scheme	40
2.13. Model capacity	41
2.14. Model complexity	42
3.1. Main elements of the design process	47
4.1. Framework fundamentals	50
4.2. Bounding box types	51
4.3. Interfaces	52
4.4. Contact face process	53
4.5. Schematic interface network	54
4.6. Process workflow	55
4.7. Bolt topology and graph	56
4.8. Flange types	57
4.9. Split flange merging	58
4.10. 2D schematic flange chain	59
4.11. Exemplary bearing layouts	60
4.12. Bearing graph	61
4.13. Rotary groups scheme	62

List of Figures

4.14. Beam structure analysis	63
4.15. M1 demo case	66
4.16. M2 application	67
4.17. M3 application	68
4.18. M4 application	69
4.19. Performance evaluation	70
5.1. Feature tree scheme	73
5.2. Segmentation interface scheme	74
5.3. Silhouette graph	76
5.4. Non-closed silhouettes	77
5.5. Contour extension process	78
5.6. Polygon investigation	80
5.7. Line / Spline mapping	80
5.8. Non-axisymmetric analysis	82
5.9. Model analysis	83
5.10. Interface distance evaluation	85
5.11. Beam evaluation	86
5.12. Representation approaches	88
5.13. Medial axis workflow	89
5.14. Branch analysis	90
5.15. Polygon adaption	92
5.16. Thickness interpolation	93
5.17. Imprint process	95
5.18. Approximated imprint process	95
5.19. Hybrid process	96
5.20. Shell representation	97
5.21. Beam representation	98
5.22. Proximity analysis	100
5.23. Database information scheme	103
5.24. C1 models	104
5.25. C1 results	105
5.26. C1 MAC evaluation	106
5.27. Evaluation of metrics of model C1	107
5.28. C2 models	108
5.29. C2 results	109

List of Figures

5.30. Evaluation of metrics of model C2	109
5.31. C3 models	110
5.32. C3 results	111
5.33. Evaluation of metrics of model C3	112
5.34. C4 models	113
5.35. C4 results	114
5.36. Evaluation of metrics of model C4	114
5.37. Process performance evaluation	115
5.38. B1 models	116
5.39. B1 results	117
5.40. Evaluation of metrics of model B1	117
5.41. B2 models	118
5.42. B2 results	118
5.43. Evaluation of metrics of model B2	119
5.44. Flange mapping	119
5.45. Assembly interface mapping	120
5.46. Beam interface mapping	121
5.47. Transfer framework process	122
5.48. A1 application	125
5.49. A2 application	126
5.50. A3 application	129
5.51. A4 application	130
5.52. Assembly process performance evaluation	131
6.1. Geometry generation workflow	134
6.2. Model pool	136
6.3. Simulation database process	137
6.4. Evaluation of database formats	139
6.5. Database structure scheme	140
6.6. Data structures	141
6.7. Eigenfrequency deviation values	143
6.8. Graph database scheme	145
6.9. Data extraction steps	145
6.10. Feature vector studies	147
6.11. Hyper-parameter study results	148
6.12. Confusion matrix	150

6.13. Dataset evaluation 151

List of Tables

4.1. Application metrics	68
5.1. Example models and process performance	82
5.2. Substructure feature vector	87
6.1. Point distribution scheme	144
6.2. Objective value category	144
6.3. Accuracy results	149
6.4. F_1 score results	150

Abbreviations

AAG	Attributed Adjacency Graph
AGM	Apriori-based Graph Mining
AI	Artificial Intelligence
APCA	Adaptive Piecewise Constant Approximation
API	Application Programming Interface
ASME	American Society of Mechanical Engineering
B-Rep	Boundary Representation
BBox	Basic Bounding Box
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing
CAP	Computer-Aided Planning
CAPP	Computer-Aided Process Planning
CAX	Computer-Aided Everything
CBox	Cylindrical Bounding Box
CoG	Center of Gravity
CSG	Constructive Solid Geometry
CSV	Comma-Separated Values
DFS	Depth-First Search
DGCNN	Deep Graph Convolutional Neural Network
DL	Deep Learning
DMU	Digital Mock-Up
DoE	Design of Experiments
DOF	Degrees-Of-Freedom

Abbreviations

ES	Expert System
FAG	Face Adjacency Graph
FE	Finite-Element
FEA	Finite Element Analysis
FEM	Finite Element Method
FN	False Negative
FP	False Positive
FR	Feature Recognition
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
GUI	Graphical User Interface
HDF	Hierarchical Data Format
HGP-SL	Hierarchical Graph Pooling with Structure Learning
HLP	Human-Level Performance
HLT	High Level Topology
HPC	High-Pressure Compressor
HPT	High-Pressure Turbine
KB	Knowledge Base
KBE	Knowledge-Based Engineering
KBES	Knowledge-Based Engineering System
KBS	Knowledge-Based System
KF	Knowledge Fusion
LOD	Level-Of-Detail
LPC	Low-Pressure Compressor
LPT	Low-Pressure Turbine
LR	Learning Rate
LSTM	Long Short-Term Memory
MA	Medial Axis
MAAG	Multi-Attributed Adjacency Graph

Abbreviations

MAAG	Multi-Attributed Adjacency Matrix
MAC	Modal Assurance Criterion
MAS	Medial Axis System
ML	Machine Learning
MPC	Multi-Point Constraint
NC	Numeric control
NN	Neural Network
OBox	Oriented Bounding Box
oLHS	Optimal Latin-Hypercube Sampling
PIP	Point-In-Polygon
PLM	Product Lifecycle Management
RAG	Region Adjacency Graph
RBE	Rigid Body Element
RDP	Ramer-Douglas-Peucker
SDF	Shape Diameter Function
SQL	Structured Query Language
TN	True Negative
TP	True Positive
XML	Extensible Markup Language

Nomenclature

$ \mathcal{A} $	Size of set \mathcal{A}
$\mathbf{A}, \mathbf{B}, \mathbf{C} \dots$	Notation for matrices
$\mathbf{a}, \mathbf{b}, \mathbf{c} \dots$	Notation for vectors
$\mathcal{A}, \mathcal{B}, \mathcal{C} \dots$	Notation for sets

1. Introduction

1.1. Motivation

Data is a precious thing and will last longer than the systems themselves.

Tim Berners-Lee

The process associated with the term product development typically refers to all stages which are involved in introducing a new product to an existing market field, redesigning a current product or introducing the existing product in a new market. The initiative for this process is often a market need or a problem which the designated product is intended to address. In many cases, new emerging technologies, customer requirements, boundary conditions as environmental targets or simply competitive reasons are raising these needs or problems.

The major objectives which commonly accompany and direct the product development process are often generalized to the strongly dependent values: quality, time and cost. A common way to describe or symbolize their relation is by using a triangle, or iron triangle as described in [1]. In analogy to this representation, optimizing one objective, so moving to one corner, affects the other values, or increases the distance to them respectively. Technological progress as well as the mentioned boundary conditions are likely to result in a continuously increasing system or product complexity [2, pp. 47 sqq.]. This complexity influences the optimization space in such a way that it often results in narrow margins for improvement. On the other hand, the increasing competition is also pushing the boundaries in this context and makes an economical balance between the values difficult to maintain.

For generating a challenge-meeting product under these circumstances, the introduction of new ideas and concepts or the integration of new efficient and solution-oriented methods is often inevitable, and so these opportunities have to be evaluated regarding benefits for the development cycle. Automation plays an important part in fields like Industry 4.0 [3] and represents a method which can help to overcome existing limitations. Automating processes can on the one hand reduce the required development time, and

on the other hand decrease manual work resources, thus depict potential cost savings. Process automation also implies advantages for consistency by reducing the probability for manual errors, which represents a considerable profit for future developments. Besides this objective-related perspective, further reaching benefits gained by automation can be identified in for example enabling a faster response to design suggestions or changes. High-volume design exploration steps are skipped due to restricting conditions as for cost and time reasons in many cases. Despite this, the design exploration is an important aspect for product development and optimization which can be significantly supported by automated methods. Also with regard to technologies which are likely to play a decisive part in the industry of the future like Artificial Intelligence (AI), automation is part of the principal set of tools. Data generation, data stream setup and connecting various data silos are aspects which are strongly reliant on such automated processes.

In the engineering domain, the product development cycle often refers to a physical product which shall meet the imposed challenges. Many engineering disciplines make use of state-of-the-art computer technologies as Computer-Aided Design (CAD) for geometry design and various types of CAx methods for analysis within this product development process. In most of the cases, the geometry created with the CAD methods serves as starting point for the downstream analysis and validation steps. The Finite Element Method (FEM) technique has proven itself for the simulation aspect, especially in the field of structural mechanics, therefore is a commonly utilized approach in the product development cycle [4, pp. 242 sqq.]. The generated FEM simulation model is then to be enriched by appropriate boundary conditions and simulation properties to reconstruct the real system environment and behavior. Suitable and validated virtual simulation models are then supporting the development process with their ability to reproduce the real behavior and by this are sufficiently accurate to virtually elaborate different design iterations and modifications.

However, depending on the complexity of the system, the modeling and simulation setup can take up a large amount of the overall design iteration, more precisely up to an estimated percentage of 50% of the process as described by Klein in [5, pp. 7 sqq.]. With respect to the fundamental development objectives, reducing the time spent in the design process can significantly create room for process, hence product, optimization [6].

Computational resources as well as time required for conducting the simulation can represent other limiting factors for more complicated models. Especially in early design phases, in which the presented work is situated, these aspects are in contrast to desired capabilities as quick design responses and design explorations. For this purpose, simulation models are often modified, idealized and simplified to reduce their complexity while aiming

for a model that is accurate enough for the intended requirements [7, pp. 16 sqq.]. These simplified representations, however, require considerable additional effort for the setup and generation which can take two weeks for a simplified shell model [8], several months for some aero engine components [9] or even more than one year for a full jet engine [10]. Besides the manual effort, many modification steps and assumptions require skilled analysts with a well-founded engineering judgment, knowledge and high expertise to achieve a satisfactory model quality [4, 9–11].

The need for simplified models and the disadvantages coming along with their preparation emphasize the demand for automation in this context, and, furthermore, automation can also be useful for rebuilding the required expertise within a data-driven strategy.

These described circumstances and problems have led to the definition of this research which focuses on the development of strategies and methods which shall help to overcome existing limitations and improve efficiency in the product development process and help to shift the focus from tedious manual work to more innovative and solution-oriented work.

1.2. Objectives and scope

The efficient development of appropriate simulation models is a crucial part of the design process, especially in preliminary development phases. As previously described, automation is a powerful tool set which is able to overcome existing obstacles and limitations in the development process, thus significantly boost process suitability and efficiency and facilitate design iterations and explorations.

Considering a CAD geometry as starting point, the guiding objective of this work is to develop strategies, methods and techniques to automate the transfer of geometry models to adequate simulation models for structural analyses. With this research being funded by a cooperation project with an aero engine manufacturer, its scope is aligned to aero engine structures. These aero engines typically represent complex systems. Consequently, the associated simulations definitely benefit from simplified representations when used in suitable scenarios to promote the improvements and possibilities mentioned in the previous chapter 1.1. In lights of these aspects, additional focus of this research work is on automated solutions for introducing simplification assumptions in appropriate context and conditions.

Another important building block of system simulation models are boundary conditions. One type of boundary conditions serve the purpose of rebuilding load paths, assembling and maintaining a connection between different components, or constraining the system

with regards to an environment. These simulation entities are also often approximated as simplified version of more detailed component connections. Another type is the category of boundary conditions which introduce different types of external influence factors to a simulation model, as for example temperature or pressure fields. In the scope of this work, the focus is set on the first type of boundary condition, and their automated integration respectively, in order to build a consistent simulation model which can in turn depict the starting point for various downstream applications.

With the FEM approach being state-of-the-art in structural aerospace design development, this method is also representing the basis for the generation of the simulation model. Detailed analyses of various Finite-Element (FE) types and meshing algorithms are, however, beyond the scope of this work since the emphasis is set on the geometric model analysis and preparation. So, the FE entities and algorithms integrated in commercial software packages are considered as state-of-the-art and serve as basis for the FEM model generation.

1.3. Thesis structure

The path of pursuing these goals within this research work has been organized in five major chapters.

The present domain, current state-of-the-art technologies as well as related research work are described in chapter 2. A short introduction to the aero engine basics is done to provide an overview of common principles and architectures, which play a role later in the work. The fundamentals of CAD and FEM technologies which represent the basis for the developed processes are presented afterwards. Common development strategies for system development and simulations are described which guide the main approach concept within this work. Researches which refer to model simplification, automation of model preparation and simulation generation as well as possibilities to integrate engineering knowledge are representing the reference state-of-the-art for the objectives described previously.

Chapter 3 describes the major strategy that has been developed for achieving the imposed challenges. As described by Hamri et al. in [12], an automated direct conversion of a geometric system to a reasonable simulation model can be either very complicated or even not possible in general. For this reason, a cognitive aspect is introduced because it is playing a crucial role within the general development cycle, thus as well for the automation of this process.

This recognition aspect and its modeling and integration into an automated process

is presented in the following chapter 4. Similar to an engineer who analyses a present assembly regarding boundary conditions, component types and regions which could be suitable for simplification, a smart recognition framework is digitally replicating these steps and producing information for the next tasks in the development process.

The geometry which is enriched by this information depicts the input to the next segment in the process chain, which is dedicated to model simplification approaches for suitable domains and the transfer of geometries to their corresponding simulation representations. Chapter 5 focuses on methods and approaches to automate this design tasks, the subsequent assembly from the transferred geometries and presents applications of the developed techniques.

Automated processes depict advantages for process efficiency, but open up also paths to other types of analyses. The last chapter, chapter 6, describes a conceptual and data-driven approach to use the developed transfer strategy to digitally replicate and mimic the engineering expertise which is playing a decisive part in the manual model transfer and idealization process.

Finally, a conclusion is drawn about the achieved objectives, developed strategies and approaches and also potential next steps are outlined.

2. State-of-the-art

2.1. Aero engines

The aero engine context is defining a guiding context for the objectives and scope of this work. The emphasis of the objectives is set on the principal development process in the structural mechanics domain. For this reason, a detailed introduction to for example aero engine related turbomachinery topics is neglected in the scope of this work. Nevertheless, a set of the developed methods and logics are based on the fundamental aero engine architecture understanding, analogously to the aero engine development engineer who is building upon his theoretical and practical background. In this regard, this section serves the purpose to provide an overview of basic aero engine principles and resulting systems.

Aero engines use the fundamental principle of increasing the velocity of the air flow from inlet to outlet to produce the desired propulsion. The main contributors to the generated impulse are the mass and the velocity. By generating a positive impulse difference between the intake and the exhaust, the aero engine is producing thrust [13]. In the present case, the mass at the inlet and the exhaust are approximately identical, so the impulse difference has to be created by accelerating the air flow. For this purpose, aero engines make use of the working principle called Brayton cycle, which is depicted in figure 2.1. Each of the major aero engine components is fulfilling its task in this scheme. First, the airflow is guided through the intake to the compressor which is increasing the pressure of the airflow stage by stage using the laws of turbomachinery. This increase in pressure and also enthalpy is represented by the curve from point 1 to 2. The compressed air is then mixed with fuel and ignited, what in turn increases its temperature, and therefore its energy, so the current status moves on to point 3. This energy is extracted by the turbine afterwards, resulting in a momentum to the shaft and a working fluid with high kinetic energy. The nozzle serves the purpose to convert the remaining energy into velocity or impulse in order to generate thrust and to maintain the pressure level. The momentum at the shaft is received by the compressor, which is requiring energy input to perform [13].

A further development of the basic aero engine concept is the turbofan aero engine, which uses two or three concentric shafts, each rotating with another speed. Aero

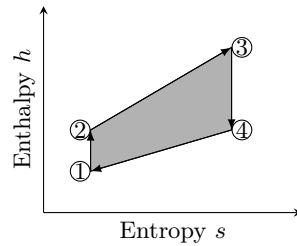


Figure 2.1.: Scheme of the Brayton cycle [13, p. 36]

engines in this form can be subdivided into high- and low-pressure areas, and in case of a three-spool engine, also into a third intermediate pressure section. In these engines, the High-Pressure Turbine (HPT) delivers the power for the High-Pressure Compressor (HPC) via the high-pressure shaft and the Low-Pressure Turbine (LPT) in turn drives the Low-Pressure Compressor (LPC). The company Rolls-Royce for example also introduces a third shaft for the mentioned intermediate-pressure category. In this approach, the fan is driven exclusively by LPT, which serves as a pure work turbine. The introduction of multiple shafts separates the engine into systems which can have different rotational speeds, which in turn implies improved capabilities to adapt to the respective optimum efficiency regions of the components.

These rotary parts are connected via interfaces to the opposing structural system which guides and contains the gas path. Due to fundamental turbomachinery working principles, a majority of the opposing components shares nearly axisymmetric nature and often resembles casing structures which enclose this gas path. These structural parts comprise for example sockets and joints for stationary vanes and auxiliary installations and also the interfaces to mounting systems which connect the core engine to the surrounding airplane structures.

2.2. Design process

Aero engines represent a highly complex systems with transient and non-linear interactions of a multitude of engineering disciplines. This leads to the fundamental question of an adequate approach to design, development and optimization of such assemblies. Many researchers over the years tackled similar problems and suggested various approaches where some of them are defined standards nowadays. A good overview on the evolution of design methodologies is given in [14, pp. 20 sqq.]. With the rise of computational power and abilities, new extended design possibilities emerge and lead to new forms of product development [4, pp. 13 sqq.]. However, this can be seen as a race between possibilities

and methodologies for enabling technology and the demand created by the strive for continuous innovation of products and processes which become more and more complex over time. This in turn also leads to the world community growing more competitive implying to be able to deal with larger and more complex systems while keeping the design process as efficient as possible [2, pp. 47 sqq.]. For this reason, an adequate development approach to cope with multi-disciplinary systems and interactions is a mandatory aspect.

Specifications build the basis and guidance for the development of for example aero engine systems. These are commonly derived from aircraft manufacturers stating their requirements and from a market research to guarantee competitiveness. Preliminary structural, thermodynamic and aerodynamic design studies are often the next steps to lay out the basic aero engine process. The developed system is then included within the mechanical design framework, which is depicting the main interface for aero engine related load case simulations as well as interfaces to the aircraft manufacturer. However, this process contains a multitude of iterative steps and cycles to cope with multi-disciplinary effects [15, 16].

The desire to gain a fundamental engineering understanding of the system is imminent and also often more important in early design phases than aiming directly for increasing the simulation complexity. Despite rising accessible computational power, the effort for high-fidelity multi-disciplinary studies still depicts a substantial obstacle for these early system analyses and adequate response and iteration times. For this reason, a common approach in the field of multi-disciplinary optimization is the hierarchical decomposition into subsystems [17]. The definition of interfaces between subsystems is playing a significant role in this process. Properly defined interfaces allow system-wise individual simulations. Consequently, these subsystems can be investigated by domain experts with higher-fidelity models and the gained information can be redirected to the whole system iteration loop [2, 18].

This train of thoughts is also commonly pursued in aero engine development. Transient and complex maneuvers and load cases, like for example fan-blade off, limit and restrict the use of high-fidelity models for simulation purposes and thus, necessitate simplified systems. As previously mentioned, the focus is set on lower complexity system models with adequate interfaces for system-wise detail analyses.

Starting from this strategic point, the next topic is the setup of these simulation models. Computer-Aided Everything (CAx) applications became more and more powerful and suitable for an extending field in engineering throughout the years. Especially CAD models have strengthened their position as global basis for various analyses in the product development cycle.

2.3. CAD

CAD systems are situated in the wider field of CAx tools which have been part of research and development works since 1940 [4, p. 6]. An historical overview of the development and more information about CAx systems for further reference is given in literature as [4, 19]. A significant driver for CAx systems has been the use of computers for the development of products in the field of Numeric control (NC), Computer-Aided Planning (CAP) and Computer-Aided Manufacturing (CAM). Apart from that, CAD systems allowed also a more robust and reasonable setup for simulations in the field Computer-Aided Engineering (CAE), especially for the FEM domain. However, today's core competence as tool for product design has first emerged afterwards [4, p. 9]. All these steps positioned CAD as a major port for engineering disciplines of various kinds like design, manufacturing, simulations and even for economic evaluations.

Especially 3D modeling has evolved in the course of the years and is the state-of-the-art for the majority of design processes today. The developed 3D model serves as consistent basis for all work steps in the development cycle. Encarnacao et al. [20] and Vajna et al. [4] summarize the main approaches for 3D modeling to wireframe, surface, volume and voxel modeling. Due to their similarity to the real geometry, volume models have the most wide-spread range of applications. The VDI 2209 guideline [21] describes volume models as topological closed combination of surfaces which contain apart from information about the object hull or skin also information about the volume.

Over the last years, several methods and kernels have been developed to build these 3D models. The most common ones are the Constructive Solid Geometry (CSG), see figure 2.2b, and the Boundary Representation (B-Rep), see figure 2.2c, techniques which have lead to further hybrid approaches.

The CSG makes use of simple and mathematically defined geometric shapes, often referred to as primitives, and combines them using boolean operators to build 3D models [19, pp. 85 sqq.]. Exemplary shapes are cuboids, cylinders, prisms, pyramids, spheres and cones. Modern kernels also allow for more complicated primitives like free-form shapes and thus increase possible complexity. The final CSG tree, see figure 2.2b, contains all logical and chronological operations and connections while the leafs depict used primitives [4, pp. 179 sqq.]. Besides advantages like mathematical correctness, this technique entails also disadvantages, especially for product development and design. Due to the method-related restrictions, the introduction of small changes especially in incremental design iterations can become too unhandy for industrial use [22, p. 5].

The majority of commercial CAD software today is based on a kernel using the boundary

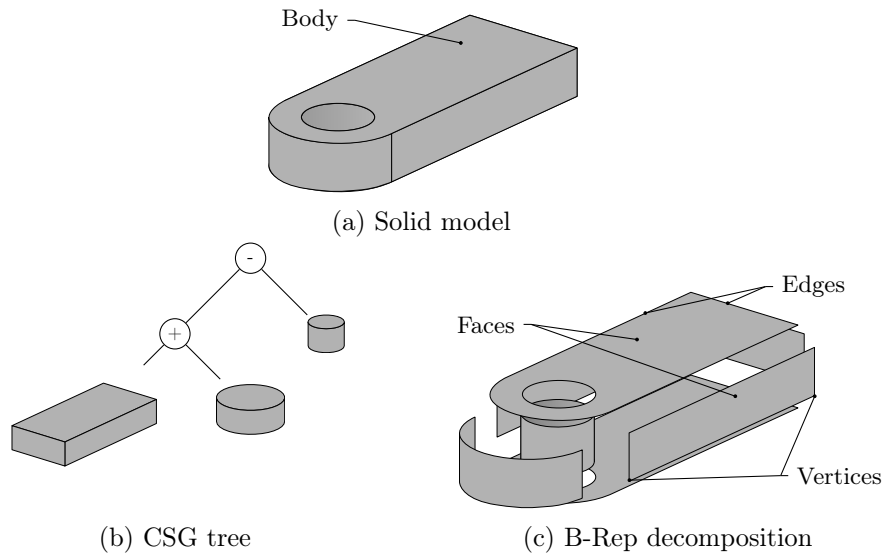


Figure 2.2.: Modeling approaches (after [22])

representation technique. This has been introduced and studied for CAD purposes by Braid in [22, p. 11, 23]. The volume model is constructed in this case by combining topological components like a set of surfaces which describe the model boundaries, see figure 2.2c. By means of conglomerating the surfaces, the entities are then bounded by trimming curves, which in turn result in objects as edges and vertices. On the basis of present surface information, the kernel is able to distinguish between the inside and the outside of the 3D model. Along with this comes information about the relation and adjacency of the topological entities, which can be used for further purposes. For building and modifying the model, specific Euler operators come into place to preserve the consistency of the topology of the solid [22, p. 11].

However, a major disadvantage is that in traditional B-Rep modeling the information is limited to the final state and shape. Thus, a morphological description of the volume model is not available [24, pp.11 sqq.]. The approach of storing the history of the B-Rep operations in a chronological tree within the CAD software has paved the way for more opportunities. In this regard, the B-Rep method involves a more flexible and also richer set of potential operations, so becoming a more sophisticated design and development toolkit. Compared to the CSG, each geometric entity is associated with geometric information which can be useful for later analyses and modifications. For a more detailed overview of basic modeling techniques refer to [19, 22, 23, 25, 26].

The idea of the history tree has led to the basic concept of so-called features. The concept

of feature-based has originated from manufacturing planning and NC programming [4, p. 194]. In the course of the years, a lot of different definitions for the term feature have been established. Shahin et al. [27] present a survey of these along with an overview of the history, the methodology and a collection of research articles of feature-based modeling in his work. The definition that fits the scope of this thesis is that “features encapsulate the engineering significance of portions of the geometry of a part or assembly, and, as such, are important in product design, product definition, and reasoning, for a variety of applications” [28]. With regard to CAD software, features are methods to ease the construction process, thus reduce design lead-time, by a predefined collection of CAD operations. Exemplary most-common features are extrusions, revolves, blends, chamfers and holes. The history of the applied features is stored afterwards in the internal feature tree and allows access and modification also later on via the software interface.

Another technique that has come along with the rise of CAD tools and applications is the parametric modeling [21]. In parametric systems, data structures can be controlled by parameters in order to change for example geometric dimensions or relations. This new principle brings along several additional benefits for time savings regarding modifications, reduce time for model creation, especially in case of design series development and thus also for parametric design studies [4, pp. 184 sqq.]. However, the topic of parametrization can become arbitrarily complex especially in larger assemblies due to increasing number of parameter relations and interactions. This emphasizes the requirement of a structured and methodological approach for an adequate parameterized model.

With a focus on strategies to build 3D models, Shah et al. [29] divides them into two major categories. On the one hand, a destructive approach with features involves volume subtracting operations from a base model in order to introduce further details and features. The other technique implies starting without the necessity of a base model and uses synthetic or constructive operations to build the model thereof [27]. The synthetic approach has proven to be more suitable for feature-based and parametric modeling for design studies.

The designed 3D models serve then as starting point for subsequent studies like structural mechanic FEM simulations. To transfer the model to the FEM system, the VDI guideline 2209 [21] distinguishes between two different variants. Either the FEM system is integrated in the CAD system or in case of different software being used, a neutral file format has to be used. For the second case, however, design and simulation problems due to lacking robustness and quality of the interfaces are quite common [4, pp. 263 sqq.].

2.4. FEM

2.4.1. Overview

In engineering fields, the task of analyzing a system behavior or predict the influence of loads on products is one of the most basic ones. Technical progress leads to rising complexity in systems and thus, the portion of problems which can be solved analytically has decreased [7, p. 3]. For this reason, numerical methods have made their way into this field and became standard for a multitude of applications. Especially FEM procedures have proven their benefits for engineering analysis [30, p. 1, 4, pp. 242 sqq.].

The name-giving objective of this methodology is to discretize complicated problems or problem domains into small and manageable units, the so-called finite elements [7, pp. 11 sqq.]. Depending on element and solution type, different functions and solver schemes are associated with these elements, which shall represent their properties. Afterwards, the discrete units are assembled under consideration of specific boundary conditions to approximate the complete system solution.

The governing equations for the element itself are derived from their physical relation. In structural mechanics, for example, the principle of virtual displacement is an approach to approximate the differential equations with equivalent discrete equations in order to develop a numeric solution [4, pp. 243 sqq.]. With different types of physical problems come different types of governing equations and conditions and thus, also different types of elements. Bathe [30, pp. 199 sqq.] summarizes the most common to truss, beam, plane stress, plane strain, axisymmetric, plate bending, thin and thick shell and general three-dimensional problems. Most of the structural problems could be defined as three-dimensional in this regard. However, the associated equations come along with a higher degree of complexity than in lower dimensional scenarios, thus leading to a more expensive solution regarding computer resources and computational time [30, p. 4].

2.4.2. Element types

Especially if computational complexity is a driving factor, the different categories of elements have their advantages and disadvantages of their own for specific problems.

One-dimensional elements like beam or truss elements depict the most fundamental element type, see figure 2.3a. The most common underlying theory for this type is based on the Euler-Bernoulli or Timoshenko approach as described further in [31] or [5, pp. 57 sqq., pp. 122 sqq.]. On the basis of these theories, the element shape functions can be directly derived from the analytical solution and thus provide high-quality results

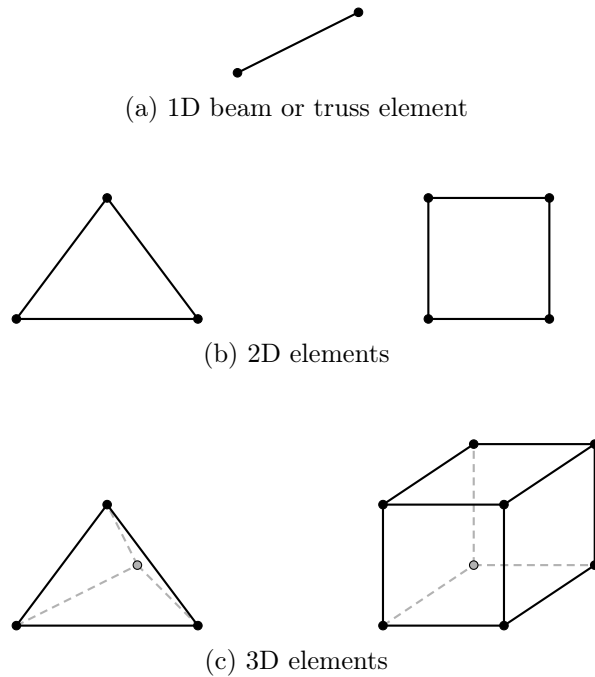


Figure 2.3.: Common continuum elements (after [30])

[4, pp. 273 sqq.]. Beam elements can consequently imply both accurate and low-cost implementations for suitable problems.

Increasing the dimensionality to the two-dimensional domain leads to plane and shell elements, see figure 2.3b. The fundamentals providing the approaches for these element types are based on the Kirchhoff or Reissner-Mindlin theories [5, pp. 140-168]. Its basic proposition is that the structure itself is thin in one dimension [30, p. 200], so recommend them mainly for likewise domains. With their lower dimensionality, they offer notable benefits in terms of lower complexity and demand of resources compared to three-dimensional volume elements.

Volume elements, figure 2.3c, are more complex units due to a higher number of degrees of freedom, the associated assumptions and shape functions. On the other hand, their 3D nature implies a certain degree of genericness which allows a wide-spread use in structural analyses. Discretizing a volume model by for example tetrahedral finite volume elements is comparably straightforward and does not necessitate further preparation effort in most of the cases. Along with this advantage comes the disadvantage that the use of 3D finite elements can lead quickly to undesirably computationally expensive models [4, p. 273]. Hexahedral volume elements allow in specific scenarios to discretize a given shape in a more efficient way, thus can be used as a means to mitigate this issue of extensive

computational demand. The more complex the given geometry is, however, the more effort is required to prepare this model for a hexahedral meshing operation. This often leads to the decision on the tetrahedral option.

Especially for thin structures, it is recommended to use more than one volume element through thickness to obtain reasonable simulation results [32, 33] which is why the commercial software Siemens NX comes with a setting for its meshing algorithm which ensures at least two elements over the thickness [34]. This in turn raises also the overall number of elements, thus degrees of freedom. In such cases, shell elements can profit from their advantage of producing adequate results while maintaining a lower simulation complexity. Nevertheless, the disadvantage of shell element application is that prior model preparation and idealization is required. A CAD surface domain for the 2D discretization has to be derived from the original 3D geometry first in case an integrated CAD-CAE path is pursued. A similar situation is present for 1D beam modeling, which on the one hand provides both quality and performance for suitable models, and on the other hand requires a prior extraction and definition of cross-section properties for beam units.

Apart from the fundamental element categories, two additional FE settings can have a significant effect on simulation results, thus quality. On the one hand, the element size, thus related to the number of elements in a mesh, is an important property to build accurate CAE models. More elements imply a better adaption to the fundamental geometry and so a more discretized and detailed result. On the other hand, increasing the degree of shape functions basis for the element formulation can support the reproduction of more complex phenomena. Both factors are often referred to as p and h method which are used for proving convergence in engineering simulations [4, pp. 275 sqq.].

The most common element formulations have been adapted and implemented in commercial FEM software. Klein presents an overview and comparison of the different FE solver kernels and software packages in [5, p. 15]. These tools are using CAD geometry as a starting point for the mesh creation. The process is structured as follows: the software receives input from a manual selection of appropriate geometric entities for suitable FE element types, e.g. solid bodies for a 3D volume element mesh. Afterwards, various internal algorithms guide the automated generation of FE meshes while considering specified mesh parameter and FE properties.

After having set up suitable FE models and meshes, the connection or interpolation of nodes and elements is another substantial step. For this reason, FE software provides a variety of options to rebuild connections or links and load extraction or application interfaces. One of those is the so-called Multi-Point Constraint (MPC) which allows the interpolation of arbitrary nodes, so Degrees-Of-Freedom (DOF), with a specified

formula. Rigid Body Element (RBE) is a subtype of those MPC elements which is often used for these interpolation tasks. Another type of connection which is often used in the NASTRAN domain is the CBUSH element, which represents a simple spring-damper element with specified parameter. A combination of both RBE and CBUSH types is often found in simplified and abstracted representations of bolt connections, for example. Apart from these, a more recent and modern approach for connections is the gluing technique of the NASTRAN kernel, which depicts a simple and effective method for joining dissimilar meshes [35, ch. 20]. Its principle bases on contact calculation including multiple iterations to optimize the connection simulation result. More information about gluing techniques can be found in [36, pp. 379-385].

However, the concrete FE theory is not central to this work and its implementation in current commercial software is considered as state-of-the-art. Nevertheless, a basic overview of exemplary FEM elements has been given as basis for subsequent logics and processes. For more information about the detailed FE history, theory and implementation reference is given to literature as [5, 30, 37–40].

In summary, each element category and type has its advantages and disadvantages. In case of suitable situations, the most significant disadvantage of elements of lower dimensionality is the required model preparation effort. According to Bathe [30], the aim is to achieve an effective model, either mathematical or physical, which yields the required response with a sufficient accuracy at the least possible cost. Also Michael et al. [7, p. 16] recommend to aim for a compromise of model accuracy and complexity, so for a model just accurate enough for its needs. Gasch further states in [37, p. 545] that the disk space increases with the square and the calculation time for an exemplary eigenvalue analysis rises with the fourth power of the number of degrees of freedom and thus emphasizes the advantages of optimized models. Consequently, especially complex and large assemblies necessitate effective simulation models for the related analyses. In early design phases, the ability to respond quickly to design changes and to create predictions with a reasonable pace is an important factor for understanding assembly behavior and optimizing the system.

2.4.3. Modal analyses

After having built a FE model, its suitability and quality regarding given objectives in the present environment has to be evaluated to allow reasonable conclusions. This step is commonly associated with the field of model validation or verification. The aim is to compare the digital model to the real model behavior to confirm the correctness of the

prediction capabilities of the model within given simulation paths. This field of model validation is of importance especially in times of increasing computational technologies and also for application cases as virtual testing, and consequently is comprising a vast amount of different techniques, strategies and approaches. A common guideline pursued in the validation and verification field has been published by the American Society of Mechanical Engineering (ASME) in [41]. This field in general is quite extensive and not part of the main objectives of this work, but basic methodologies thereof have been utilized for general model evaluations.

A simple and straight-forward approach to model validation is for example to apply a similar load on two models that are to be compared and compare the resulting model displacements at distinct locations to investigate the quality of the model matching. Another application for model validation would be for example the comparison of a virtual to a real physical model. However, this approach in general is likely to target too specific model or load path related properties and thus does not depict a reasonable validation foundation for general conclusions in more complicated scenarios.

In this context, an approach which is often referred to as modal validation can depict a more suitable approach to produce more general statements about the model matching capabilities. As described by Ewin in [42], modal analysis or testing can play a significant role in the correlation process of experimental and theoretical results and help to identify the causes for discrepancies. One of further applications is the so-called substructure modeling which aims for the development of a mathematical surrogate for a component to be embedded in a structural assembly which in turn requires all component characteristics, so modes, to be considered as described by Ewin [42]. A similar idea is pursued in a third application which uses modal properties to predict the effects of modifications to the component, so influence of substructures [42]. The evaluation of model quality, correlation, and influence of components and substructures represent aspects of interest within this work, which is why this modal approach has been utilized for the mentioned purposes. Modal testing can be expensive and depicting an obstacle on the path to improved design efficiency. Apart from comparing real physical to computational models, a lot of research work also deals with a validation approach using highly detailed FE models as reference.

One of the major advantages of this modal strategy is that it is targeting basic characteristics and dynamic properties of a model or system. The starting point from a theoretical point of view is the equation of motion of the system in which the matrix \mathbf{M} contains mass properties assigned to specific DOF of the structure, \mathbf{D} is related to damping properties and \mathbf{K} relates to stiffness attributes:

$$\mathbf{M}\ddot{x} + \mathbf{D}\dot{x} + \mathbf{K}x = \mathbf{f} \quad (2.1)$$

With regard to generalization, assuming neither external constraints nor forces results in a so-called free-free modal analysis which is often used in aerospace engineering [43, 44]. Moreover, especially in structural mechanics using FE methods, the damping term is neglected which leads together with the free-free conditions to equation 2.2 [30, p. 786]. Assuming a harmonic solution allows transforming this equation to an eigenvalue problem, equation 2.3, which yields two types of results: natural or eigenfrequencies and eigenvectors or modes. The eigenfrequencies describe the frequencies at which the system is witnessing natural resonances and the eigenmodes are describing the displacements of the DOF when vibrating at the respective frequencies. These two outputs are providing general statements about the component behavior and depict the basis for modal analyses.

$$\mathbf{M}\ddot{x} + \mathbf{K}x = \mathbf{0} \quad (2.2)$$

$$(\mathbf{K} - \lambda\mathbf{M})\phi = 0 \quad (2.3)$$

For the correlation purpose mentioned before, the Modal Assurance Criterion (MAC) which has been introduced by Allemang et al. [45–47] has proven to be suitable and thus state-of-the-art criteria to compare the derived mode shapes afterwards. The MAC is defined as follows with ϕ being the mode vector for either analysis A or reference model R , or for either analysis mode i and reference mode j respectively:

$$MAC(i, j) = \frac{|\phi_i^A \phi_j^R|}{\left(\phi_i^{AT} \phi_i^A\right) \left(\phi_j^{RT} \phi_j^R\right)} \quad (2.4)$$

Due to the normalization, the resulting MAC value for two modes is within $[0, 1]$ with a high value indicating a better mode matching. Matched modes allow more reasonable comparisons of the associated eigenfrequencies, so a more reasonable conclusion about the effect of for example model modifications.

After having set an investigation basis for comparing different variants of models or systems and integrating the MAC to allow a more sophisticated evaluation of the results thereof, the last point is to define an evaluation scope for the analysis. The scope in this context refers to the number of modes, or eigenfrequencies respectively, to consider in order to be able to draw a sufficiently well-founded conclusion about model quality. This train of thoughts leads to techniques as the evaluation of modal masses or modal

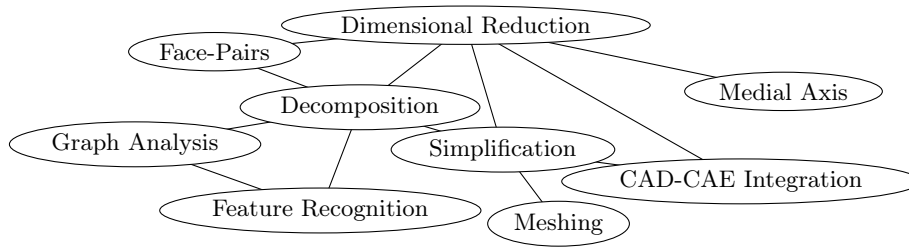


Figure 2.4.: Related fields

participation factors [48–51] to distinguish modes with a high significance for rebuilding the model behavior [52, 53]. However, most of these approaches are not leading to suitable results in free-free conditions, what in turn directed to research works as [43, 44, 50]. As stated in [44, 53], a modal mass of more than $\approx 90\%$ is sufficient in most of the cases to approximate the original component behavior. Most of the studies conducted in works like [50, 52] for constrained systems and [44] for free-free systems show that especially the lower modes are contributing to the mentioned $\approx 90\%$ in the majority of the cases. With reference to these state-of-the-art conclusions, this work is not extending its objectives by going further into detail and considers a focus on the first 50 eigenmodes in the modal analyses as sufficient to investigate model quality to a reasonable extent.

2.5. CAE model generation

Model preparation and simplifications can have an advantageous effect on simulation complexity, thus model responsiveness and usability for various processes. On a different note, these tasks require manual effort and can become tedious quickly, so overshadow the advantages. For this reason, a lot of research has been conducted in multiple fields which are interconnected to a certain degree. Figure 2.4 shows the main clusters of research associated with this topic.

In the following, this thesis is setting up a central theme which guides through these topics and research fields. The fundamental objective of those is to support component analysis, preparation and simplification, build interfaces for automation and to develop the automated process itself.

2.5.1. Simplification

The first step towards increasing model efficiency is the direct CAD model simplification. Often details in the CAD model increase the complexity of either creating a simulation

model thereof or of the simulation itself but do not exert a significant influence on basic component characteristics, so simulation results. This can result in unnecessary high design cycle times. In general, the basic idea has not only been part of engineering research but is also present in other fields like computer graphics. The approaches developed to cope with this can be divided into three major categories: methods that are based on mesh, so either facet or surface mesh, or B-Rep methods which focus on the geometric CAD entities, see figure 2.5. Combining the information gathered from mesh or B-Rep objects leads to the formulation of higher-level features which can also be considered for analysis [54]. Regarding this, Thakur et al. [55] conducted a state-of-the-art research and presented an overview in their work.

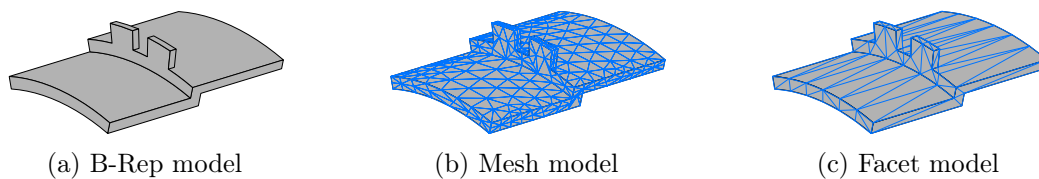


Figure 2.5.: Different model representations

Mesh-based methods consider an assembly of various mesh elements, mostly surface elements, as a starting point. Different algorithms have been developed which focus on decimation of element count by merging elements. Hoppe et al. [56] for example present an approach for mesh optimization including simplification using an energy function to reduce the number of vertices in a dense mesh. In a later work [57], the term Level-Of-Detail (LOD) is introduced within this scope in association to defined simplification, compression and refinement steps. Schroeder et al. [58] pursue a similar approach along with an exemplary application on an engineering related model, an aero engine blade. Soucy et al. [59] make use of a hierarchical triangulation approach to compress mesh surface information while preserving topology. Garland et al. [60] use quadric error metric for surface simplifications approximations with respect to geometric errors.

In the engineering field, to be more detailed in Finite Element Analysis (FEA), element size is an important parameter that exerts a significant influence on simulation quality. As a consequence, mesh decimation can lead to undesired mesh distortions. Additionally, mesh distortion negatively affects the similarity of the mesh and underlying exact CAD geometry, which can be the reason for different simulation results.

FEM Software or in general meshing algorithms nowadays often include parameter and settings to induce similar mesh simplification methods internally. An exemplary parameter is the small feature tolerance of Siemens NX which allows an integrated automatic mesh

simplification [34].

However, the exclusive focus on mesh elements without taking context into account has its limitations. For this reason, further research generated methods that involve region building and analysis. After having identified target regions, methods like element decimation or removal can be applied to achieve simplification effects. Starting from a CAD mesh, Gao et al. [61] describe a process which uses a watershed approach to identify and segment regions. This information is used in form of a Region Adjacency Graph (RAG) for feature recognition on behalf of a user defined feature facility and thus allows a straightforward feature suppression for simplification purposes. A similar objective is pursued in [62] and [63] which present different methods for CAD mesh segmentation and deriving meaningful features. Quadros et al. [64] show up approaches to suppress features after their identification.

Since the fundamental idea of dividing an object into meaningful parts is not only interesting for engineering, but also represents a basic psychological aspect, a lot of research has been conducted in other fields. Mesh information with context has been used as basis for self-learning algorithms in order to recognize shapes on their own as well as subdivide it into recognizable parts. Overviews of conducted research and development in this regard is given in [65] and [66]. While works like [67–72] focus on basic 3D objects, researches as [73–76] put CAD models into their focus.

Other approaches are pursuing a similar target but on a different path. Instead of mesh elements and their connectivity, their starting point is the B-Rep model including its association information and so share a closer similarity to the CAD world. Earlier approaches, as in [77], combine both B-Rep and FE mesh information since a direct connection between CAD and CAE has not been common. On behalf of element and surface data, simplification operations have been introduced. Features like blends, holes and fillets introduced for manufacturing and detail reasons are often a problem for meshing algorithms, thus increase complexity to an unnecessary extent. For this reason, their automated suppression or removal has been the topic of several works as [78] and [79]. Nevertheless, this procedure is not achieving the desired lower level of detail in many cases.

Similar to mesh-based approaches, the next logical step is to consider entity context for more information, so a more sophisticated feature-based simplification. To achieve this, the features have to be identified first. Shah et al. [29] gives a discourse of various methods for geometric feature recognition in the CAD field and categorizes them to topological, heuristic, symbolic, volumetric, process-centric and hybrid techniques. Another review of state-of-the-art techniques in this context is given in [80]. For the objectives of this

thesis, topological and volumetric approaches depict the points of major interest.

In the topological approach, graph-based methods make up a large section. These graph-based methods are similar to the approaches described in previous section except for that instead of mesh elements B-Rep surface entities are representing the graph components, see figure 2.6. In addition to face and edge relations, attributes describing the components themselves are introduced to the graph to extend its degree of information in the so-called Attributed Adjacency Graph (AAG) [29]. The edges connecting face nodes are attributed with e.g. integer values to denote their concavity properties and thus support reasonable graph structure browsing. This idea has been pursued further and lead to the development of a multitude of various graph representations [81], as for example the Multi-Attributed Adjacency Graph (MAAG) [82] and Multi-Attributed Adjacency Matrix (MAAG) [83]. To overcome remaining drawbacks like recognizing interacting features [84], further research has been conducted in [81, 83, 85–87]. Based on the resulting B-Rep graphs, a technique called graph isomorphism is used to identify and recognize sub-graphs associated with specified features. Mao et al. [88] evaluate the use of Apriori-based Graph Mining (AGM) for the problem of frequent subgraph discovery while projecting the AAG to a two-dimensional plane which describes specific face shape characteristics. However, Ma et al. [88] concludes that AAG comparison is not flexible enough to account for small differences in identified structures and furthermore, potentially important aspects like dimensions of structures and their relative ratios are not part of the shape comparison formulation. To improve the flexibility of graph-based recognition, Di Stefano et al. [89] introduce semantics as additional information to their approach. In this scope, semantemes describe the minimal element of meaning that the system can identify given a geometry model, which build interfaces to graph relations and information.

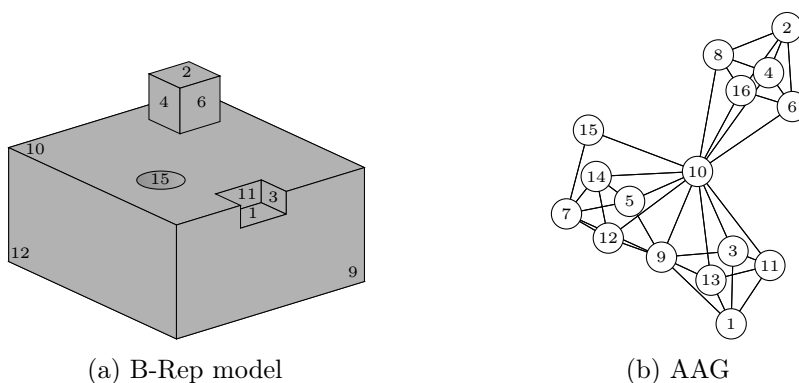


Figure 2.6.: AAG process

Due to still remaining limitations of graph-based approaches, Cao et al. [90] published research on an interactive feature recognition which utilizes user-interaction to identify, analyze and extract more complex features.

In general, most of the presented graph-based research is dedicated to recognizing machining features for Computer-Aided Process Planning (CAPP) purposes. Compared to machining features, which share a similar topology due to their nature in most of the cases, the aspect of interest of feature recognition in this thesis is to identify substructures suitable for simplification effort and methods to remove negligible structures. These in turn can be of arbitrary shape and topology and thus make a recognition rather complicated.

For this reason, other Feature Recognition (FR) methods set a decomposition of a given 3D model in first place and so depict an attractive interface for component simplification and understanding.

Convex hull or cell-based decomposition focus on segmenting an input model into intermediate volumes which are manipulated afterwards to produce form features [80] and for that reason are defined as volumetric methods. As the name suggests, the convex hull method utilizes the smallest polyhedron convex hull that is circumscribing the present 3D model. By building the set difference between hull and model, the so-called convex deficiency is obtained [29]. This step is repeated recursively until an empty deficiency is achieved, see figure 2.7. The basic idea of this methodology has been initiated by Kyprianou [91], further pursued by Woo [92] and later extended by Kim [93] as described by Han et al. [94]. Additional extensions, modifications and proposals to this process have been developed and evaluated in [95–98]. Han et al. [94] describe the major steps of the hull based decomposition as follows: decomposition using the alternating sum of volumes with partitioning (ASVP), recognition/generation of form features, generation of primitive machining features and machining feature aggregation.

The fundamental research on cell-based approaches has been done by Sakurai et al. in [100, 101]. In the course of the years, many more researchers dealt with this topic and extended methods to overcome existing limitations. Similar to the convex hull decomposition, the cell-based approach consists also of multiple steps: retrieving delta volumes, decomposing delta volumes into minimal cells, cell composition and final feature classification [94]. Its core principle is based on traces created by machining features and extending surfaces of the delta volumes based on local criteria, e.g. edge concavity, to create the cells [102]. However, this can lead to numerous cells in some cases which facilitate their composition to meaningful features. Works like [103] and [104] introduced the method to combine resulting cells to so-called maximal volume features for a more

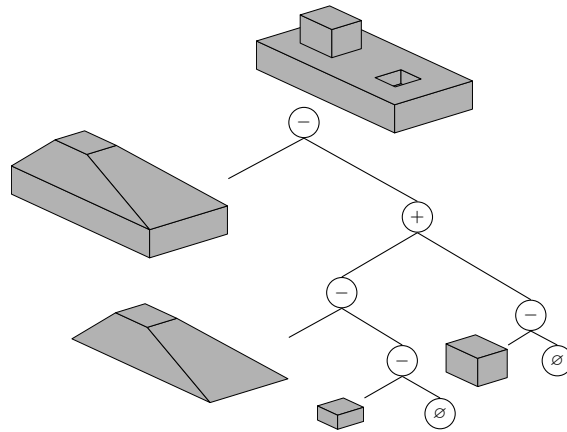


Figure 2.7.: Convex hull decomposition (after [99])

systematic approach. In [105], Woo describes the application of the maximal volume methodology for the model decomposition by localized face extension. Woo et al. in [102] further pursue this idea and present a divide-and-conquer approach to improve process scalability and mitigate the problem of multiple feature interpretations.

Regli [106] presents research on a trace-based feature recognition in the machining environment. In the associated context, traces are related to geometric and topological properties embed in the solid model and combined with a hint-based approach for feature identification, as for example the conical ending surface of a hole supports drilling feature recognition.

Nevertheless, methods focusing on machining features limit the potential application spectrum. More general approaches for CAD preparation, simplification or decomposition open up a wider application spectrum. White et al. [107] have been one of the first to use the virtual term for geometry decomposition. Via pseudo or virtual geometry, complex topology is decomposed to mappable sub-volumes suited for more optimal meshing primitives. The presented approach aims for a reduction of required manual interactions and is based on mesh entities and uses a node and face traversal process to identify virtual cutting planes. Sheffer and Blacker et al. [108–110] introduce the virtual topology in the B-Rep framework, which is a way of separating topological information and geometry while maintaining their association. An advantage of this is that topological operations can be executed on the decoupled virtual topology in a more efficient way because geometric changes are typically expensive and complex [108]. This virtual construct is built from basic B-Rep entities, as faces, edges and vertices, and contains their adjacency information. Basic operations as merge, split, collapse and connect are described to apply modifications

to the topology in order to prepare and simplify geometry for the meshing process. The merging for example focuses on the combination of geometry of similar topology in order to eliminate boundaries and small details, so obstacles and unnecessary boundaries for meshing algorithms. Similar to this, split operations are designed to separate domains for a clean mesh boundaries. On top of the virtual topology, Sheffer et al. use a local analysis to cluster faces to regions which are target for the topology operations afterwards in [111] and [112]. The idea of virtual topology is still pursued in later works. Tierney et al. [113] employ virtual topology to link boundary conditions defined on the original model to simplified and decomposed analysis models. In [114] Tierney et al. go into detail regarding virtual topology for model simplifications and describe both virtual volume partitioning and splitting operations. They present a flowchart for the process of topology extraction, virtual decomposition and meshing using the software Siemens NX and ICEM CFD.

The term cellular modeling has also emerged in this scope in order to provide another approach to model preparation and feature analysis. Bidarra et al. [115] describe the use of a cellular representation to solve various feature modeling problems and operations to prepare a geometry based on this methodology. This cellular model depicts the basis for defining and maintaining feature semantics, especially during and after modeling operations in [116]. Armstrong et al. present in [117] a framework using this cellular modeling strategy to overcome different structures of different CAE software and modeling paradigms. Later research as [118] utilizes this for the preparation of B-Rep models for analysis including small feature suppression possibilities. The principle of combinatorial topology is used for the cell complexes to develop transformations and operator pairs as collapse and explode which implies adding a dimensional increase or decrease like e.g. a face to edge conversion. Other operators provide splitting, joining, inserting and removing functions for further cell transformation and conversion. The authors of [118] use this as a starting point for feature suppression but point out that many features require more than one operator. Another path is taken in [119] which combines the cellular topology idea with feature-based model sequence concept to build progressive solid models for providing different LOD. In this regard, a model is decomposed into various cells, which are evaluated based on their delta volume and composition type: positive or negative. This cellular modeling framework has proven another use for transferring or maintaining boundary conditions and semantics between models of different detail level in [120–122].

Other than described in the previous two sections, Belaziz et al. [123] utilize a morphological analysis of the component for model simplification and idealization. In this approach, topological modifications of entities, as faces, contours, edges and vertices, are

investigated to identify morphological changes from one state to the next one. Based on this changes, features can be derived which shall finally support the construction of a tree similar to the CSG. This tree can then be used to apply idealizations to the product model using external information from an analyst.

Based on a similar idea, other decomposing methods focus on edge loops and local concavity in order to identify potential splitting planes, thus decomposable structures. Lu et al. [124] identify and analyze edge loops of a B-Rep model. Based on the edge loop type, different methods for constructing and selecting the cutting surfaces are developed and presented. Concavity information is also used in the approach in [125] where adjacent faces are extended in order to build similar cutting surfaces. The resulting segmented structures can then be matched with predefined features for recognition purposes. Concavity of edges, islands, depressions and joints are also making up the key aspects for solid decomposing in the research of Chong et al. [126]. This approach first splits islands formed by closed concave edge loops and afterwards identifies the closest edge pairs and collapses them to medians. The type of medians guides the selection of appropriate splitting procedures. Also Koo et al. [127] put concave and convex loops in their focus in their wrap-around operation. In this, inner convex loops are traced to derive and delete interior faces, so create a topological change which exposes potential features. On part and assembly level, the authors utilize the previously mentioned Face Adjacency Graph (FAG) methodology to identify starting faces for the operations and present two case study examples. Later work by Kim et al. [54] presents a method which combines and extends state-of-the-art methods. Prior chamfer, fillet and round decomposition build a better accessible starting point for wrap-around, volume split and cell-based decomposition. As already described, the wrap-around is focusing on inner convex loops. Contrary, the subsequent volume split step targets inner concave loops and copies resulting entities for the volume splitting operations, see figure 2.8. By this, the effort for the final cell-based decomposition is reduced, thus positively contributing to the overall performance. In this last step, the principle of maximal volume decomposition is implemented, which is also putting the focus on concave edges and the extension of the adjacent faces. Afterwards, Kim et al. visualize the decomposition steps in a CSG tree-like representation which is basis for LOD considerations. Feature analysis yields feature importance measures to determine LOD reduction steps.

Boundary edge loops are also key to simplified regions in [128] which provide model knowledge for simplification steps. In this method, different entities as co-surface loop, loop chords or loop bridges are introduced as tools to determine these regions, which can be simplified by repair or patching operations. The selection of which region to be

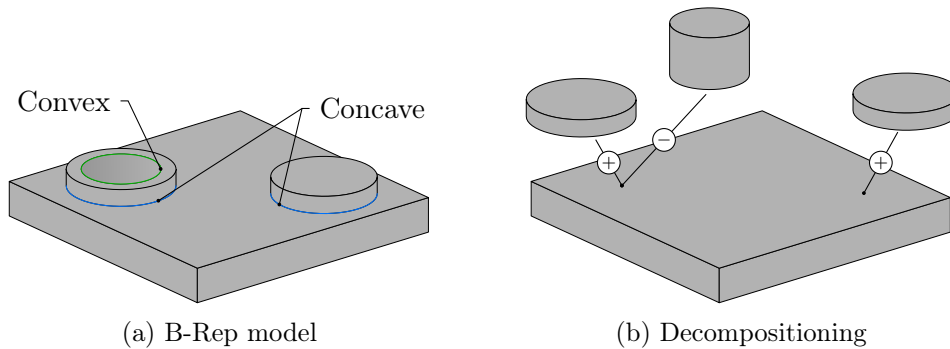


Figure 2.8.: Volume decomposition using concavity information (after [54])

simplified is either done automatically by defining a threshold value, by manual user input or a hybrid variant. Zhu et al. [129] concentrate on the decomposition and recognition of rib-like features for additional preparation purposes. In this scope, concave and parallel edges and neighboring face properties help to identify rib features which are segmented and regarded as separate structures in the ongoing process.

Boussuge et al. [24, 130, 131] developed a methodology to extract generative processes from B-Rep shapes which shares the topology and entity analysis of previously mentioned methods but puts its focus on other face and edge relations. Starting with a principle based on maximal faces [132] and edges, its objective is to identify and remove primitives iteratively to reverse the construction history to the root primitives. So-called lateral edges and faces have to be retrieved in relation to the maximal face, which in turn guide the extrusion and define the extrusion distance. After this primitive shape has been evaluated, it is removed from the model state and the iterative process starts anew.

Other approaches make use of information besides local properties which is embedded in the CAD model during the design step. Robinson et al. [133] utilize sketch data which is basis for a majority of primitive CAD features to gain insight into substructures, thickness distributions and possible feature types. The derived information is used to identify slender regions for subsequent model partitioning and simplification purposes. In [134], Chentao et al. present a method which makes use of integrated and accessible feature information. The feature tree which is built in the modeling procedure is accessed to suppress features, so to simplify the geometry.

Besides local properties or features, symmetry is a popular aspect to be exploited for simplification assumptions in the simulation stage. Tate et al. for example state in [135] that symmetry of solids has been interesting for research for a long time but methods for detecting symmetric in B-Rep models are still hard to find. For this reason, they

evaluate state-of-the-art methods and present a method for symmetry detection using face loop analysis to derive candidate axes and planes of symmetry. Furthermore, they provide an outlook for isolating symmetric and asymmetric elements. This aspect in particular has been center to other researches as well. In [136] for example, Boussuge et al. analyze B-Rep CAD models to identify axisymmetric and cyclic repeated sectors in order to decompose the model to produce volume meshes of appropriate resolution and quality. To achieve this, face properties and especially topological aspects as associated edge loop types help to derive potential axisymmetric segments first. Cyclic entities are then detected based on similar and patterned shapes along the center axis. This knowledge is central for the subsequent isolation and segmentation of axisymmetric and cyclic structures using cutting planes in Siemens NX [34]. Tierney et al. [137] continue to pursue this idea and extend it by interface information on assembly level in order to propagate symmetry information. This is done to guarantee appropriate mesh transition conditions, thus simulation model quality. Another form of data for CAD analysis is used in [138]. Casero et al. start from a faceted 3D model and identify silhouette edges after projecting facets to a 2D representation. Algorithms from the field of computer vision offer techniques to convert the rasterized representation back to 2D contours for CAD purposes afterwards.

2.5.2. Dimensional Reduction

Shell or beam elements can bring a lot of advantages in appropriate cases compared to a full 3D volume meshing approach as described in section 2.4. Especially in order to derive fast simulation responses, a lot of research has been conducted to reduce dimensionality of models, therefore involved FE elements. Two major methods have emerged and proven as most suitable in this scope: the medial axis or object and the face-pairing technique.

Medial Axis

The origin of the term medial axis can be traced back to the works [139, 140] by Blum H. who invented the principle of medial axis for shape description in biology. The motivation for this idea has been to overcome the difference between the present biological problem and the physical problem which is to solve and to provide methods to assess which purpose a geometry is trying to accomplish. In the course of the years, a lot of research has been dedicated to this methodology and its tuning, extension and application. [141–164] are selected exemplary works for further reference. Medial axis methods or algorithms related to the medial axis are also often referred to as skeletonization algorithms. In computer

graphics, for example, a medial axis related method called Shape Diameter Function (SDF) emerged for this skeletonization purpose and for movement and shape description [71, 165–167].

A common technique to derive the medial axis is to use the so-called Delaunay triangulation and Voronoi graphs. These are state-of-the-art methods in various fields of science and implemented in a multitude of open-source software packages. In this regard, only the fundamentals are described in the course of this thesis. Voronoi graphs describe regions in a \mathbb{R}^n space which are defined by a center and contain a subset of points which are closer to their respective region center than to every other center, see equation 2.5. The Delaunay triangulation is built around the optimization objective to maximize the minimum inner angle of triangles in \mathbb{R}^2 or tetrahedrons in \mathbb{R}^3 . As a consequence, the condition has to be fulfilled that the circumcircle of a triangle may not contain any further point of the given set of discrete points. Delaunay [168] has proven the duality of the Delaunay triangulation and the Voronoi diagram, see figure 2.9.

$$R_k = \{p \in \mathbb{R}^2 \mid \text{dist}(p, D_k) \leq \text{dist}(p, D_i) \forall j \neq k\} \quad (2.5)$$

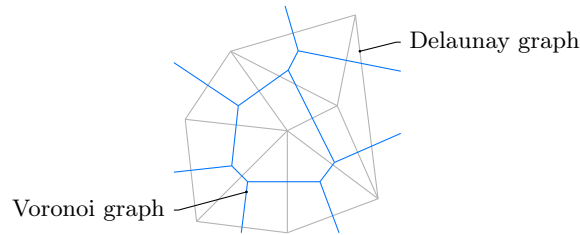


Figure 2.9.: Duality of Voronoi diagram and Delaunay triangulation

In the 2D space, filtering external Voronoi edges and connecting internal Voronoi vertices leads to the desired skeleton line, or medial axis, and the combination with the triangulation yields local circumcircle diameters. Analogously, the 3D dimensional process leads to medial surfaces, often referred to as medial object, with its associated inscribed spheres, see figure 2.10.

Face Pairing

Another technique that has emerged especially in the CAD field is the technique of face pairing has been introduced by Rezayat in [169]. In this research, Rezayat presents a novel method to abstract geometry models for the simulation-driven design process via detail removal and dimension reduction by mid-surfacing. This approach consists of steps

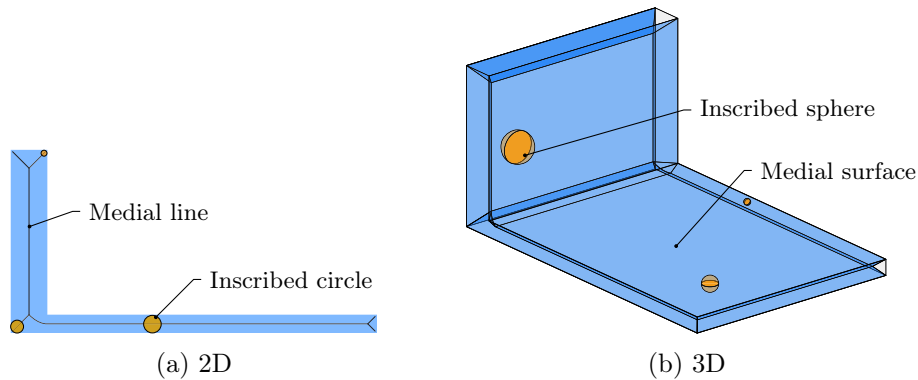


Figure 2.10.: Medial axis/object

as B-Rep surface pairing, setup of adjacency graphs based on topology and geometry, see section 2.5.1, generating the midsurface patches and finally sewing and connecting the patches based on the adjacency information. An exemplary application is shown in figure 2.11

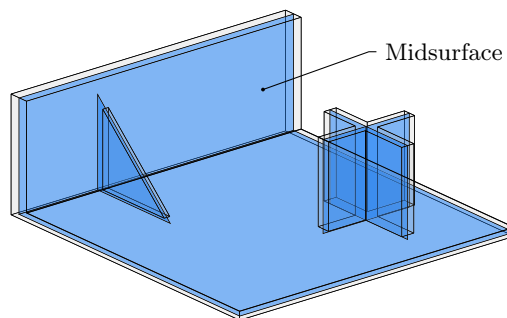


Figure 2.11.: Midsurface by face-pairs

On account of the B-Rep CAD information, potential face pairs are filtered by a maximum thickness, thus distance, parameter. The proximity to the root geometry on the one hand allows an efficient approach, and on the other hand can generate appropriate and clean midsurface patches which seamlessly fit into the model. This is a potential advantage compared to methods based on medial axis. Ramanathan et al. [170] combine both medial axis and face-pair mechanisms to extend face pairing capabilities. Another approach to face pairing is pursued by Chong et al. [126] who investigate edge relations, concavity properties and conditions, collapse edges to medians and then use this information to build face-pairs for midsurface creation. Lee et al. [171] propose an extended face-pairing method which casts rays to identify face-pairs and then sets up different graph representation for the pairing as well as the midsurface patch generation and joining process. Edge

information is stored in these graph representations to allow information mapping and provide more detail for joining operations. More detail about evaluation metrics of face pairs is given in [172]. Here, Sheen et al. describe different conditions which have to be fulfilled for pairing to generate a more robust and suitable starting point for the simplification as for example face normal properties and overlapping states. In a later work, Sheen et al. [173] present a different approach called solid deflation, which aims for a more general applicability of mid-surfacing methods. After face pairs have been detected, a medial surface is calculated for each pair which then replaces the original face entities in the topology and by this a deflation effect is achieved. Instead of deflating a model, Lee et al. [174] describe an offsetting technique for non-manifold topological models. By geometry analysis and Euler operations, a similar effect to deflating is introduced in their work which can be used for both positive and negative offsetting.

Applications

The described methods have found applications in various processes and approaches for component preparation and simplification. In general, the major fields of implementations are component analysis and creation of simplified models of lower dimensionality, for example midsurface shell structures. Both ideas are naturally strongly connected and often rely on each other.

Stanley [175] presents a holistic approach for model simplifications and dimensional reduction using the medial axis on aero engine components. In this case, a 2D section oriented application of the medial axis using Matlab [176] is presented as a tool to build shell models. These are evaluated regarding suitability and validity. An extension of this approach to a 3D domain is presented afterwards including an application to a sector model of an aero engine component which is validated against a reference 3D simulation model. However, the performance and robustness of the approach have led to further research as in [177] which focuses on accelerating the process. In this regard, faces leading to unnecessary medial branches or flaps are neglected in the process what decreases complexity, thus improves method performance. The identification of these faces is achieved by a tagging and geometry preparation step prior to the main process. Subsequent research [178, 179] uses this work as starting point for implementations and evaluations in an extended whole engine environment.

A direct approach, however, is not leading to desired results or not possible in many cases. Consequently, another cluster of researches in this field set a geometry analysis prior to geometry modification tasks. Two major paths have emerged for this geometry

analysis. One emphasizes the identification of thin-sheet regions while the other one pursues a more hierarchical approach based on decomposition techniques, see section 2.5.1.

In order to identify thin regions, which could be approximated by beam elements, Donaghy et al. [180, 181] pursue a medial axis approach. Similarly, Armstrong et al. [182] investigate the use of the medial axis for detail suppression and dimensional reduction of 2D faces to a set of skeleton curves. Furthermore, the extension to the 3D domain with the help of axisymmetry and the associated 2D section is described. A similar idea is pursued by Robinson et al. in [133] where they concentrate on CAD feature sketches which represent 2D domains. These are then analyzed using medial axis tools to identify thin regions. The skeleton line related to thin regions is afterwards used for CAD face creation and for rebuilding the geometry using reduced representations.

In [183], Robinson et al. extend the applications and identify thin 3D domains based on lateral dimensions and ratios, pursuing the target of separating thin from chunky regions. Based on commercial medial axis or object implementations, midsurfaces are created and those related to thin regions are filtered by the associated inscribed circles or spheres. As subsequent clean-up operations, a 2D MAT is applied to the face pairs of thin-regions which then serves as another input for cellular modeling. The resulting projection is then used to build cells for shell and volume representations, thus mixed dimensional models. More similar applications are described in detail in [8, 9].

Another extension is introduced by Nolan et al. in [121] which adds long and slender domains to the list of regions to be identified. The authors present a FE beam modeling method for those regions and describe approaches for combining and reconnecting the interfaces between domains of different dimensionality by for example MPC constructs. Makem et al. [184] present an ellipsoid-based fitting algorithm which helps to additional subdivide and categorize these domains for a more structured meshing approach.

While the mentioned approaches put emphasis on the isolation of thin regions, another cluster of research deals with the decomposition of geometry first. An overview of different decomposition methods has been given in section 2.5.1. One major advantage of this approach is that structures which would imply difficulties for face-pairing or medial axis methods are separated into better accessible and potentially cleaner structures.

In this context, Woo et al. [185] make use of these methodologies to build more robust and comprehensive midsurface models. The solid model is decomposed into simpler volumes, which are then abstracted using face-pairing methods. The resulting midsurfaces are extended, trimmed and merged according to the original geometry. A similar idea is pursued by Boussuge [24] who first investigates the geometry derived from the previous

decomposition process. The related maximal faces are investigated using medial axis methods in order to identify a maximum thickness. Based on that and the extrusion distance, the domains get classified into solid, thin and long and slender regions automatically. Afterwards, the segmentation information and thus geometry interface relations guide the assembly of the converted geometry representations to the final simplified model. Zhu et al. [129] also follow a hierarchical idea for the midsurface abstraction of thin-walled models and present a rib-focused decomposition and simplification process. Edge properties guide the geometry splitting and setup of a hierarchical structure, which is then used for building midsurface entities by an extension of the chordal axis method.

2.5.3. Information integration

Besides the amount of research in the field of component preparation and simplification, other aspects have to be taken into consideration for the creation of assembly simulation models. In this regard, boundary conditions, interfaces, connections as well as simulation objectives play a crucial role for the setup of simulation models, thus for the resulting model quality and suitability. Setting up these can become a quite tedious and repetitive process coupled with disadvantageous amount of manual effort. This in turn affects process efficiency, and therefore depicts an attractive target for automation research and development. To manage the driving factors for automation, the manual process has to be investigated first and broken down into major steps. Knowledge about for example interface locations or faces which are used for boundary condition application is mandatory for automating the process. Additionally, strategies of applying and implementing simulation conditions and decisions have to be developed and defined for the process. In general, the most common strategies can be described as name or tag-related, user input, automated identification and feature or software-embedded information.

Modern CAE packages often come with a direct interface for CAD geometries, which enables a straightforward simulation setup. In this regard, meshing algorithms, boundary conditions or properties are often directly applied to geometric entities as volumes, faces, edges, curves or vertices. For the automation of a full simulation setup, researchers have proposed a strategy which uses CAD tags in order to guide an automated process [186–188]. In this context, tags refer to names which are assigned to geometric entities and inherited from the CAD in the CAE model. These tags are used as targets for instructions within the CAE process, thus allow an automated workflow. The input to this workflow is an external database file which contains both CAE instructions and associated tags.

On the one hand, this depicts a straightforward and also robust approach, but on the other hand comes with difficulties, especially in larger assemblies. The introduction of tags can become undesirably tedious. Furthermore, geometry modifications can affect the integrity of the tagging strategy, thus require a re-tagging, especially if interface entities are altered. Another way of utilizing tags is achieved by Benaouali et al. [6] who associate exported geometry with a nomenclature for the subsequent FE meshing and load application process.

A similar goal is pursued by Nolan et al. who developed a strategy to capture simulation decisions so that they can be used to link different representations and to set up assembly analysis models [120–122]. They introduce and present this idea named simulation intent in [122]. Based on the three technologies cellular modeling, equivalencing and virtual topology, the link between root and modified models can be maintained and global information for other modeling and simulation steps is embedded and accessible for downstream processes. The simulation intent database includes modeling and idealization decisions and settings for boundary conditions, mesh and solution types [122]. Lecallard et al. [189] describes this database as an external common data structure which is stored in SQL format and provides attributes associated with the created cells. Works like [10, 136, 190] demonstrate advances in this methodology, the user interface for the input of objectives and variables and showcase additional applications.

Both of the described techniques are basically based upon a similar principle: integrating information for subsequent steps and decisions within the model. This leads to another group of researches situated in the field of CAD-CAE integration. In this scope, the aim is to embed different types of information and knowledge in a geometry or data model, which then provides the input for downstream automated processes. Different ways have emerged in the researches which focus on external databases, CAD software integrated data or feature-based information.

Lee et al. [191] propose a feature-based modeling system to overcome the problem of CAD and CAE being not well integrated. The authors exploit feature modeling capabilities to provide multi-resolution models which serve as starting point for CAE models with a different level of abstraction. At the current state, the presented process is limited to a number of simpler features as bosses, ribs or holes and user interaction is recommended for more difficult features. The information required for CAE or analysis is merged into feature modeling operations in the master part, thus is directly integrated in the modeling or feature tree. Hamri et al. [12, 192] aim for a CAD-CAE integration at software environment level and enrich a CAD model with additional mechanical information, e.g. boundary conditions. Based on their mixed shaped representation and

the High Level Topology (HLT) technique, model simplification and adaption is executed while considering the integrated boundary conditions. Another approach to integrating information is to use a common data model (CDM) [193]. Both CAD modeling and CAE analysis information is stored in a parametric form in the CDM and associates CAD and CAE entities while maintaining dependencies throughout the process. Additionally, knowledge-based tools and software which include for example engineering calculations using provided model parameter. The advantages of a common data model, however, imply the disadvantages of additional effort for setting-up the models, especially for frequently changing and non-standardized parts. Similarly, Xia et al. [194, 195] propose a unified data model which supports the analysis attribute setup for material and boundary conditions.

2.5.4. Assembly analysis

Integrated models usually imply a high suitability for specified problems, especially in standardized cases within relatively constant environments. However, disadvantages can emerge besides advantages in more frequently changing or non-standardized cases. Setting up integrated CAD-CAE models demands significant effort, which increases exponentially when moving to larger and more complex systems. More components and interfaces results in more parameter which can be interfering, thus restricting design space. In other cases, component modifications by e.g. parametric changes can distort the integrated relations or information, which in turn necessitates updating the integrated model. For this reason, a major aim is to decrease effort for steps as geometry tagging or database construction.

Interfaces between components are an important part of assembly simulation models. For this reason, an automated recognition of interfaces and their type brings benefits for setting up simulations or reducing effort for integrated models. Different paths to cope with this have emerged in various research works.

Commonly, assemblies are constructed by components and so-called assembly features which denote relations and DOF between components [34, 196–198]. On the one hand, these assembly features guarantee a consistent geometry, and on the other hand provide information for interdisciplinary tasks like process planning or also for the objectives of this work. In many cases, however, these assembly features cannot be taken as consistent or granted. For this reason, research has been conducted to automatically instantiate those for tasks as rebuilding an assembly structure for legacy models or reverse engineering. Similarly to the AAG methodology presented in section 2.5.1, the AAG can be extended

to an assembly scope and enriched with contact face information, which is a starting point for feature or shape matching. In [199], this contact information is either gathered by manual user input or by an automated contact identification process, which is followed by an interactive user verification.

This identification of contact face pairs or areas is, in general, a substantial part of the assembly analysis. One approach to cope with this is to work with B-Rep entities and the related face information. Based on the topological face type, i.e. planar and cylindrical, functional faces can be filtered and matched according to the inherited properties and a specified distance tolerance [200–202]. A more abstracted path is proposed by Joudes et al. [203] who use a point cloud and GPU ray-casting methods to identify and imprint interfaces and retrieve concrete contact areas.

The identified interfaces and contacts are key points for various downstream applications. Besides the assembly feature re-instantiation, the set of contact arrangements, either derived by contact evaluation or assembly features, can offer additional information about mechanical and kinematic constraints and connections between interfering components. Similar to the cognitive process in the engineering domain, this aspect significantly supports the derivation of semantics in an assembly.

In for example [204], assembly features are analyzed in detail to build an assembly design semantic model. Based on mating or joining conditions between two parts, the related position, dimension and fit properties help to specify the present situation to either welding, riveting or screwing feature. Another application case is described in [200, 202] where joints and connections are used for component mobility investigations, thus part DOF, or for automated disassembly methods.

Extending the perspective leads to the role of interface arrangements for parts in an assembly. Part relations not only inherit joining or mating face pairs but also information about the parts and their functionality. Assembly search and retrieval methods as described in for example [205–207] enrich adjacency graphs with interface information and utilize graph mining and pattern matching approaches for assembly or part search operations in a database, which shall promote model and knowledge re-use. A similar objective is set in [208–210]. Here, Lupinetti et al. introduce the concept of enriched assembly model, which adds different layers to the attributed graph structure. Statistic information about for example model area or surface coverage types, structural data about components or patterns and shape details of parts are put into relation with interface information which is retrieved by the SolidWorks Application Programming Interface (API). Based on this as input, multi-criteria similarity assessments of CAD assembly can be conducted and similar sub-assemblies and components can be retrieved from a query

base. More component focused shape retrieval or part analysis and query methods are described in [211–216].

Apart from pattern matching and shape search, the function of components is also often directly or indirectly associated to its relations within an assembly. Fittings, contacts or interferences are common design intents often related to a specific function. Shahwan et al. [201, 217] focused on this idea and investigated component interfaces in detail and proposed an approach based on qualitative reasoning to identify design arrangements of importance for later FE analyses. In this scope, a set of interfaces is analyzed in order to elaborate the equilibrium state of participating components which in next turn describes associated functions. An application case of this process is given on a screw connection example by the authors.

Vilmart et al. [218] also took up on the problem of non-consistent CAD assembly models and missing assembly information. The authors extend the scope and propose an intrinsic knowledge-based assembly model. Their approach combines many of the already mentioned methodologies and strategies and connects them with a knowledge-based system. To achieve this, the process utilizes a set of open-source available software packages for CAD or general geometry analysis which provides the required API interfaces for the developed methods. Geometric interfaces, component shapes and symmetry data as well as repeated sets or patterns are extracted by an automated process from a present assembly and the derived data is forwarded to populate the knowledge base. At the current state of this work, only generic geometric properties are stored while functional dependent information is left for future work according to the authors. This stored data is fed into an engine for inference mechanisms, which uses ontology and symbolic-based operators to derive higher-level functional information. The built knowledge base then incorporates the user or program interface for querying the assembly content using assembly ontology.

2.6. Knowledge representation

2.6.1. Knowledge base

The term Knowledge Base (KB) directs to the next fundamental building block in the engineering design and simulation process. A lot of tasks are executed according to engineering judgment, knowledge and experience from physical relations, past investigations and design iterations. This represents an essential aspect for decision-making in the design process, which is at the same time complicated to cope with and integrate with regard to a full automation.

Knowledge base is a general term which is present in various domains and an extension to the basic database principle. They commonly inherit more structured, hierarchical or relational information compared to tabular storage of databases [219, pp. 6 sqq.]. This knowledge base is used by a Knowledge-Based System (KBS) in order to solve more complex problems, for example guide process tasks as an Expert System (ES) [220]. Inference mechanisms or specified rules deduce decisions, data or general information from the knowledge being embedded in the KB on behalf of the posed input query and by this, help to improve processes in various domains. An example of a powerful and omnipresent knowledge base is the internet [221]. Works like [219, 220, 222, 223] provide further insight and information about general KB, ES and KBS strategies.

The described KBSs have found their way to the engineering domain, where they often appear as Knowledge-Based Engineering (KBE) or Knowledge-Based Engineering System (KBES) [224–226]. The fundamental principle of extracting information from various data types and sources and deducing a requested result remains the same. Chapman et al. [224] portray the engineering knowledge base as processed expertise derived of computer software, regulations, design guides, handbooks, existing designs and analysis results. More specifically, common information sources in KBE are spreadsheets, documents, rules of thumb based on engineering intent, engineering formulas or drawings [227]. The stored knowledge captured from engineers is then assisting designers in product creation or development or related business processes. A majority of KBES is heavily connected with CAx software to enable a simultaneous and straightforward information transfer. Exemplary use-cases of deduced KBE information are automated processes to generate product information, 3D models or drawings [228].

In order to derive an answer to an imposed problem, a reasoning or so-called inference mechanism is elaborating available information. Common symbolic representations for structuring and formalizing the knowledge, and thus shape it for the mechanisms, are based on rules and frames [229, 230]. Rules in this context are defined by if-then schemes and are the most straightforward approach to deduce information. Frame-based approaches share their working principle with object-oriented programming. Similar to the programming domain, frames are objects which store information in a structured way as attributes and can be used to instantiate different occurrences. This method allows a concise and efficient storage of information and also hierarchical relations which can be processed by rules afterwards.

Due to the proximity of KBES to the design process, commercial CAx systems often provide a knowledge base package as a supplement. Knowledge Fusion (KF) for example is a KBS framework embedded in the Siemens NX platform [34, 231] which shall support

the design and analysis process in the CAD domain. Along with Graphical User Interface (GUI) integration comes the ability to directly associate knowledge frames and rules with CAD entities, which allows process streamlining.

Exemplary applications and descriptions of KBE approaches are proposed in [225, 227, 228, 230, 232–237]. While Zhan et al. investigate KBES for assemblies including axles, supports and interfaces in the CAD domain in [234], Zeng et al. [233] evaluate their use for FEA modeling and multi-body coupling problems. On the FE side, more precisely meshing, Dolšak et al. [235] populate their knowledge base with attributes and topological relations of edges and associated meshing conditions in order to produce adequate meshes based on the stored knowledge. A more extensive KB related research is done by Katona et al. [236, 237] who deal with the subject of deviations between real and simulation model characteristics and results. For this purpose, different types of process- and product-specific knowledge are integrated in the knowledge base which supports the decision-making process and recommends model preparation if required.

2.6.2. Machine learning

However, the more complex the problems, the more complicated the reasoning process. Crafting effective rules or even data structures to cope with highly complicated relationships is depicting a major challenge and obstacle for adequate knowledge mechanisms in knowledge base constructs. The field of Artificial Intelligence (AI), which shares strong relations to the KBS field, offers a plethora of methodologies and strategies to deduce information from given data, so knowledge. Machine Learning (ML) is a subfield of the AI domain which is currently on a prominent rise due to increasing computational power, amount of data and data resources and better and more sophisticated models and methods [238, pp. 2 sqq.]. An advantage of most of ML algorithms is the paradigm change that is entailed by their introduction. Contrary to conventional programming, no explicit detailed design is required. Conventional algorithms traditionally combine expert rules with data in order to derive answers to a posed question, while machine learning makes use of data to retrieve the rules behind a problem [239, pp. 5 sqq.].

The focus of this work is on developing a conceptual approach towards a reasoning mechanism tackling complicated scenarios associated with engineering judgment or knowledge in engineering design and development process. For this reason, the emphasis is set on method and process development for creating a database and a pipeline for building a knowledge representation using state-of-the-art methods. As a consequence, only an overview of these methods within this field is given and a common thread is built towards

the selected strategy within this thesis. The subsequent evaluation of these methods is mandatory to allow adequate conclusions, so common evaluation approaches are described in a later part of this section.

The basic principle defining ML methods is to develop machines or models that can learn to perform specific tasks. These can be subdivided to tasks related to supervised, unsupervised, semi-supervised or reinforcement learning objectives. Thereof, the supervised learning field is closest to the problems described in this work. The aim is to digitally rebuild the engineering judgment and knowledge by learning from decisions which led to either desired or undesired results, thus learning from a labeled set of data. The prediction or model output in this case can be either discrete scalars, so a classification, or a continuous number as in a regression problem.

For the basic overview, some fundamental terms are explained next. The input data which is used to build supervised models is often denoted as \mathbf{X} of dimension $\mathbb{R}^{m \times n}$ with m being the number of training samples and n representing the feature count of the samples. In this context, the term feature describes important pieces of information of the input data and depict the input variables which shall be used to learn the correlation [238, pp. 26 sqq.]. The model M is now to evaluate and learn the relation between \mathbf{X} and the output \mathbf{y} which is of the dimensionality $\mathbb{N}^{m \times 1}$ in raw classification problems. For evaluation purposes, it is common to split \mathbf{X} in a separate training and test set. The training set is used to train the model, while the test set contains hitherto unseen data to evaluate the quality of the model predictions. A third set called validation set can be additionally introduced to cope with overfitting issues. Overfitting, or variance, and its counterpart underfitting, respectively bias, are two phenomena to describe the result quality of predictions.

The end-to-end process can be summarized to the major steps of data collection and preparation, model setup and training, the evaluation of the model and final tuning. Regarding the model itself, literature and research provide a plethora of methods. A well-known and straightforward approach is for example the decision tree or random forest technique. Based on training data, thus features, a tree which resembles rule and frame-based decision-making in expert systems, as described in section 2.6.1, is built and optimized [238, pp. 77 sqq.]. In more complicated and complex situations, these tree structures, however, either become too impractical or are not able to capture the present relations.

This is where (artificial) Neural Networks (NNs) come into play and convince with their potential. They provide enormous capabilities and extensions which make them suitable for a wide spread of applications, but often at the expense of interpretability.

Large neural networks with several (hidden) layers, being the center of focus of the deep learning field, are state-of-the-art methods for image, language and speech recognition, self-driving cars and a lot more applications. The basic scheme of NN is aligned to the working principle of the human brain: a construct of neurons and connections between them assigned with weighting factors, see figure 2.12.

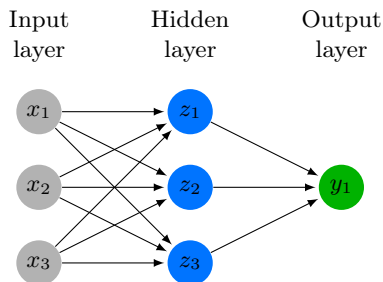


Figure 2.12.: Neural network scheme

The fundamental building blocks of a NN process are forward and backward propagation, neuron activation and model optimization with regard to a labeled dataset. The working principle and underlying math is considered as state-of-the-art which is beyond the scope of this work, and therefore reference to literature as [239–244] for further interest is given.

Besides model type, architecture, optimization methods or activation functions, other variables exert a noticeable influence on model quality. So called hyper-parameters are defined previous to the actual learning or optimization process and influence the model in form of the number of neurons in hidden layers, learning rates or specific layer-related parameters [244]. To ultimately evaluate the quality of the trained model, different metrics have emerged over the course of the years. An available performance metric is the loss function or value which is also used for the minimization objective in model optimization. For classification problems, accuracy is often related to the number of correctly predicted labels in relation to the number of samples, see equation 2.6.

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (2.6)$$

Going into more detail leads to terms as True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). These are derived from binary classification schemes but are also used for multi-class problems to evaluate the prediction quality and assess its reliability and robustness. A so-called confusion matrix can be filled with these values to provide a clear visualization of the prediction results [245]. Metrics like precision and recall are built upon these values, see equations 2.7 and 2.8. The F_1 score is another

widely used metric which can be derived and depicts the harmonic mean of the precision and recall [246, 247].

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (2.7)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2.8)$$

$$F_1 = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \quad (2.9)$$

In multi-class (N classes) problems, the F_1 score is extended to a macro, micro or weighted averaging score [248, 249], see equations 2.10-2.11.

$$F_{1,macro} = \frac{1}{N} \sum_{i=0}^N F_1(i) \quad (2.10)$$

$$F_{1,micro} = 2 * \left(\frac{\sum_{i=0}^N TP_i + FP_i}{\sum_{i=0}^N TP_i} + \frac{\sum_{i=0}^N TP_i + FN_i}{\sum_{i=0}^N TP_i} \right)^{-1} \quad (2.11)$$

with $N = \text{Number of Classes}$

Another important aspect in evaluating model quality is to assess model capacities and complexities, which are related to the already mentioned terms under- and overfitting. The figures 2.13 and 2.14 shall help to understand these phenomena by visualization. To overcome issues implied by high bias or high variance, different approaches have been developed to achieve a more optimal solution. Besides the described validation dataset in combination with an early stopping mechanism, methods like regularization or additional dropout layers in the model can help to improve model capacities. More information about these techniques can be found in [241].

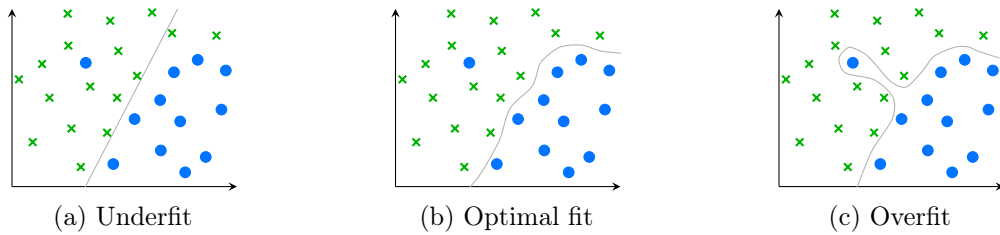


Figure 2.13.: Model capacity

Several subbranches have emerged from the field of neural networks for specific ap-

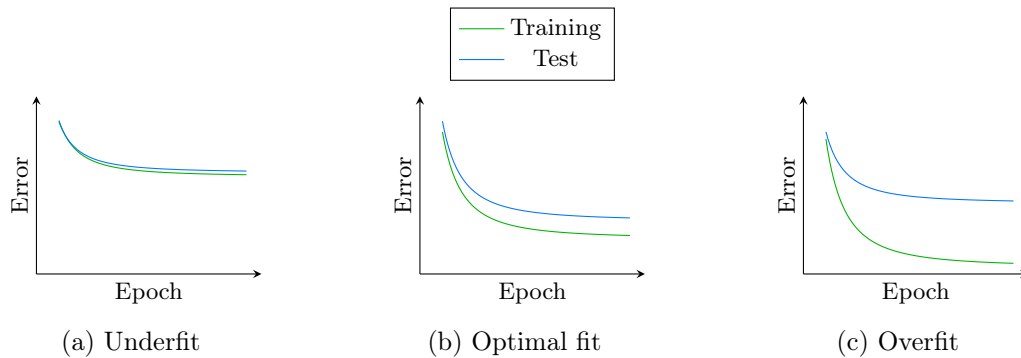


Figure 2.14.: Model complexity

plications as convolutional networks for image or shape recognition or recurrent neural networks for speech or language analysis. Due to their capabilities, methods from the machine learning field have also already found their way to the engineering domain.

General shape recognition and shape segmentation methods, which have already been described in short in section 2.5.1, are based on neural networks and specifically convolutional neural networks. Exemplary applications thereof have been described in [62, 74–76, 250]. Other authors describe the use of deep learning to build a surrogate model for a fast prediction of FE stress distributions on a bio-mechanical engineering example [251]. Also Roewer-Despres et al. [252] conduct research on a finite-element simulation using methods derived from the deep learning field. More focus on the elements themselves is set in works like [253, 254] who investigate smart finite elements which utilize machine learning logics. The application of elements is conducted in the meshing process which is analyzed with respect to ML techniques in [255].

In another context, Danglade et al. [256] utilize a machine learning approach to evaluate geometry defeaturing and simplifications for simulation purposes. The simplification scope there is mainly including manufacturing features as for example pockets, holes, blends and chamfers. In a later work [257], these fundamentals are used to build an approach for a priori estimating the impact of model simplifications as minor feature removals, face decimation or part filtering on analysis results using neural networks and decision trees. Kestel et al. [258, 259] developed an approach towards a knowledge-based FEA assistance system which shall support CAE engineering by machine learning algorithms. Data and text mining algorithms are processing expert interviews and validated models and shall support model setup and plausibility checks. For this purpose, CAE features have been developed to connect CAD and CAE environment and a user interface is embedded to manually acquire knowledge and to store it in the internal data structures.

Many of the mentioned methods rely on a consistent input data format and shape. The 3D model segmentation model described in [67] for example starts from models with ideally the same number of edges, general image recognition models recommend images being prepared to have an identical resolution and shape or decision trees relying on a uniform input format. However, unifying the data of complex systems is not practicable nor easy to achieve without leaving important relations behind. For this reason, another type of data structure is considered. Graph structures of form $G = \{\mathcal{V}, \mathcal{E}\}$ with \mathcal{V} representing vertices or nodes and \mathcal{E} depicting connecting edges are representing the input for Graph Neural Network (GNN) which are another subbranch in the field of deep learning. More detailed algorithms can also work with structures as $G = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$ with \mathbf{X} containing additional node features. Starting from this point, the adjacency matrix \mathbf{A} is built which is a symmetric matrix containing node connections in undirected graphs and depicts a second important input structure besides \mathbf{X} .

A lot of recent work is conducted in the GNN field, which is gaining more and more popularity. The methodology covers applications for various purposes as text or image analysis as well as science related topics like physics, proteins or molecular fingerprints analysis [260]. The field of GNN is quite extensive and their working principles significantly exceed the scope of this thesis. However, GNN have been used for a conceptual application case in this work and thus, a short introduction as given. For this purpose, reference to the work of Zhou et al. [260] is given, who provide a review of methods and applications of GNN. A comparison of several GNN methods is given by Errica et al. in [261]. Jandaghi et al. [262] additionally provide information on graph learning with neural networks.

Exemplary applications of GNN in the physics domain which also emphasize the use cases of computer science methods in engineering are described in [263, 264]. In [263], Battaglia et al. from the Google DeepMind team propose interaction networks which shall learn about objects, relations and also physics. Park et al. [264] describe a more practical-oriented use case for wind-farms in order to improve their efficiency as a whole system using GNN.

In this thesis, a specific focus is set on the work of Zhang et al. [265] who developed a GNN including hierarchical graph pooling and structure learning mechanisms. According to the authors, the methods embedded in the architecture help to aggregate node information in a hierarchical way, contrary to other GNN approaches. By this, nodes which play a different role in the system shall contribute to the graph level representation in a different way. The authors provide the example of protein-protein interactions graphs, in which specific substructures represent functionalities which are of greater importance to the whole system characteristics than other substructures. This aspect alone implies

an interesting analogy to engineering problems and consequently, promises an attractive potential for this domain. Finally, they present a benchmark with six widely used datasets, which result in an accuracy of 68 % to 85 %. Ultimately, they provided the source code of their algorithms in [265].

3. Process overview

3.1. Scientific gap

Simulation models based on CAD geometry are a crucial building block in the design and development process of products. Especially in early design phases, the ability to quickly adapt to design changes, to gather knowledge and understanding about system and model characteristics and achieve manageable models and data is decisive. In this regard, automation is a key aspect to build simulation models for the imposed needs while improving the efficiency.

As it can be seen in chapter 2, a lot of research and work emphasizes the weight of this objective or partial aspects thereof. Different approaches towards automated model transfer and simplifications to reduce simulation complexity, thus supporting design responsiveness, have been developed and analyzed.

In the aero engine context, especially the field of dimensional reduction is attracting attention. However, a straightforward 3D Medial Object approach to transfer a component to a sheet has shown to be not efficient or robust enough or not leading to the desired results. Forcing regions into a shell representation which are not reasonable targets for simplifications can lead to undesirable drawbacks in model quality. Moreover, geometric entities derived from the 3D Medial object process are likely to have a coarse topology, which in turn can result in an unnecessary number of elements or an undesirable mesh resolution. This proves counterproductive for the objective of reducing simulation complexity. Furthermore, 3D Medial object calculation can become quite extensive as described in [145, 163, 173].

The face-pairing approach is able to overcome some limitations and produces cleaner topology due to the proximity to the B-Rep ontology. The fundamental methodology has been adapted and integrated in commercial CAx software like Siemens NX for this reason. Nevertheless, retrieving suitable pairs in complex geometries is hard to achieve, neither easy is the adequate combination of separated midsurfaces in complicated geometries afterwards. A lot of surface matching and merging in the manual process is dependent on visual identification and visual validation of midsurfaces which is an aspect hard to

automate. This and other limitations are further mentioned in [172, 185].

Model decomposition methods, see section 2.5.1, pursue the goal to decompose the target geometry into better process-able structures and apply simplification techniques to them. Decompositioning and its implied complexity, however, depends on model topology and due to this can entail several limitations as well. Convex-hull oriented methods are not guaranteed to converge especially in complex curved or quasi-axisymmetric cases and consequently, to achieve a satisfactory result. Another approach to identify regions for splitting geometry is using concavity information in the form of islands and loops of concave edges. On the one hand, these are not expected to be given in every case, and, on the other hand, splitting based on complicated loops or subsequent closing of existing holes is either hard or not even possible to automate [127, 182]. Similarly, if graph or topology-based approaches are able to gain information about substructures, creating appropriate splitting procedures also represents a problem. Another issue is that decompositioning in complex cases, if possible, can lead to a high number of minor disconnected substructures which are, nonetheless, contextually dependent. Consequently, a too detailed process will result in losing this context, which can in turn be mandatory for the identification or downstream design process. An exemplary application would be to decide for and implement a suitable strategy out of different types of simplification approaches and the following combination and setup within the model scope.

Moreover, not every substructure or component is suitable for dimensional reduction or general simplification. Forcing those into simplified representations requires tedious manual effort and tuning and validation steps, which depict an obstacle for automation. Setting this effort into contrast to its advantages leads to the conclusion that dimensional reduction is suitable and recommended in appropriate scenarios and a more standardized 3D approach without extra tuning and validation effort is more reasonable in non-feasible cases, especially in times of rising computational power.

Moving from component to assembly level reveals also boundary conditions as important building block of simulation models, thus targets for constructs and simplifying assumptions for the final simulation model. The decision for boundary condition integration is relying on their type, the simulation objective, the desired level of detail, the required interfaces and local company best-practices. In the scope of this work, so for automation, retrieval and identification of geometric entities for these boundary conditions are cardinal points of interest. Exploiting integrated information in the form of databases, Product Lifecycle Management (PLM) systems, assembly features or CAD internal tags outlines a straight strategy but requires effort and knowledge prior to the process itself. Updating the required information after geometric or assembly modifications can become

undesirably tedious what depicts another potential disadvantage of this approach.

3.2. Strategy

These investigations uncover gaps for the goal to automate the setup of simulation models for desired needs and requirements. In this regard, this thesis is aiming for the development of methods to overcome existing limitations and further exploit model data for automation purposes and use cases.

For finding a holistic approach for the automation of the setup and simulation process, the idea of this work is to take one step back and to elaborate the manual process in order to understand key interfaces and points within the workflow.

The task of building a suitable simulation model depends on various aspects. The first step is often to visually analyze an assembly model. On account of existing knowledge about the engineering discipline, thus for the present system, conclusions can be drawn about assembly and component relations which guide subsequent cognitive actions. The recognized entities are analyzed and then to be transferred to adequate CAE representations with a, if applicable, simplification intention. Decisions for these simplification or conversion steps are made based upon a combination of visual analysis, physical understanding and experience gathered from last design iterations or previous system evaluations and validations. After the orchestrating framework has been defined, the processes are to be executed manually and the final model has to be assembled using the identified relations and transferred models. From a general point of view, the major blocks can be summarized to visual analysis, knowledge and manual execution as depicted in figure 3.1. In analogy to these findings, Ullman describes major aspects in the design process as sensors, for example visual sensors, short-term and long-term memory in [2, pp. 47 sqq.]. When transferring this to the present context, the short-term memory corresponds to the knowledge about for example the context of a standalone component within the previously analyzed assembly while long-term memory refers to basic engineering and physical judgment, understanding and reasoning and gained experience.

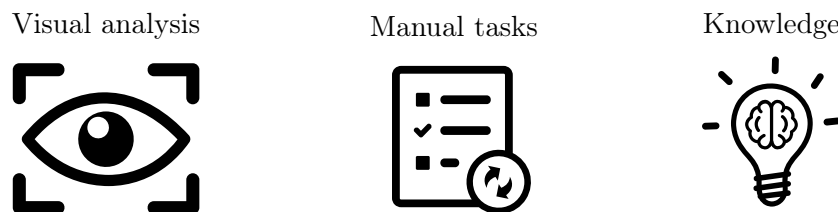


Figure 3.1.: Main elements of the design process

These considerations sparked the idea for the strategy developed within this thesis and guide the process development. The research in this paper is dedicated to methods which digitally imitate the steps and logics in this process with respect to the automation intention.

A recognition framework is developed to build the fundamental assembly, component and boundary condition knowledge and provide it as a form of short-term memory throughout the process. The subsequent step of component transfer is depending on component potential and suitability for simplifications. For this reason, component categories are introduced in order to provide specifically optimized, adapted and automated process pipelines for each category. The process used for the transfer of the casing category for example is also utilizing an adapted method for component segmentation which shall, on the one hand, imitate the visual analysis, and on the other hand extract information which can be used for mimicking engineering analysis and logics. Modifications of geometric structures and boundary conditions can be associated to the built short-term memory, and so can be tracked for further use in downstream processes as assembly model setup or post-processing.

The third aspect, long-term memory, is related to experience and knowledge. Automation can prove its necessity and advantages in this regard as well. This work outlines an approach to utilize the presented automated framework in order to automatically build uniform and centralized training data which can be used for data science applications afterwards. A learning algorithm from the field of Deep Learning (DL) is used to construct an artificial knowledge representation thereof which shall help to make design decisions in the process, similar to the manual process conducted by an expert engineer.

4. Recognition framework

The recognition of systems and subsystems, structures and components is a cardinal information basis information for the process, thus as well for automation. This as an objective is pursued by aligning with the human manual process, which is elaborated and broken down into small and manageable steps and logics.

In multiple sub-fields in section 2.5, methods associated with the term recognition have been discussed. The concrete recognition, as in for example model re-use tasks, is not building a suitable basis for the objective of this work. Aero engines depict highly optimized systems where a majority of designs is often adapted to present circumstances, thus non-standard, and by this affect the use case for model re-use applications. Methods from the field of computer vision deal with the recognition aspect, like in 2D image or 3D shape recognition topics. Besides the problem of recognition of 3D entities in a two-dimensional space and returning the 2D information to the 3D CAD environment, changes and modifications of a known geometry can imply that this geometry is either falsely labeled or not recognized at all, so have a negative influence on process robustness. For this reason, a purely shape and topology-based approach is either not robust or not meeting the posed challenges.

The goal and strategy of this work does not set the focus on a concrete recognition of every single entity in the first place. The fundamental of this approach is to mimic the process of an engineer's way of thoughts during the development of a structural whole engine model from a geometry. So, the aim is to understand how information access is working in this context and in which way this information is processed as such to derive conclusions from the present geometry. This reveals that, additionally to shape properties, semantic-driven and hierarchical conclusions are expedient for the geometry investigation task.

A recognition framework has been developed which is built upon three different fundamentals to cover the aspects of recognition, see figure 4.1: shape, function and contextual or positional knowledge. For each area, specific methods have been developed to access, analyze and connect information.

Similar to the human process, some tasks rely more on one aspect than on another but

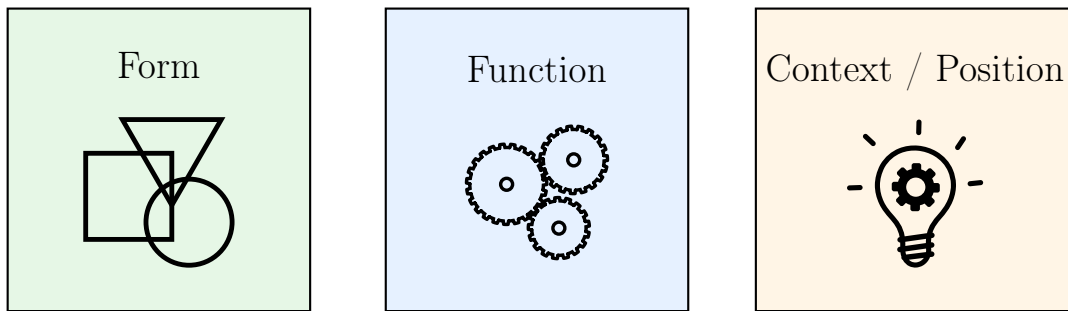


Figure 4.1.: Framework fundamentals

this framework allows a flexible utilization and combination of methods. Components which are mainly assigned to a specific function can be identified even if their shape and geometry is altered by targeting the functional semantics. Other entities which depict standardized objects can be analyzed with a focus on shape and topological properties. Nevertheless, functional or form-related information is not sufficient for recognition in some specific cases. Here, positional or contextual information, as knowledge about basic aero engine architecture, is accessed and combined with present data. For the implementation, the software Siemens NX and its API NX Open has been used to build this framework as an holistic approach within a lean process landscape.

4.1. Interfaces

A fundamental part of functions and context is associated to, thus embedded in, interfaces. The interfaces can occur in different types and arrangements: either a contact, an interference or a clearance. In the field of structural analyses, and with regard to related boundary conditions, contacts depict a cardinal part for the automation of model setup process.

Identification of interfaces has been a topic to research as described in the previous chapter. Besides performance considerations, a lean process landscape shall ease the use and increase usability of the developed methods. Modern commercial software packages like Siemens NX often provide methods and functions to retrieve contacts. Unfortunately, the information which can be accessed by these is not completely meeting the requirements of the application in this work. Geometric tolerances, discrepancies and different strategies in modeling do not guarantee a consistent setup, so a uniform model access. This is depicting an obstacle for predefined functions. For this purpose, existing capabilities have been extended and additional procedures have been developed.

The contact condition can be further subdivided into three subtypes: point, line or surface contact. Especially surface contacts are key for boundary conditions in the simulation model, therefore depict the target for the extended capabilities. The contact identification process starts with a component analysis of the involved parts based on their topology, more precisely their geometric faces. Contact evaluation can become a performance-demanding process, and this is why the developed process makes use of several filter layers. To first avoid model complexity, the geometry is abstracted to bounding boxes. Three different types of bounding boxes have been implemented in internal classes, which are also provided in a framework library for further applications in this work. The first, Basic Bounding Box (BBox) figure 4.2a, is the most basic one using maximum dimensions according to the global coordinate system and is integrated and optimized with the NX Open API. The second figure 4.2b shows an Oriented Bounding Box (OBox) which is similar to the BBox except for a coordinate system oriented along the elaborated principal axes of the target body. Due to the fundamental architecture of aero engines, another construct is introduced exploiting the dominant geometry orientation in the form of a Cylindrical Bounding Box (CBox), see figure 4.2c. To build this, geometry entities are discretized to point clouds which are then transformed to the CBox using a cylindrical reference system, most commonly aligned to the component or engine axis.

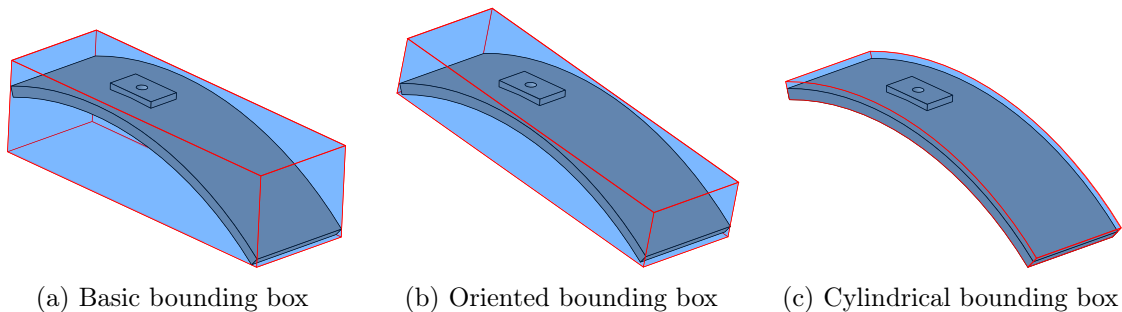


Figure 4.2.: Bounding box types

Starting with a set of bodies in an assembly, the first step is to evaluate potential pairs of bodies P_B which share interfaces. To improve performance and neglect unnecessary evaluations, the pairs are filtered based on proximity analyses first, see algorithm A.1. The parameter ϵ has been introduced as distance tolerance for the search algorithm. The bounding boxes depict an efficient approach for a first estimation, yielding a set of potential body pairs in contact. The next step is to investigate the contact conditions of these in detail which is based upon face arrangements, thus face pairs.

For the analysis of contacts, the objective is to retrieve face contacts of importance for

the structural analysis purposes. Especially planar and cylindrical interfaces play a crucial role in major boundary conditions, which is why emphasis is set on these arrangements. For each member in the derived set of body pairs \mathcal{P}_B a detailed face pair evaluation is conducted, see figure 4.3 and 4.4.

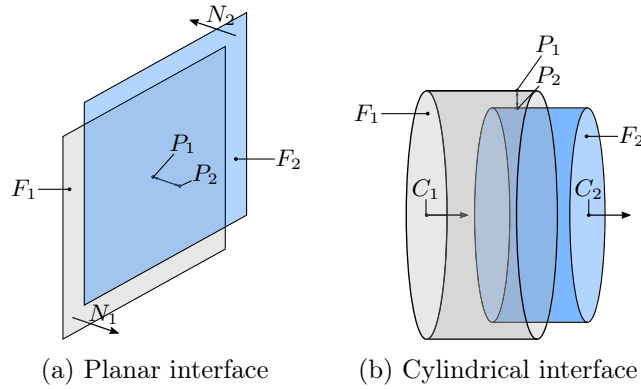


Figure 4.3.: Interfaces

After confirming a consistent surface type, a distance filtering similar to algorithm A.1 is done which yields the points defining the shortest distance between the two entities P_1 and P_2 . Different face properties are considered next depending on the face type. In planar cases, the face normal vectors are evaluated while cylinder axes and cylinder center points C_1, C_2 are important for the cylindrical case. Comparing face vectors with the vector of shortest distance reveals first information about overlapping properties. After these basic evaluations, the topology, so internal boundaries as edges, are mapped to further assess a potential touching condition. By this approach, also boundary conditions which come along with modeling discrepancies can be taken into account. Methods similar to the described ones have been implemented for other types of interfaces.

As a consequence, each contact body pair construct $P_B = (B_i, B_j)$ in \mathcal{P}_B contains a further set of contact face pairs \mathcal{P}_F . Each face pair is connected to further interface properties, which can be accessed via this construct afterwards. This layout leads to a graph resembling layout $G = \{\mathcal{V}, \mathcal{E}\}$ where G describing the graph with vertices \mathcal{V} and edges \mathcal{E} . In this context, the vertices are CAD body nodes while the connecting edges are representing one or more interface constructs. Setting up this graph representations results in the interface network, see figure 4.5.

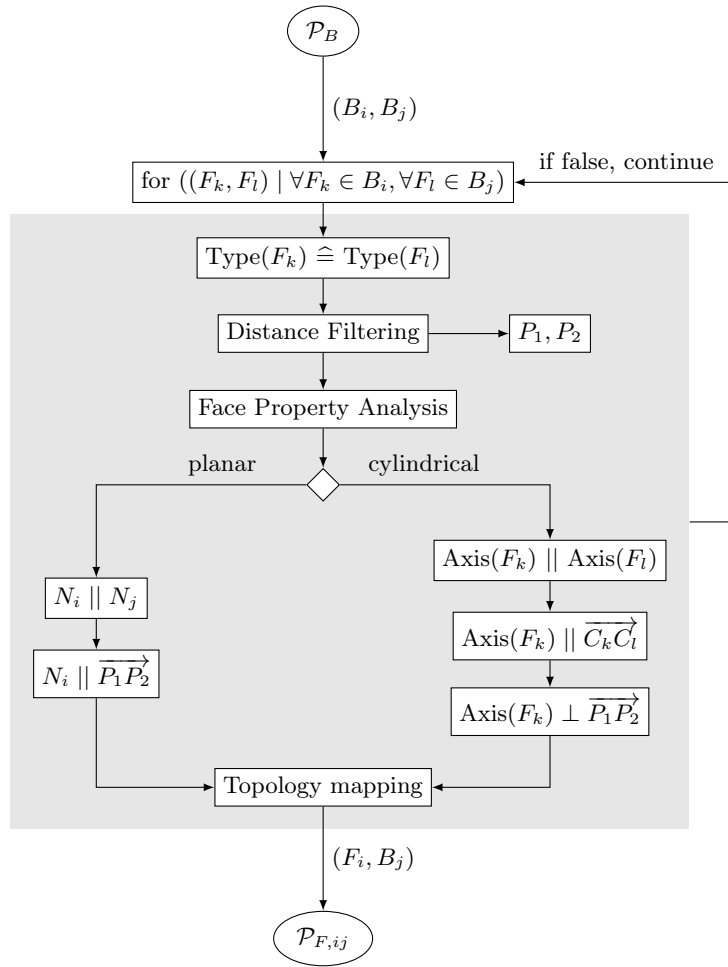


Figure 4.4.: Contact face process

4.2. Recognition

The interface network is representing a key aspect in the recognition framework. The next step is to set the objectives for the recognition process itself. The focus is set on entities and conditions which are either requisites for process automation, for model preparation or for a more sophisticated and solution-oriented post-processing. Investigating the manual process of setting up a structural analysis model reveals boundary conditions as important part for assembling the simulation model, implementing adequate constructs and access point for post-processing and model evaluation. Flanges are for example depicting such boundary conditions which are key elements in aero engine systems. The load on flanges is a critical aspect for the design or essential for subsystem modeling. Similarly, bearings

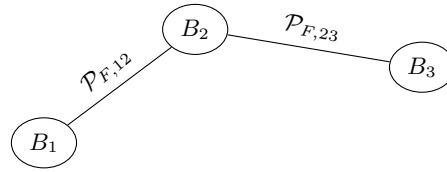


Figure 4.5.: Schematic interface network

also provide necessary information about the interface between rotating and stationary subsystems. In the common case, the rotating part is considered in a different simulation path which accounts for rotordynamic effects while the resulting loads are applied via the bearing interfaces in the static domain. The identification and subsequent specific consideration of static and rotating groups is consequently another major way point in the process.

Although computational power is increasing, the complexity of models is still often a driver for the necessity for the simplification of models. This simplification benefits model responsiveness, analysis effectiveness and model or data manageability. Besides simplification of boundary conditions, also other types of model preparation, as already mentioned in previous chapter 2, are points of interest. However, not every component is suitable for simplifications in this regard. Forcing a simplification can result in unreasonable model modifications which lead to undesirable simulation deviations on the one hand, and, on the other hand, the modifications themselves can require disproportionate amount of effort which in turn questions the simplification purpose and depicts hurdles for automation.

In lights of these considerations and the fact of increasing computational power, this thesis paves the way towards a process which decides itself if a component implies an attractive potential for simplification, or if a simplification is worth the effort respectively, and, if this is the case, proceeds in analyzing and preparing the component. Similar to the manual process, this first estimate is done based on recognition and component categorization. Different categories are introduced to streamline the process and set up specific processes for each category. This allows more specifically optimized workflows and mitigates the problems of robustness or limited capabilities in forced generic approaches.

Furthermore, categorization bypasses the discussed problem of recognition. This workflow is depicted schematically in figure 4.6. After boundary conditions have been investigated, the component categorization is taking place. The categorized parts are then directed to adapted workflows which process the components. Afterwards, the processed models are assembled using the identified boundary conditions. An important aspect that

has to be taken into account is that, for example, boundary conditions can change due to model modifications. For this reason, a database continuously keeps track of all changes and adapts entries accordingly throughout the process.

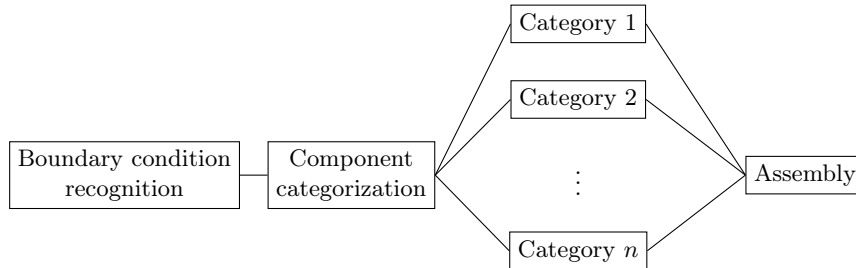


Figure 4.6.: Process workflow

4.2.1. Bolts

The first entity of interest in this work is a comparably straightforward object: bolts. Bolt connections play an important role for introducing adequate boundary conditions and an interface for extracting information in the post-processing. Interface loads are often critical values for the design as well as for subsystem modeling.

The major hint for recognition is contained in the aspect of norm and standardization of these objects. In analogy to the user logic, the identification points have been identified in a combination of dimensional aspects, screw thread and tool interfaces. From a CAD perspective, threads are often reduced to symbolic threads to keep complexity low, making the thread a less conclusive aspect. The tool interfaces, however, remain per definition and assembly logic, thus represent a consistent target for the algorithm. To evaluate this arrangement of faces, a conversion to a topology-based graph (FAG) is done, see figure 4.7. In the scope of this thesis, double-hex bolts and nuts are selected as demonstration example, but extending the capabilities to other head types can be achieved accordingly.

The patterned faces are retrieved from this graph by adjacency information and face properties, so the angles and concavity data between neighboring faces. Taking all this information into account and subsequent processing leads to an efficient identification of the bolt head, or the tool face pattern in general. Starting from this first access point, other properties of the screw or bolt can be retrieved by investigating other face constellations, as the bolt axis, length or diameter. Similarly, the double-hex nut counterpart possesses a similar pattern. After retrieving the axis, a ray tracing algorithm along this axis is examining the presence of a hollow structure condition and combined with dimensional properties, a distinction between bolt and nut can be achieved. The ray tracing process

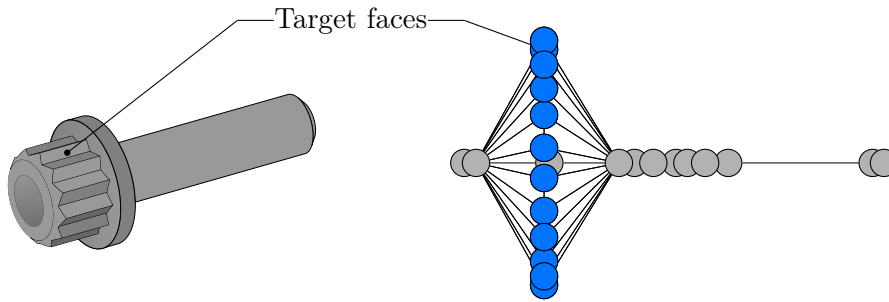


Figure 4.7.: Bolt topology and graph

uses an origin and a direction and is an efficient way to calculate hit points of the created ray and specified targets.

4.2.2. Flanges

Closely related to the bolt connection category is a specific type of interface which implies key objects for design automation. As already discussed previously, flanges are important in aero engine design. Besides being target for different modeling strategies, they inherit major boundary conditions and are depicting the access points to extract simulation results and loads for both design evaluation and downstream applications as subsystem modeling. Due to this, the identification of flanges brings advantages for streamlining and automating the procedure.

For the identification, major aspects and functions of flanges are put into focus. Considering the three recognition fundamentals, the function takes a meaningful role besides topology. A geometry segment containing a planar topology with holes must not necessarily represent a flange and often requires additional contextual knowledge. Furthermore, without a present flange counterpart, so no flange connection, the downstream use case of identifying a standalone flange is comparably less solution-oriented. The function argument introduces additional reliability, thus robustness, to the shape aspect for recognition.

One function is incorporating the interface for bolt connections, thus for the connection of components. The second prevalent function is the containment of fluids at these interfaces. Examples are the major flanges in aero engine systems, which connect casings and other structural components while surrounding an inner gas path. Besides, minor flanges can be found at piping and utilities installations. Due to assembly reasons, casings are often split in two halves, what in turn leads to separated flanges. This split also necessitates additional flanges in the component cardinal direction to ensure fluid containment. In conclusion, three different types of flanges have been identified in this

research. The standard flange, figure 4.8a, is a flange which is surrounding an inner path for, for example, fluids. The split flange, figure 4.8b, is the result of splitting a standard flange into two or more sections which together form a path surrounding standard flange. Other flanges which do not give a hint on any enclosing function are specified as general flanges, see figure 4.8c for example.

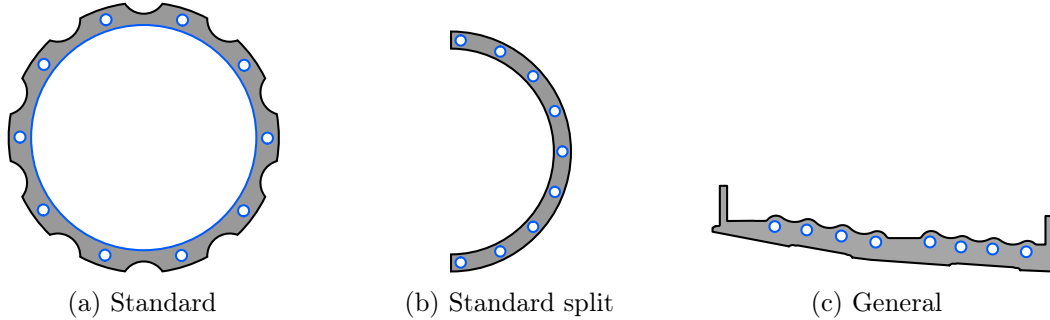


Figure 4.8.: Flange types

The entry point is provided by the interface function and thus, the interface network. Planar interfaces and associated face pairs are retrieved thereof and are subject to furthermore detailed analyses. Another hint on flanges is embedded in the bolt connection and the associated holes. For this purpose, a topological analysis of the interface faces is done. In general, faces can be bounded by outer and inner edges. Outer edges are colored black and inner boundaries are visualized in blue in figure 4.8. These edge loops are stored as objects associated with attributes $E(T_T, T_L, C, D)$ with T_T describing the topological edge type, T_L the type of the edge or the loop, as e.g. single circular edge, C being the center and D representing the minimum hole size or radius in case of a simple circle.

As it can be seen in figure 4.8b, the characteristic path surrounding inner loop is connected to the outer topological entity, forming a surrounding outer edge loop. This in turn prevents information about an internal fluid or gas path. The interface network provides more information for this purpose with its contained face pair data. If this split flange is in contact with for example a standard flange, it can be concluded that this standard flange is in contact with two split flanges. By this, the counterpart to the split flange can be accessed. Both face topologies can be virtually merged, leading to an overall outer edge loop, an inner loop and connecting edges, see figure 4.9.

The set of inner loops is then analyzed with respect to potential inner paths to distinguish general loops from holes for bolts. Flange center points and loop properties guide the selection of the inner load path and so yield potential connection holes. If there

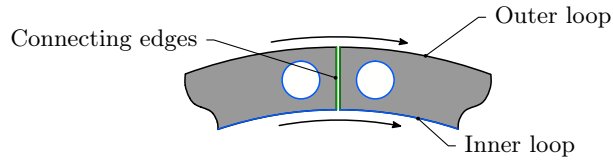


Figure 4.9.: Split flange merging

is no suitable inner path, the process categorizes the flange as general flange. The next step in the flange analysis process is to focus on their fundamental function of describing a connection by bolts. Based on the virtual and digital representations of the flanges, the internal holes of related flanges can be mapped towards their counterpart, which results in a set of coincident hole pairs. Based on hole type T_L and hole size D , the hole pairs are grouped which allows subsequent pattern and hole type analysis, as for example to separate from centering bores which are less of importance for the analyses in this scope.

All these 2D analyses are extended to a 3D space to account for additional flange properties. Flange thickness and height are distinctive features for their visual recognition. For the retrieval of the flange thickness, an algorithm which crawls through body faces and properties has been developed. Analogously to the Point-In-Polygon (PIP) problem [266], the rear flange faces are evaluated by using a ray-tracing algorithm, see algorithm A.2. After some guard clauses similar to those already described, a ray tracing algorithm retrieves and counts the hit points \mathcal{H} with body B to elaborate if the rear face is representing a potential counterpart to the flange face. From this set of rear faces \mathcal{F}_R , an approximated flange thickness can be calculated.

Both inner and outer boundaries of the front and rear face are discretized to separated point sets, which are used to calculate the shortest distances thereof for the approximation of flange height. The resulting distances are then averaged to build a representative virtual flange height.

The potential set of holes derived from the 2D topology analysis is now transferred to the 3D environment. On the basis of the knowledge about front and potential rear faces and the hole edge entity, an algorithm unveils the chain of hole inner faces until a rear face is reached in case of through-holes or until there is no further path in the topological graph which identifies them as blind holes. Ultimately, all surfaces and interfaces required for automatically setting up bolt connection constructs are collected and stored internally.

The described methods result in a set of flange objects and flange connections, which contain all gathered information as attributes for potential use in downstream applications.

Apart from the mentioned scenarios, another arrangement can occur in assemblies. In

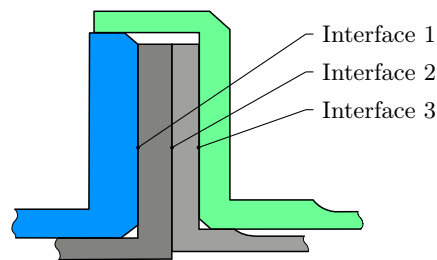


Figure 4.10.: 2D schematic flange chain

some cases, multiple components are assembled via a single flange connection consisting of multiple flanges as shown in schematic form in 4.10. This case shows three different interfaces which lead to three flange connections at the current state. The rear face of one flange depicts the interface face to another connection. Based on that, identifying the flange rear faces allows mapping the single flange connections towards each other. A crawling algorithm then detects the chain of flanges and creates a top-level construct named flange chain, which contains the single flanges and flange connections as children. Besides generating a better model understanding, this also helps to extend the standalone hole pairs to a hole chain through all flanges, required for setting up suitable bolt construct interfaces. The internal objects allow for example a sorting according to a cylindrical coordinate system for a consistent setup and post-processing throughout the process and models.

This recognition process builds a database containing information about required and optional entities for setting up boundary conditions in the simulation model. The internal objects provide information about contact faces or face chains and also hole related data as hole faces and edges, diameters, centers which can be accessed in subsequent processes.

4.2.3. Bearings

Another crucial interface for simulations is attached to bearings. A basic bearing fulfills per definition the purpose of providing a kinematic connection between two systems, which can have different rotational speeds, and provide a theoretically free rotation around a fixed axis. Different rotational speeds, where one system can also be static without any rotational movement, means different simulation paths, and therefore this interface has to be considered in the analysis. In the common case in preliminary structural analysis, the bearing is modeled with a surrogate construct of different FE objects which allows transferring loads and motions between the participating systems. As a consequence, the identification of bearings within a system is mandatory for properly setting up this type

of interface in an automated way.

The process of visually identifying a bearing is based a lot on context investigation and logic. Due to the fact that a bearing consists of multiple bodies, with some of them being adapted to a present environment, purely shape-driven and component-focused approaches do not imply a sufficient degree of robustness. An example for this aspect is depicted using a bearing as shown in figure 4.11b. Compared to figure 4.11a, identifying the given model as a bearing is not straightforward without additional context or for people without engineering knowledge. The next logical step is to elaborate the interconnected system by taking a closer look at all participating components and interfaces, as it is done in the manual process by stepwise isolating bodies or switching to a wireframe view like shown in figure 4.11c.

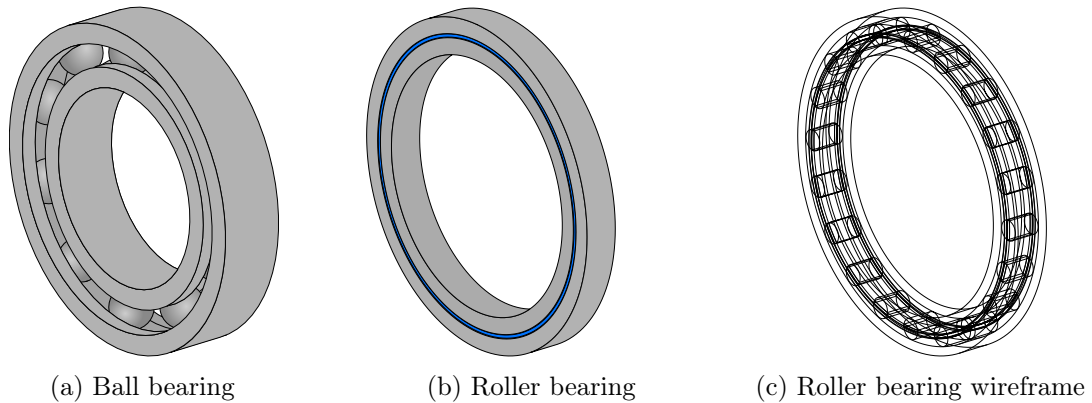


Figure 4.11.: Exemplary bearing layouts

The process for the automated identification is developed aligned to the logics identified in this way of thoughts. The analysis of the interfaces and components lead to the fundamental bearing graph structure as depicted in figure 4.12. The combination, arrangement and type of interfaces within this basic graph setup provides additional information which is considered to be representative for the main bearing functions. The function of a bearing will remain similar in different situations, regardless of individual component designs or optimizations. Due to this, the graph structure depicts an appropriate target for the automated identification process.

Another identification key point is the shape aspect of bearing bodies which are standardized, thus incorporate a consistent anchor for the graph and pattern matching process. In the scope of this work, the focus is set on spherical, so ball, and cylindrical roller bearing body types for identification. With these shapes being comparably straightforward and well accessible, a shape matching algorithm based on the FAG principle is able to

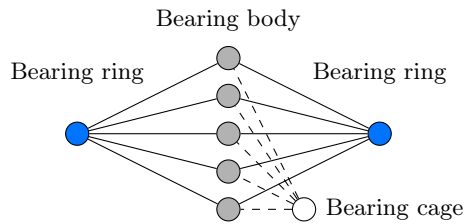


Figure 4.12.: Bearing graph

filter all potential bearing bodies from the set of bodies within the interface network. The access to the connected interfaces and relations supports an adjacency-based grouping of all bearing bodies. The interfaces are investigated group-wise in more detail regarding contact conditions. Afterwards, this information is compared and matched to the bearing graph scheme.

In case of matched sub-graphs, the process continues to extend the retrieved information. The bearing center and axis is calculated based on the bearing body arrangement. Furthermore, part neighbors reveal a potential existence of a bearing cage. The contacts and interfaces between bearing bodies, outer, inner and bearing cage are analyzed to distinguish between the different parts. The contact locations and types, e.g. point or linear contact, are used to send and trace rays in reference to the bearing center. The type and order of hit points on selected components is providing information to distinguish between inner and outer ring or a potential bearing cage. Kinematic assumptions based on the bearing contact conditions also allow conclusions about fixed or floating bearing variants.

The identification of the bearing rings provides access to the integration interfaces within the assembly, which are mandatory for setting up adequate simulation boundary conditions. In some cases, bearings are integrated into minor structures for assembly reasons, which exclusively fulfill the function of building a bearing structure. A cylindrical bounding box analysis is evaluating the dimensions of these adjacent components and if they can be approximated to be a part of the bearing itself, thus reduce complexity in the system. All the derived information as bearing parts, axis and system interfaces is stored internally in structured classes for direct access later in the process.

4.2.4. Rotary Groups

The identification of bearings allows drawing logical conclusions by exploiting the gathered information for the systems connected by the interface. In analogy to the procedure of analyzing a general arrangement drawing, the next step includes the separation of

the connected systems on account of the bearing information. This provides substantial aspects to re-engineer the knowledge which is embedded in the system architecture or the geometry assembly. The network which is the basis for these investigations is shown in a schematic view in figure 4.13 which includes the bearing sub-graphs as collapsed single entities connected to representative neighbors.

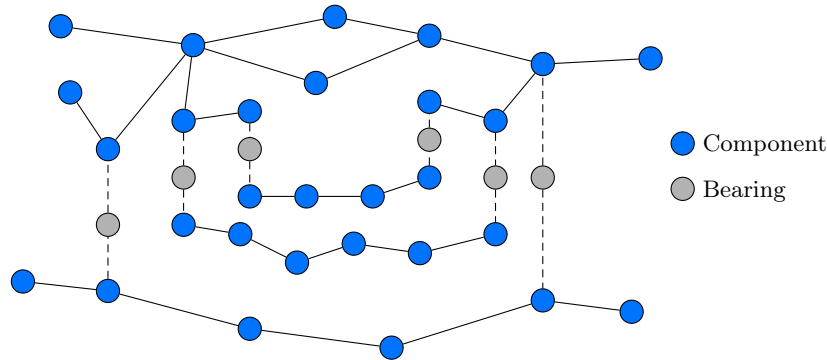


Figure 4.13.: Rotary groups scheme

Path-finding algorithms crawl through the nodes and edges in the interface network, retrieving sub-graphs where boundaries are set up by the bearings. The derived sub-systems are stored in containers and related to the inward and outward bearing objects, meeting the demand for system separation, including the retrieval of adequate interfaces.

From a semantic perspective, this opens the path to a multitude of further identification possibilities. Contextual knowledge can elevate the level of cognitive capabilities. With this thesis being situated in the field of aero engines, contextual knowledge about basic aero engine architecture is integrated in the presented recognition framework. The assembly consists of structural parts enclosing the gas path and rotating systems including rotors and shafts in general. In multi-shaft systems, different rotating systems are often associated with the related pressure field, as mentioned in 2.1. Within this frame of context, the low-pressure system is associated with the front section of the compressor, or fan, while the rear section is related to the rear turbine segment. This leads to the conclusion that the low-pressure system represents the inner system with the maximum axial dimension.

The retrieved rotary groups are analyzed regarding these aspects by abstracted bounding box strategies. Center axis alignment, bounding box dimensions as well as the containment within other rotary groups are criteria for the low-pressure system. Another example for semantic and contextual retrieval could be the drive shaft for the external gearbox, which represents another rotary group type. This system is similarly bounded by bearings and

related to a bounding box structure of slender type but an axis which is not aligned to the engine center axis.

From the low-pressure system as starting point, the bearings can be recursively traced to the next rotating system up to the static structural subsystem including mounting systems.

Conceptually separating the systems allows also streamlining the simulation paths, which is aligned with the manual procedure. The objectives of this work are set within the structural system, and therefore the next methods are working with the identified structural groups and the related systems.

4.2.5. Beam structures

The recognition aspect in the exclusive structural domain now puts a major emphasis on simplification opportunities. As mentioned in previous sections, dimensional reduction of FE entities is an effective method to reduce model and simulation complexity, and nonetheless achieves satisfactory result quality in suitable scenarios. Moreover, 1D FE elements imply a simple solution and tool for trimming and tuning properties in order to match real component behavior or conduct preliminary design studies. An inferred potential for these scenarios are geometries which resemble a beam-like structure, commonly long and slender structures, or respectively having two dimensions substantially smaller than the third. Consequently, the first step is to filter components based on this shape aspect with respect to the resulting potential in order to transfer them in an appropriate process pipeline to simplified structures.

The recognition or estimation of their potential is straightforward, utilizing the methods already presented. By calculating the principal axes of a given 3D object via methods provided by the API, a new virtual coordinate system can be instantiated. This coordinate system is taken as reference for setting up an oriented bounding box, see figure 4.14. Based on the derived dimensions, a first statement about the shape can be concluded.

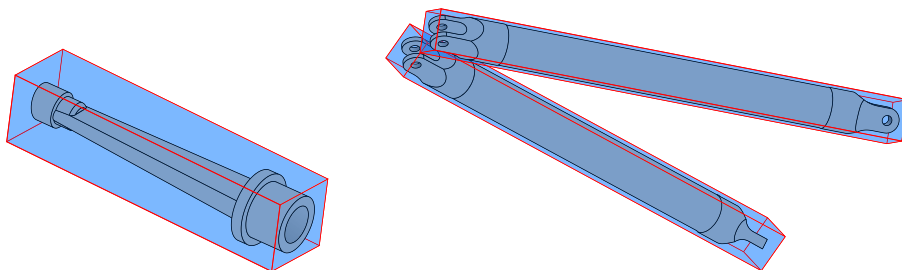


Figure 4.14.: Beam structure analysis

4.2.6. Casings

The simplification aspect is also central for the casing component category. Due to given circumstances in aero engine development as weight aspects, thin-walled and lightweight components are prerequisites for the design of competitive products. The fundamental working principle concludes casing structures to enclose the inner paths, contain pressures and provide interfaces for air systems and auxiliaries. These commonly thin-walled designed structures consequently make up a large proportion of the structural system. The thin-walled circumstance fits the assumptions of model simplification using 2D shell elements in this context. However, the general transfer of general geometries to suitable models of reduced dimensionality is often coupled to a lot of effort which in turn can result in the automated process being not efficient, not leading to the desired quality or even not being applicable.

In lights of these considerations, the strategy here is to put emphasis on those components which imply advantageous potential. With casing structures coming with such a potential, the framework aims for a recognition of components of this category type in order to introduce specifically optimized transfer processes for increasing procedure capabilities and precision but also robustness.

The basic points for the component categorization are the aspects of casing structures which have already been described. First of all, these structures are part of the structural static rotary group. The enclosing and surrounding aspect is elaborated by calculating the CBox and evaluating both axial, radial and angular dimensions. This aspect concludes that the basis topology is axisymmetric or at least close to axisymmetry. The topological characteristics of the bodies provide information about this and their orientation. Based on the characteristics of individual surfaces, their cardinal orientation is elaborated using the face axis or face normal in the planar case, which are then compared to the component axis. The surface area of the matched cases is summed up and put into relation to the overall body surface area. If this ratio is meeting a specified criterion γ , the component is added to the category set of casings \mathcal{B}_C .

Otherwise, if the ratio is below the tolerance but the component technically resembling a casing structure, the algorithm concludes that the level of axisymmetry is comparably low due to attached structures affecting this ratio. This implies additional complexity, which is in turn opposed to the simplification objective and suitability. Since the objective is to put more emphasis on parts with potential for a reasonable transfer process, the deselection of such parts is in line with the strategy.

The resulting set of casings is stored in the database and prepared for a more detailed

investigation and CAE transfer in subsequent pipelines.

4.3. Application

After describing the recognition framework, its embedded logics and methods, this section describes their implementation, inputs and outputs, and applications. The algorithms are developed on top of the API provided by the commercial software Siemens NX. The framework itself has been designed towards a modular architecture, allowing later application and business cases for recognition to be seamlessly integrated. Each recognition category comes with internal classes, factories and dedicated methods incorporating the modular structure and making subsequent access and use straightforward.

All information collected in the individual recognition activities is stored in a local database, which is provided to later process chains. In order to be easily accessible and modifiable or readable in general, the widely used Extensible Markup Language (XML) format has been selected. Binary databases have their advantages but depict unnecessary overload for the given case. The content of the XML database is separated to containers regarding boundary conditions and specific component categorization and descriptions. The boundary condition section contains every information about interfaces and also higher-level data as flanges and bearings. For example, each flange content contains information about participating components, related faces and also sub-elements which describe eventual hole chains including geometric entities and basic properties as radius and center. Bearings are added in a similar way, collecting all participating bodies and interfaces to neighboring systems. The basic idea is to store all data that could be potentially for use in other processes to allow a purely data-driven process and by this, digitally imitate the aspect of short-term knowledge.

The next set contains data about categorization and rotary groups. Part properties, part file location and data about occurrences in the assembly associated with offsets and rotation are stored within the respective element.

In the following, the developed methods are demonstrated on general example cases. After discussing the models and results, an evaluation of selected metrics of the process is conducted.

Demonstration case 1, or M1 in the following, represents a simplistic general combination of a rotary and static system with reference to aero engines. The starting point is depicted on the left in figure 4.15a as a raw geometry assembly without any additional information. The recognition framework is then taking over and analyzing the geometry. After having initialized all interfaces, the interface network, see figure 4.15b is set up and recognition

tasks are set next in the pipeline. The results of the framework are visualized in figure 4.15c. This figure shows the identified rotary groups, bolts, bearings and the flange connections which are illustrated using a flattened bounding box skeleton. Regarding flange modeling in the CAE environment, the bolt connection is coupled to the flange connection, thus both data streams can be utilized to elaborate a FE construct. In this scenario, only the center component has been categorized as casing with the other being theoretically thin-walled but too complex in terms of radial dimensions, bearing interfaces and structures to depict a low effort and high simplification potential casing type. As a consequence, these are selected for 3D transferal.

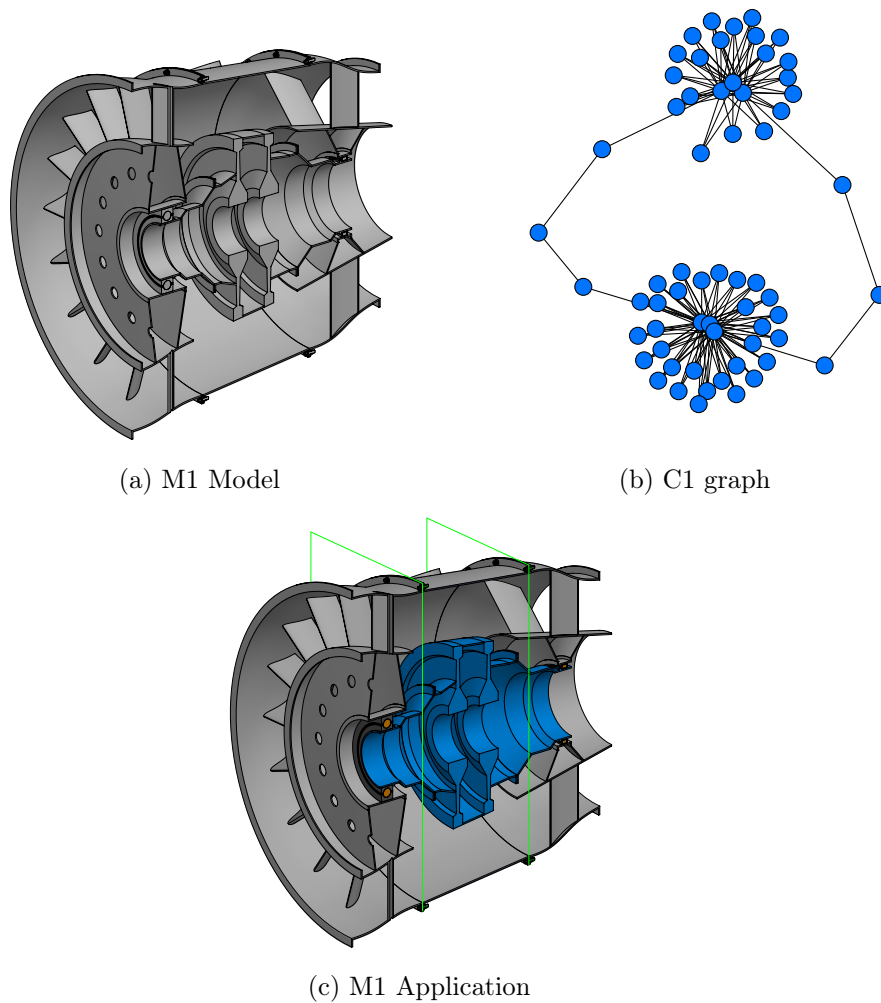


Figure 4.15.: M1 demo case

The next case M2 shows another simple assembly aligned to aero engines with extended

complexity and a second rotary system. The application and its results are shown in figure 4.16. Aside bearing, flange and rotary group recognition, this assembly also contains beam-like structures in the rear section. Similar to the M1 model, the bearing support structures are considered as general components, while the connecting casing line is recognized as expected.

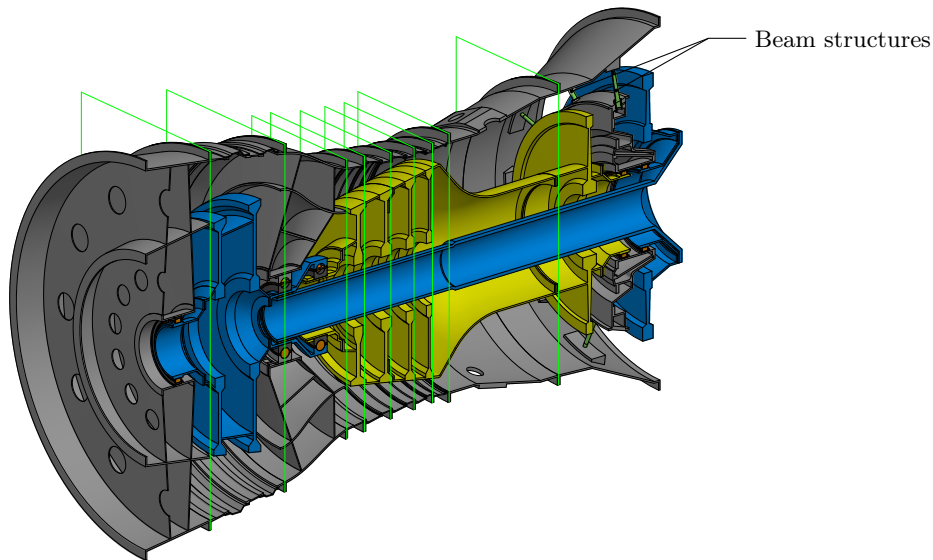


Figure 4.16.: M2 application

Demonstration cases M3 and M4, see figures 4.17 and 4.18, imply a higher level of detail and are designed for preliminary basic aero engine architecture studies. M3 represents a two shaft aero engine which is commonly used in civil small and medium engines. The model M4 in contrast contains three shafts which implies an architecture being more common in larger civil engines. The interface network generated for M4 is shown in addition in figure 4.18b. A detailed look is revealing the bearing patterns which have been described in previous sections.

Finally, different metrics and the performance of the application are put into focus in the last part of this section. The body count and number of body and face interferences are selected as measures for model complexity and size for this purpose. However, the demonstration case components themselves are designed comparably simple. Highly complex components can contain a lot more B-Rep entities, increasing their topological complexity which can affect application performance. The required computational time is averaged from a set of test runs and selected for evaluating the performance of the algorithms and the process itself. To assess the automation factor, the number of lines in

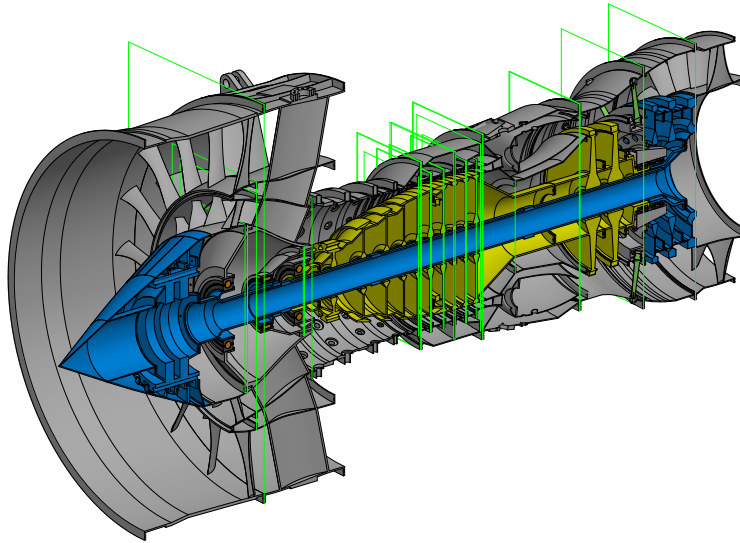


Figure 4.17.: M3 application

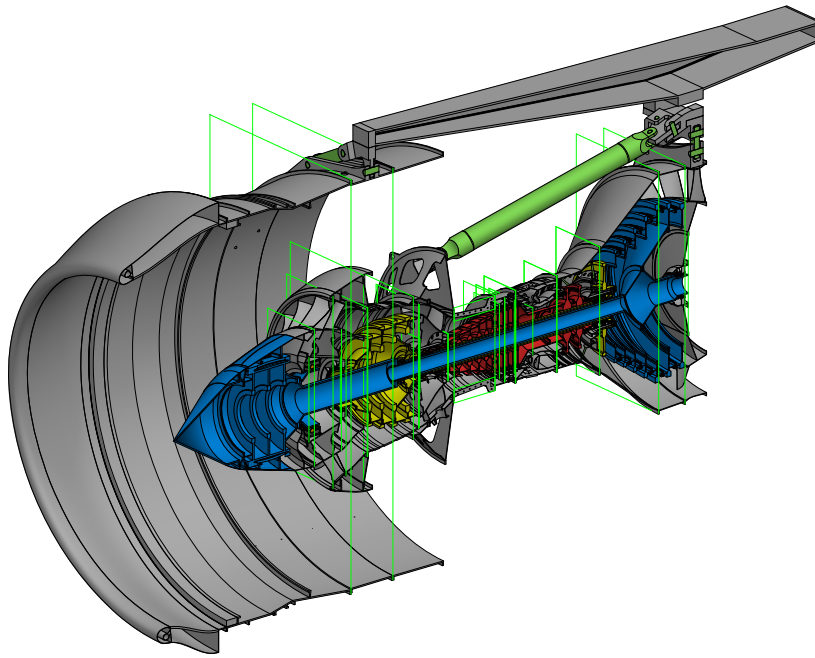
the resulting database is investigated as its content would have been also cardinal for the manual procedure. The results of these studies are shown in figure 4.19 and table 4.1.

Table 4.1.: Application metrics

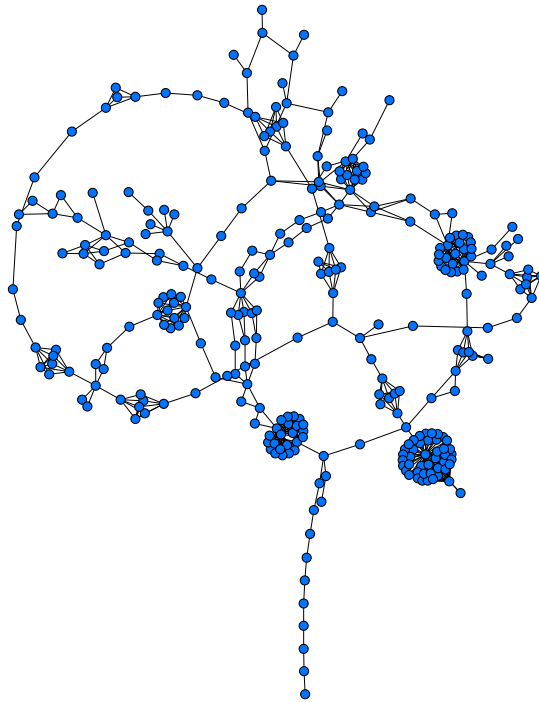
	Bodies	Interferences	Database size
M1	62	160	185
M2	171	426	911
M3	136	250	5630
M4	276	500	3823

The application of the recognition framework has been conducted using an AMD 3900X CPU and a NVIDIA Geforce 1060 3GB GPU. Despite the achieved performance reaching satisfactory levels and meeting the objectives, the maximum potential of modern CPUs including multi-core architectures is not completely exploited due to single-core limitations of common CAD kernels. From this perspective, a remarkable gain could be unlocked if newer technologies would allow to overcome this limitation.

The advantages of the automation are clearly visible in the database sizes. Especially in larger assemblies, the selection and derivation of all geometric objects required for the setup of different structures including the comprehensive knowledge about the present conditions can quickly become extensive. Moreover, since the models M2-M4 do not contain full-



(a) M4 result



(b) M4 graph

Figure 4.18.: M4 application

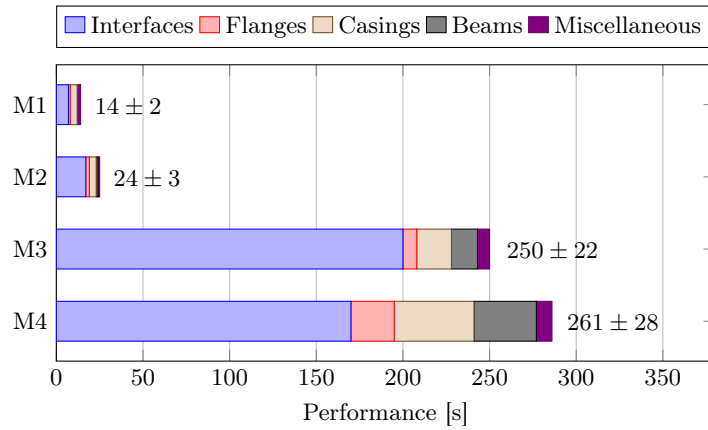


Figure 4.19.: Performance evaluation

featured flanges for example, the database size can be assumed to be significantly larger for more detailed scenarios.

In conclusion, the developed logics, algorithms and methods and the framework comprising these have proven their suitability for the objectives of this work. By categorizing components, specifically adapted process pipelines are introduced and assigned for the transfer to the simulation environment and adequate simulation models. Moreover, the presented approach also shows up a plethora of future applications for objectives beyond the scope of this thesis. The next logical step is to develop the transfer pipelines for the identified categories.

5. CAE transfer

The transfer of components from the CAD environment to suitable simulation models is the subsequent step in the development cycle. FE components, connection and coupling of models, boundary conditions, loads as well as simulation type and settings make up the largest parts of this transfer and setup process. The first two aspects depict the fundamentals of a basic model setup, and so for the objectives of this work.

CAD components often serve as basis for building a FEM mesh model, especially in CAD-CAE integrated systems. CAE software therefore have interfaces for the exchange with the CAD environment and provide methods to apply meshing algorithms to given B-Rep body or entity in general. In this context, each type of object, so 3D body or 2D surface or 1D curve, can be selected for internal meshing algorithms with their related class of elements.

With regard to simplification, there are two major paths which can decrease the model and simulation complexity, as described in section 2.5.1. First, there is the option to coarsen or adapt the mesh in order to reduce the number of elements. Secondly, the geometry to be meshed can be modified, prepared and idealized to simplify the geometry or allow elements of lower dimensionality.

Many of the state-of-the-art methods have already been adapted and integrated into available meshing algorithms. Siemens NX, for example, provides the option small feature tolerance which identifies and accounts for small geometric features and sets up an approximating mesh rather than a fine exact one within these areas [34]. A similar idea is embedded in a parameter called geometry tolerance which allows mesh vertices to be slightly off original geometry boundaries, introducing more flexibility to the algorithms, so less unnecessary mesh agglomerations.

The section 2.5.1 has presented several approaches to geometry simplification. Model preparation in terms of midsurface creation is a topic of a majority of them and different approaches to meet the challenges have been presented. A divide-and-conquer idea to split the geometry to access more handy sub-parts has proven its advantages in this regard. However, these geometry segmentation algorithms can imply drawbacks or restrictions as discussed in section 3.1.

This is where the benefits of component categories come into play. By focusing on specific types of components, the methods can be specifically adapted, and therefore extended in their capabilities and applicability. This allows also reducing and focusing the effort on components which are potentially worth a closer investigation. As mentioned before, casings make up a large percentage of aero engine assemblies while revealing this simplification potential what makes them a reasonable target for the development of simplification methods. Moreover, simplifying suitable beam structures requires comparably small effort while holding important model reduction profits.

5.1. Casings

The aspects which have been decisive for the recognition of casings, as quasi-axisymmetry, are also guiding the component analysis and method development. In order to understand the objectives and associated way of thoughts, the manual process of deriving a simplified model from this point of view is elaborated first. Similar to the recognition process, the first task includes a visual analysis and a first assessment of the major structures. Afterwards, general characteristics as axisymmetry are exploited to manually construct a midsurface model for the major structures – if applicable. Minor structures are analyzed regarding influence on model quality and suitable representations for the FE model are selected and manually introduced.

This concludes that the separated analysis of component substructures is a crucial step for subsequent purposes. The automated process which is presented in this thesis digitally mimics the logics in this path, and therefore sets segmentation as first cardinal building block for the transfer. Two different approaches tackling this aspect have been developed for this purpose.

5.1.1. Feature-based analysis

The first approach is focusing on the CAD structure itself. Modern CAD environments often imply history and feature-based modeling environments for model creation and further PLM interfaces. Especially for preliminary design development and studies, parametric designs in combination with a synthesis modeling strategy imply noticeable benefits. The opposing path to this is the destructive modeling technique, which is more popular in manufacturing domains [267, pp. 132 sqq.].

The structure of the aero engine casings is often designed as such that there is one or more major unconnected parts which are modified and equipped by minor or auxiliary

structures. Combined with the knowledge of axisymmetry, this concludes that commonly the main axisymmetric structures are created as starting point in synthetic modeling by, for example, revolving features.

This information is exploited in the feature-based segmentation approach. The strategy is to target these root bodies and track their changes throughout the model history. For this purpose, a structure has been set up which is different than the CAD internal feature history in providing more data required for the tasks.

The term feature state F is introduced and denoted with $F_{i,j}$ with i representing the state of structure i and j depicting the associated time stamp. This feature state contains information about associated features, the topology of the current state of the body and basic properties describing volume and dimensions. The collection of all feature states is stored in a feature tree as depicted in figure 5.1 which resembles the structure of a CSG tree.

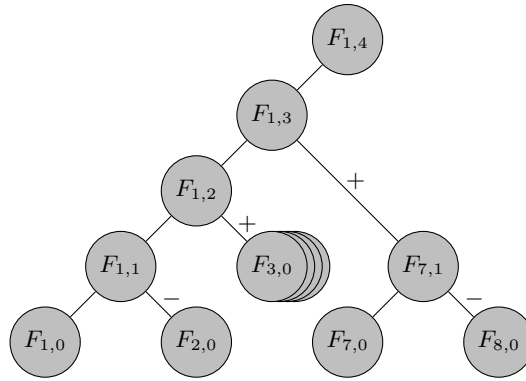


Figure 5.1.: Feature tree scheme

A change $C_{i,j}$ in the solid i is taking place from state $F_{i,j}$ to $F_{i,j+1}$ due to interaction with features. In general, these features are of boolean type, so additive (+) or subtractive (-). This results in a connection between the branch of solid i and another solid k . The resulting change $C_{i,j}$ is built as an internal construct which holds information about the interfaces between body i and k . This also implies changes in topology and so provides access to both state branches and information about a change in volume and dimension, which allows conclusions about the boolean type.

Detailed substructures can result in a higher number of separated bodies, thus states. This aspect introduces complexity which is in contradiction to the simplification objective and can blur the interpretability of the various data streams. The manual and visual investigation process hence often makes use of a logical abstraction process, especially if a simplification goal is pursued.

The developed methods offer opportunities to overcome this obstacle. By accessing information about dimensions, interfaces and also volume relations, groups of substructures can be united to a single substructure which is easier to deal with in the process. As a result, several feature states can be summarized, decreasing complexity of the feature tree. A similar effect is achieved by analyzing patterns, either by given pattern feature information or by analyzing the body properties and distributions.

These programming constructs offer straight access to all necessary interfaces for the later evaluation of substructures and the relation and connection between the single bodies, see figure 5.2. These relations offer a concrete and detailed access to the structure connection itself, and so also to the real cutting faces required for segmenting the structures, which would be more complicated to extract using state-of-the-art segmentation algorithms. This approach reveals suitable and reasonable splitting faces and relations even in more complicated scenarios. Moreover, by focusing on the change in topology and volume, a more detailed information about the feature used to instantiate or create the geometry is not mandatory.

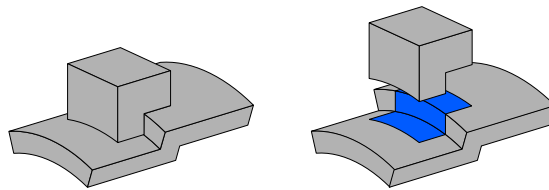


Figure 5.2.: Segmentation interface scheme

5.1.2. Silhouette-based analysis

An appropriate feature-based approach entails advantages as robustness, performance, model precision but also extended use cases like parametric design studies and also the possibility to integrate information in the model for later purposes. The feature setup strategy of CAD models, however, can vary depending on the individual designer, design objectives and design domain. These aspects and the implied effort to build such adequate models are representing contrasting arguments to the advantages. This leads to the conclusion that suitable feature-based models cannot be considered as guaranteed in more general situations.

In this regard, a second approach has been developed for suitable substructure segmentation and analysis, which utilizes a more general working principle. This technique exploits the given circumstances of quasi-axisymmetry and flat hierarchy of the model setup. Flat hierarchy in this context means that the major structure is expected to be a

more or less axisymmetric casing part with eventual non-axisymmetric minor structures in the form of feature additions or removals.

In analogy to the manual and visual-based process, the target is to derive a kind of least common multiple structure excluding subtractive features as holes. Methods to automatically remove and close holes provided by CAD software can become inefficient or are not achieving the desired result in more complicated models. Consequently, this aspect of geometry abstraction is embedded in the developed way of approximating the structure.

5.1.2.1. Data extraction

The target is to derive a common least multiple cross-section for the axisymmetric main structure or structures on account of the symmetry fundamentals. The guiding idea for the derivation of these sections is to extract silhouette information from the 3D B-Rep model. Silhouettes in this context mean projected curves of surfaces of the model which are of basic axisymmetric type, see algorithm A.4. After having identified potential silhouette faces, the projection into the 2D domain is taking place which is a logical step due to the identified symmetry. Silhouettes which can be uniquely described by two points as for example the projection of cylinders, cones or planar faces are transferred using surface boundaries and projecting these to a cylindrical coordinate system and setting the angular dimension to zero. More complex surfaces are transferred to 2D silhouettes by approximating the projection with an evenly distributed point set and linear spline fitting.

In addition to the silhouette information, another type of data is participating in this approach. A projected set of 2D silhouettes does not imply information about the inside and outside conditions of the body. The face information of the 3D B-Rep surfaces is therefore used. A point cloud is sampled along the retrieved silhouettes and a respective point is calculated using the local face normal vector scaled to a small distance for each point on the curve. All resulting normal points \mathcal{P}_N are projected to the 2D domain afterwards. By the fact that face normal vectors are always pointing outwards in the solid B-Rep case, this transfers a basis information about inner and outer structures to the 2D projections.

Both information, the contours \mathcal{C} and \mathcal{P}_N are stored in an external database for subsequent procedures. In this, the curves are separated into annotated categories for more details about the silhouette types: lines and splines with the splines being stored in a subset \mathcal{S} .

5.1.2.2. Section retrieval

The next part of the process is making use of this information to extract and rebuild potential cross-sections. Algorithms combined with open-source methods as available in the Python library Shapely [268] are utilized to clean the data. Surface gaps and transitions in the geometry model can lead to multiple sub-faces, so in multiple similar silhouette curves. Due to geometric tolerances, the resulting lines can show up slight deviations. Approximately parallel and coincident segments and nearby points are merged using tolerance criteria and nearest-neighbor algorithms as the KD-Tree [269–271] included in the Scipy package [272]. After the silhouettes have been simplified, merged and adapted, information about silhouette end points are packed to a graph-like representation $G = \{\mathcal{V}, \mathcal{E}\}$ with G representing the graph, \mathcal{V} containing the vertices and \mathcal{E} describing the edges, so adapted silhouettes, as shown in 5.3. This structure provides additional information as closed loops, free ends, branching points and connected segments.

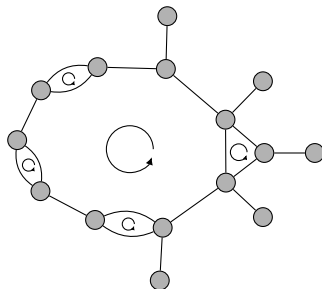


Figure 5.3.: Silhouette graph

The normal information from the database \mathcal{P}_N is also stored in a KD-Tree representation for efficient access later in the process. Based on the graph information, this normal data is pruned by points within proximity to branching points to avoid ambiguous associations in overlapping areas.

Analyzing the graph in more detail can reveal closed loops. These closed loops of silhouettes in turn mean boundaries for polygons, and by this also boundaries for the quasi-axisymmetric section of the root CAD geometry. In simple cases, a single polygon implies sufficient information about every region, so the detection of an axisymmetric cross-section is comparably straightforward and efficient. However, complex components yield silhouettes which either imply ambiguous sections, free ends or no closed loop at all due to missing concrete and axisymmetric information in some regions. For example complex substructures which interfere with the basis cross-section can lead to unclear or non-existing information in this area. In such cases, the objective is to investigate these

areas and evaluate methods to approximate an adequate section. Exemplary non-closed loops and sections are shown in figure 5.4.

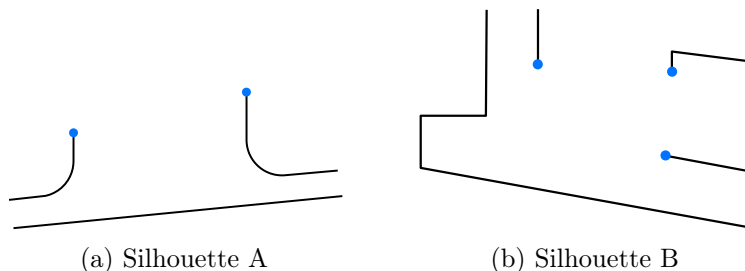


Figure 5.4.: Non-closed silhouettes

The method to close such potential unknown areas which has been developed in the course of this work is aligned to the logics in the manual process. The fundamental objective is to approximate a common least multiple oriented cross-section which is responsible for bearing the major load path of the casing component.

To close these gaps, either new paths have to be introduced or existing ones have to be extended. Both variants require the identification of start points and directions. Without any additional information, simple standalone line objects as in \mathcal{C} depict the only data source the process can work with. This makes it difficult for the process to approximate any geometry. Consequently, more data streams have to be combined to take more information into account.

The graph representation as shown in 5.3 gives access to the leaf points of the structure. To introduce an approximation of load path related information, the Medial Axis (MA) method is used, which is implemented using the duality of Delaunay triangulation and Voronoi graph as described in section 2.5.2. The package Scipy [272] provides implemented versions for those techniques and is used in the following.

After discretizing the silhouettes by sets of points, the Delaunay triangulation is applied and used for the setup of Voronoi vertices including each respective minimum circumscribed circle radius. To filter Voronoi vertices belonging to a potential inner load path, the Delaunay triangles \mathcal{T}_D are evaluated regarding their proximity to the previously mentioned set of normal points \mathcal{P}_N :

$$\mathcal{T}_{D_{internal}} = \{t \in \mathcal{T}_D \mid \nexists p \in t \forall p \in \mathcal{P}_N\} \quad (5.1)$$

The filtered Delaunay triangles $\mathcal{T}_{D_{internal}}$ then yield inner Voronoi vertices, therefore indicate an inner path. The resulting set \mathcal{P}_V represents the basis for the medial axis. This

is considered as additional information to rebuild unclear regions using the medial axis as approximation for the structure load path. Consequently, \mathcal{P}_V serves as an indicator for the direction and basis of a contour closing procedure.

The graph G also depicts the starting point for identifying unknown regions by providing leaf nodes which indicate open contours. The process of finding a suitable point and direction for extending contours is illustrated in figure 5.5 and described in algorithm A.5. Starting from a leaf point (1), the relation to the closest point on the inner load path, so \mathcal{P}_V is retrieved (2). An iterative search for suitable neighbors from the graph G is conducted until a convergence is reached (3,4,5,6). In the given context, splines are often resembling blends or more complicated topologies. In these cases, a generally applicable and reasonable extension and extension vector cannot be guaranteed, which is why the algorithm purposely skips these blend sections. The convergence point and the next neighbor which is not within $\mathcal{P}_{visited}$ is then the basis for the vector for the contour extension.

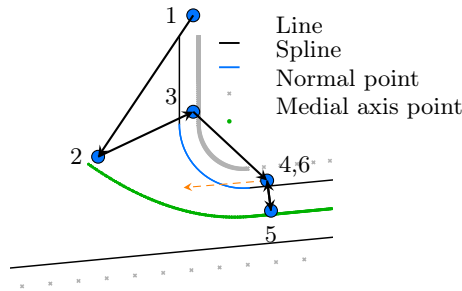


Figure 5.5.: Contour extension process

The algorithm is creating a set of potential contour extensions \mathcal{L}_{ext} which are to process next. By intersection analysis and finding the closest counterparts within \mathcal{L}_{ext} and \mathcal{C} , these extensions are paired and trimmed to the corresponding segments. To increase robustness, a small buffer parameter is applied to these rays, which is built analogously to dilation operations in the Shapely package. If a contour extension neither leads to an intersection nor approaches the proximity of another extension, an iterative process is conducted, which is assuming the current ray origin as starting point and repeating the algorithm A.5 accordingly. The resulting combinations of extension variants are then re-investigated in the process.

From a graph perspective, this approach creates additional nodes and lines in G which support the contour and loop closing by additional geometric information and reasoning. Closing contours in turn means building a final polygon, or final polygons respectively.

5.1.2.3. Polygon analysis

Especially in case of multiple polygons, the set of polygon objects \mathcal{O}_P does not allow a direct conclusion about a potential cross-section yet. An investigation on the contribution of each polygon to the major cross-section has to be conducted first. Major cross-section in this regard refers to the main quasi-axisymmetric section which is bearing the load path. This means that polygons created by silhouettes of for example holes or subtractive features which are interfering with this main section are abstracted and considered as axisymmetric in the first instance. General geometric holes, as flange holes, do not project any silhouette but more complicated geometric shapes in turn can. Other polygons being a result of features which are not participating in the major load path are also neglected in the polygon pairing and combination step. This does not imply that these features are neglected in general. With reference to the fundamental strategy, the goal is to reproduce a 2D section for the major axisymmetric part and analyze non-axisymmetric features in the 3D CAD domain afterwards.

The next step includes the identification of the described major polygons. The medial axis is also considered as a metric or identification logic in this context to similarly estimate potential load paths. From all polygons in \mathcal{O}_P , neighboring pairs are identified by dilating the polygons with a negligible value and assessing their mutual intersection. For each polygon pair, their associated cleaned Medial Axis System (MAS) is investigated next. More information about the developed medial axis methods is given later in a more comprehensive form in the dedicated section 5.1.4.1.

The first aspect copes with the principle of smooth load path continuation. This is captured by so-called medial axis flaps as shown in figure 5.6a and the properties of the cleaned MAS of each polygon. The flaps in this scope are identified by an inscribed circle radius trend $\lim_{x \rightarrow length} r(x) = 0$ and a proximity of their respective branching points. This algorithm first filters potential smooth transitions between polygons, thus potential consistent load paths, leading to a subset of polygon combinations $\mathcal{O}_c \subset \mathcal{O}_P$, see figure 5.6a. An iterative algorithm then is combining pairs, rebuilding their MAS, retrieving new polygon combinations thereof and repeating this process until the major path is assembled.

The second logic is targeting the aspect of main path participation. A separation of two neighboring polygons means that there is a non-axisymmetric object affecting and splitting the region. To deal with this aspect, the MAS of each structure is assessed towards a decision of either actively being part of the major path or representing a minor additional structure. If the MAS of the minor structure is rather representing an

additional branch than a part of the existing major path, see P_2 as example in figure 5.6b, the detailed investigation is left for the later 3D analysis and the polygon is neglected in this procedure.

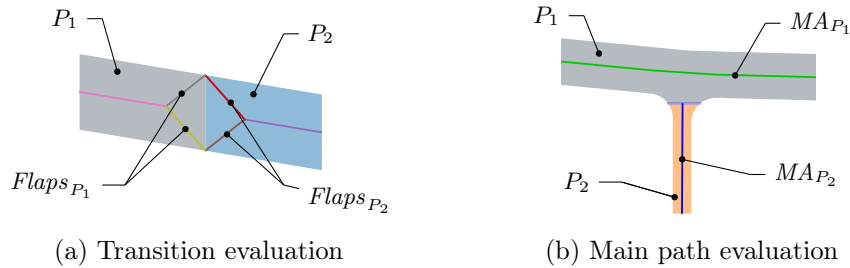


Figure 5.6.: Polygon investigation

After all polygons are evaluated and combined, it is important to clean its contours in order to find adequate CAD sketch entities for the subsequent rebuilding process. To achieve this, the polygon boundary curves and vertices are mapped to the original contours \mathcal{C} , so pseudo 3D face representations, including their line information as straight lines and splines. This is achieved using the basis silhouette graph G and KD-Tree methods to map the polygon points to the graph nodes and finding adequate paths from \mathcal{C} to connect these nodes. Evaluation of the resulting path reveals the line and spline information, see figure 5.7, which is stored accordingly in the process.

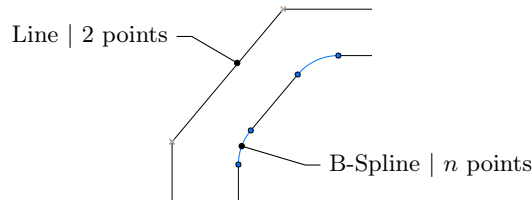


Figure 5.7.: Line / Spline mapping

The resulting section is then saved in a XML database in the form of line coordinates, adjacency and line type information. Based on the cross-section, a first volume estimation is done and compared to the 3D solid volume for assessing the degree of axisymmetry on the one hand, and the suitability of the cross-section on the other hand.

5.1.2.4. Component analysis

The stored cross-section information is now accessed from within the 3D CAD software, where the process is automatically constructing a sketch entity containing the lines. If

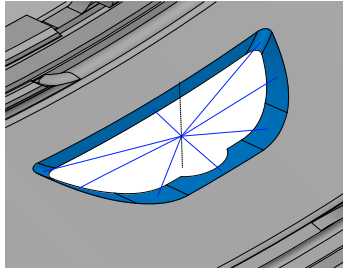
a contour is annotated with a spline type, a B-Spline method is fitting a curve based on the database points, otherwise a simple line object is created. The resulting sketch, or sketches respectively, is referred in a revolve command which is set according to the calculated angular dimensions of the original component body B .

Based on the purely axisymmetric revolve body B_R , the investigations on non-axisymmetric features are conducted. Boolean operations as intersection and subtraction reveal differences in the bodies and topologies. By subtracting the bodies $\Delta B = B - B_R$, ΔB provides data about additive features which are incorporated by solid bodies and topologies. The intersection $B_I = B \cap B_R$ leads to changes in the topology of B_R as a result of subtractive non-axisymmetric features in B . The developed process analyzes the changes in topology, which are mostly attributed to faces, and groups the modifications by adjacency. The arrangement of the faces in each group allows conclusion about possible semantics of the feature.

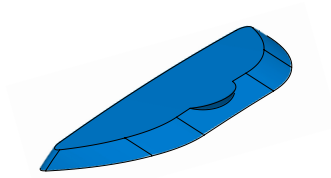
Understanding patterns on the one hand helps to understand the structure, and on the other hand can significantly increase process performance. For this reason, those face clusters are analyzed regarding area, position, number of participating faces and dimensions to identify patterns. Different types of bounding box algorithms are utilized to find the most suitable for the given case, which is then the basis for further pattern detection steps. Parameters associated with patterns are for example span and count or pitch and count. The advantage of identifying those is that only one representing member R has to be evaluated in detail, and the operations can be efficiently instantiated using CAD pattern features with the retrieved parameters.

Consequently, a representative R_{G_i} of each group G_i is analyzed to retrieve its volume properties in relation to B_R . Especially retrieving complex face arrangements can lead to CAD errors due to noncontinuous boundary edges which can occur in boolean operations. Boundary faces belonging to R_{G_i} are analyzed by building a virtual center, casting rays and evaluating face normal vectors and boundaries. On account of the derived information, the feature type can be distinguished in more detail as related to for example holes or general material removals. The result is then used to build a closed volume using the original topology B_R , see figure 5.8. Based on the pattern data in G_i , the enclosed volume is instantiated accordingly, stored in a feature group and its access is enabled via an internal interfaces. These objects can be used to derive associated properties, pattern data, features, bodies and all other types of related topological entities.

An exemplary use case of all derived information is the semi-parametric automated rebuild of components as described further in [273]. Table 5.1 shows three application cases of the presented methodology including a performance evaluation.




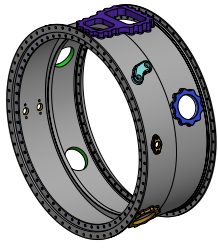
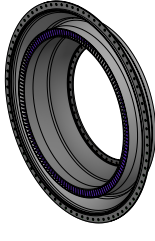
(a) Hole analysis



(b) Negative volume

Figure 5.8.: Non-axisymmetric analysis

Table 5.1.: Example models and process performance

	Model 1	Model 2	Model 3
			
Section	3 s	1.5 s	3 s
Rebuild	51 s	42 s	57 s

5.1.3. Substructure analysis

After having segmented the components using either feature or silhouette-based approaches, the next step is to elaborate them in detail. The manual process includes steps and reasoning which assess the effect of the substructure on the component or system behavior. If the component is assumed to be important for the type of simulation and for result quality requirements, ways to represent the substructure in a simplified form in the simulation model are elaborated. Based on physical understanding and modeling and simulation experience, different representations and simplifications are manually introduced with regard to the objective of achieving a sufficient compromise of accuracy and detail. Clearly, the analysis of substructures is defining the next logical step for the automation. To align with this process, the involved engineering and cognitive thoughts have been analyzed and translated to corresponding computer language and programming ideas.

Similar to the recognition process, the context and relations of the substructure are important for the procedure guidance. The interfaces between the separated substructures can be similarly visualized by using a graph representation method, see figure 5.9.

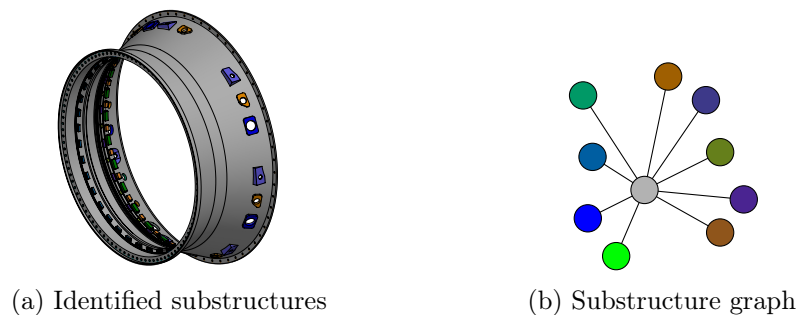


Figure 5.9.: Model analysis

As it can be seen in the graph and the component, the number of concrete substructures is already reduced by methods which recognize patterns. Basic properties like volume and various bounding box dimensions, projected body center points and topological attributes support this identification of patterns within the collection of segmented substructures. This does not only reduce complexity, and thereby improves process performance, but also supports model understanding. The consequence is that it is sufficient to analyze one representative of each pattern for the evaluation of the structures. The focus in this work is set on circular patterns along the component center axis since this type of pattern is the most common and most crucial one in the given scope.

Volume properties are logical criteria to assess the influence of a structure on component

behavior. Small substructures are often assumed to be negligible without having to sacrifice simulation quality and are also the target of a plurality of simplification techniques as described in section 2.5.1. However, some parts exert a significant influence on behavior despite having a comparably low volume share. Regarding this, more criteria and logics have to be developed to analyze the substructures in a more sophisticated way. Dimensional properties are next to be embedded in the process algorithms. A best-fit bounding box algorithm is applied to the representative in order to provide a reasonable construct for structure dimensional extents. The resulting properties provide data about general dimensions but also about the ratios of the dimensions, what in turn allows conclusions for potential and suitable simplification approaches. Identified patterns and therewith the derived structure count and span parameters are also participating in the developed analysis methods since they provide additional semantic information.

The role of a substructure within a component, and so the system, plays an important role in the evaluation process. The substructure graph described earlier is providing access for this criterion. First, the major parts, thus the root bodies derived from segmentation algorithms, are representing the main load path bodies. By identifying these nodes in the graph, other functional load path relevant nodes can be derived. An example of this can be seen in model 3 in table 5.1 in which vane structures represent a cardinal part of the component. From a higher level perspective, the graph model also reveals the interfaces of a substructure within a system or assembly which can be linked to entities within the recognition framework described in the previous chapter. These interfaces play as well an important role for model decision-making.

These interfaces are therefore taking part in the analysis logic. The topology of the interface between substructures holds information about mutual influence aspects and potential load paths. The ratio between the interface area and the structure volume is condensed in a criterion which is denoted as averaged height in this context for this purpose. The idea here is to assess an approximated structure height in case of an even volume distribution. Another interface criterion is targeting the topology of the interface and elaborates the existence of inner topologies as a measure for complexity.

Other strategies pursue a similar idea and evaluate the mass distribution with reference to the interface, defined as the term interface distance in this work. The Center of Gravity (CoG) is considered as reasonable property for this. Its projected distance to the related interfaces reveals this interface distance and shall help to elaborate the distance of the mass center to the base structure. Due to the present axisymmetric layout, the CoG aspect only provides meaningful value for structures spanning small sectors, see figure 5.10. If a substructure is spanning the complete circumference, the CoG is positioned

in the cylindrical center, therefore not adding appropriate value to the criterion. In this scenario, the cylindrical bounding box is used to evaluate the distance of its box center to the interfaces. The juxtaposition of the averaged height and interface distance then yields additional information about model complexity.

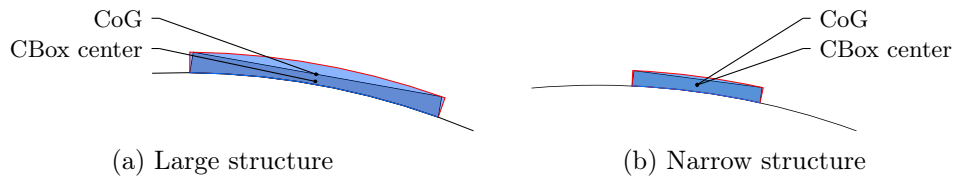


Figure 5.10.: Interface distance evaluation

The next integrated logics to be presented deal with aspects which are already pointing towards distinct simplification paths. The face-pairing approach (see section 2.5.2) is integrated in several commercial CAD software but, as already mentioned, can imply robustness and applicability issues in complex scenarios. However, this specific circumstance is exploited to evaluate on the one hand an approximate wall-thickness distribution, and on the other hand to estimate the structure complexity. Research has shown that a previous segmentation eases the application of face-pairing approaches, hence the presented approach is providing suitable conditions. Applying the method creates a set of sheet bodies related to identified face pairs. The API provides interfaces to access each instantiated face thereof and retrieve the assigned averaged thickness.

The next task is to evaluate the generated sheet bodies and assess if they represent an adequate midsurface structure. This is done by geometrically analyzing the proximity and consistency of the original structure interfaces and resulting sheet boundaries. It is necessary to assure that the created midsurface does imply a geometry which is able to rebuild the load path of the actual structure. Moreover, sheet boundaries are analyzed towards mutual intersections and offsets to investigate the structure consistency. If both analyses meet the imposed criteria, the maximum wall-thickness is used as a measure for the midsurface applicability. Another representative value is introduced by multiplying the average thickness of each participating face with its area to put the volume of the surrogate shell construct in relation to the original volume.

The last part of the substructure of the substructure analysis process deals with an aspect similar to this idea of assessing midsurfacing capabilities. Based on the bounding box fit and the derived dimensions, a principal axis in straight box configurations or a curve in cylindrical scenarios is approximated and defined as guide. First, the existing faces which are approximately normal to the guide are studied regarding their surface

area. From this set, their representative characteristics are elaborated by building a volume using their respective area and guide length, which is then compared to the original volume. If none of the faces are reaching satisfactory matching levels, the process is utilizing a virtual scanning procedure to investigate the structure in more detail and to potentially find more suitable representative faces. This method builds artificial cross-sections throughout the structure as depicted in figure 5.11 and repeats the volume matching process. The result of the process then serves the purpose of an indicator if the structure can be described to a satisfactory extent by a single section which allows drawing conclusions about beam simplification potential.

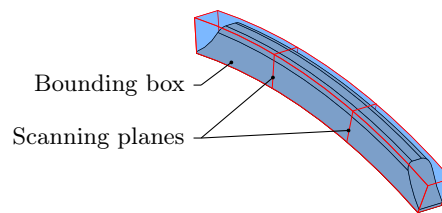


Figure 5.11.: Beam evaluation

All the described criteria are collected in an internal feature vector, which is described in a schematic way in table 5.2. The aspects embedded in this vector shall mimic key aspects for engineering decisions and conclusions. Entries which are of boolean type as midsurface or beam potential are assigned with integer values 0 and 1 while others are of the standard float type in \mathbb{R} . Furthermore, the derived properties also build a basis for related recognition goals. Vanes, for example, commonly connect two casing structures while being thin-walled and appear in a patterned arrangement.

5.1.4. Simplification

Performing the manual tasks to introduce various simplifications or ways of representing substructures is done after the visual analysis of substructures. On account of this analysis, decisions are made how structures could be approximated and represented so that the surrogate design increases model efficiency or reduces complexity without affecting quality to an unsatisfactory extent. As described in chapter 2, the aim is to pursue two goals: either to reduce the number of elements required or to reduce the element complexity. Thin-walled structures are often approximated by 2D shell structures, while some parts also allow a dimensional reduction to 1D beam elements. In lights of these considerations, the cardinal goal is to find ways of simplifying structures which despite this maintain adequate stiffness and mass properties to match simplified and

Table 5.2.: Substructure feature vector

Entry	Type
Volume	<i>double</i>
Volume share	<i>double</i>
Count	<i>int</i>
Graph position	<i>int</i>
Assembly interfaces	<i>int</i>
Averaged height	<i>double</i>
Interface distance	<i>double</i>
Interface topology	<i>int</i>
Dimensions	<i>double</i>
Dimension ratios	<i>double</i>
Midsurface	<i>int</i>
Wall-thickness	<i>double</i>
Shell volume deviation	<i>double</i>
Beam structure	<i>int</i>
Beam volume deviation	<i>double</i>

original model characteristics for given simulation purposes.

In complex scenarios nonetheless, simplification is not guaranteed to be applicable without entailing unavoidable deviations in the simulation results. If a simplification is set as definitive target, the manual process commonly includes experimenting with different modeling approaches and parameter tuning to achieve a satisfactory validation result. This in turn represents an obstacle which is hard to deal with from the automation perspective.

With regard to the transfer processes, the major types of approaches to represent structures which are included in this framework are depicted in figure 5.12. The figures 5.12a and 5.12d already give a hint on different substructure types which are denoted with for example boss and fixing type in the course of this work. The decision for a substructure description is made based upon the previously described feature vector and guides the representation transfer process. The following sections go into detail about the method development and approaches to convert and introduce these representations.

5.1.4.1. Midsurface

The major key point for the simplification of the present thin-walled structures is the use case of 2D shell elements, so the retrieval of a midsurface structure. Due to the component categorization, the suitable components are already pre-selected and represent

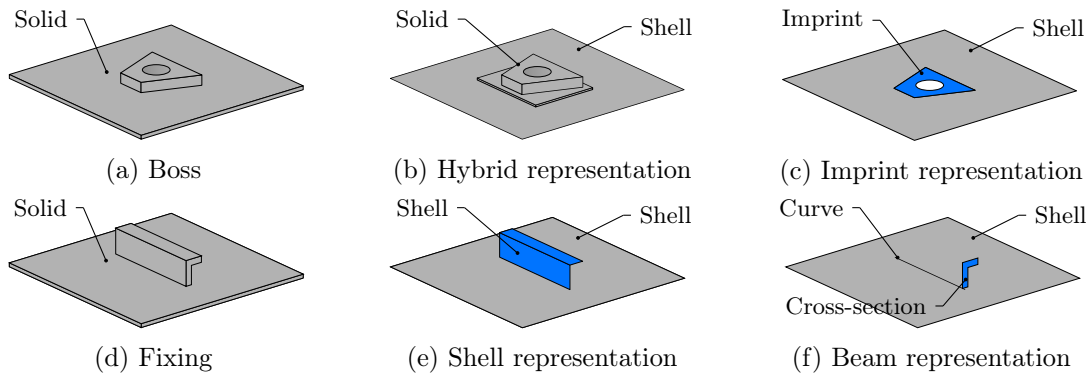


Figure 5.12.: Representation approaches

quasi-axisymmetric casing structures which share a high potential to have thin-walled regions.

As discussed earlier, face-pairing and medial axis depict two common approaches for the midsurface approximation. Face-pairing represents a straightforward approach situated close to the B-Rep domain, already creating suitable CAD geometries. However, evaluating these in the 3D domain can be ineffective and less expedient in automated processes due to complicated or missing programming and analysis interfaces. On the other hand, medial axis or object methods are often limited in performance, especially in a 3D environment, or are likely to produce complex structures in detailed regions which interfere with meshing algorithms.

The quasi-axisymmetry basis implies an aspect which this process is drawing advantage of. By exploiting the symmetry knowledge, the process is able to focus on a 2D data basis which is transformed to a 3D model afterwards, similar to the cross-section methodology in section 5.1.2. Also similar is the idea to focus on the main structures in advance in a 2D environment and put emphasis on the remaining and additional structures afterwards.

The medial axis approach has proven its applicability in the 2D case, and furthermore comes with a lot of capabilities for efficient model control, modifications and adaptations. To overcome its limitations and suitability issues, a workflow for building geometry-based and optimized structures has been developed and is presented in the course of this thesis. This workflow is illustrated in figure 5.13 and described in the following paragraphs.

The input to this process is the approximated axisymmetric cross-section from the CAD part which is stored in a database for the outsourced process pipeline. In addition to this, the original component and system interfaces related to the quasi-axisymmetric component play a cardinal role also for the simplified representation, hence are appended to the database. Especially for load path evaluations and distinguishing between mandatory

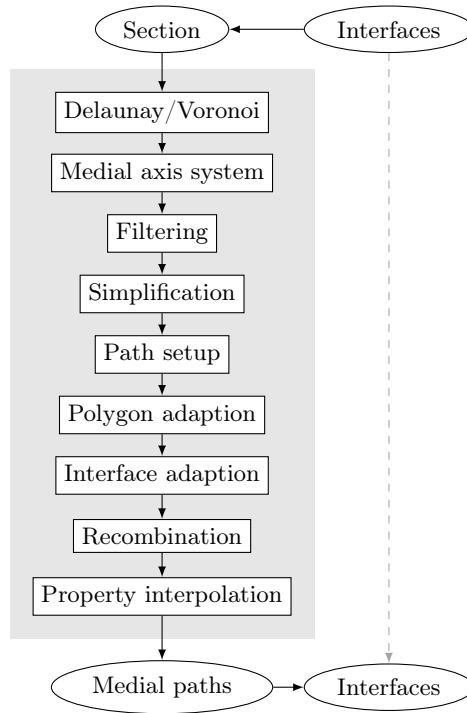


Figure 5.13.: Medial axis workflow

and minor structures, interfaces are often used as reference also in the manual process. The basic way of storing this data is involving general data objects as boundary line or curves created by projecting faces to the 2D domain as for example flange interface faces.

The first algorithm of the process handles the creation of the polygon P_B from the boundary curves \mathcal{C}_B as well as a discretization of the boundary curves to a set of points. The set of points is input to a Delaunay triangulation integrated in the SciPy package [272] with its accuracy depending on the discretization resolution. The Voronoi graph is sharing duality with the Delaunay triangulation, and thus can be directly derived. To filter the significant Voronoi vertices \mathcal{P}_V , the created polygon P_B is used in a vectorized operation to assess the point containment. Since each vertex is associated with a specific Delaunay triangle and represents the center of the associated circumscribed circle, a radius or diameter value can be assigned to each point $p \leftarrow (x, y, r)$ with $p \in \mathcal{P}_V$.

Based on the filtered Voronoi vertices, the Delaunay or Voronoi graph neighborhood properties reveal remaining Voronoi ridges which describe the connections of neighboring vertices. This circumstance can be automatically processed, which results in a set of vertex pairs. An algorithm aligned to the Depth-First Search (DFS) technique [274] has been developed to build a coherent system of branches and branching points from these

pairs which is denoted with MAS in the following.

Due to the fundamentals of the medial axis, the resulting MAS can be a far cry from a simplified structure. So-called medial flaps are common byproducts which are not expedient for the simplification intention. So, the first MAS pruning step is targeting these flaps by elaborating the branch thickness trend: $\lim_{x \rightarrow length} r(x) = 0$. Another type of branches which can be neglected are minor branches which do not add additional information to the system. The identification is conducted on basis of the branching points. If the branch itself is contained by the circumscribed circle of the associated branching point, it is not considered as meaningful, hence the branch is filtered from the MAS, see the exemplary illustration in figure 5.14. The third type of interfering branches are underdetermined branches. These are occurring as a result of external triangles which despite this have their related Voronoi vertex within the polygon boundaries. Compared to the internal Voronoi ridges, these show a significantly higher ridge length, so a larger distance between two vertices with less local information, and are thereby sorted out.

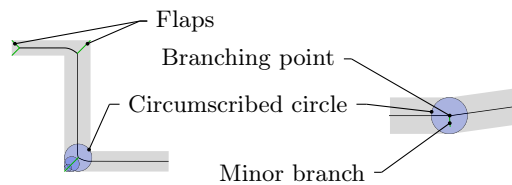


Figure 5.14.: Branch analysis

The points which describe the remaining branches are stored within a KD-Tree structure T_0 as a mean of thickness distribution function within the polygon for later access. In order to decrease the data embedded in the MAS, methods for simplifying the branches are executed next. The Ramer-Douglas-Peucker (RDP) algorithm [275, 276] is implemented for this purpose and reduces the number of points required to describe a curve while maintaining the fundamental shape. The remaining points are mapped to the points stored in T_0 to reassign related thickness values.

The next sub-process deals with merging selected branches to reasonable paths. Especially with further adaptations and modifications being planned, the focus on standalone paths allows better and more structured access to the entities. Two aspects are guiding this process.

On the one hand, the cross-section is representing the main structure, so the path which describes the majority of the polygon is considered as the main path. The MAS is transferred to a graph structure with its branching points setting up the nodes and branches describing the edges. Each edge is assigned with the inverse of the branch length

in order to make use of existing shortest path algorithms, which in turn reveal the branch combination recommended for building a main path.

On the other hand, interfaces in the 3D model domain have to be existing in an accessible way also in the simplified model. To cope with this, the projected interfaces from the database are mapped to the existing branches. This mapping process uniformly discretizes the entities to point sets which are then the basis for assessments of distance, distance relations and geometric curve properties. As a result, each interface is assigned to a segment or segments of a branch, which therefore are crucial branches for the final structure. The remaining parts of the MAS are considered as minor paths.

A new internal architecture is built around the identified paths, which also contains information about the mutual connections and connection locations for the anew branch combination after modifying the paths. The paths, however, still do not depict a sufficiently simplified representation due to minor and complicated segments. A crucial objective, also in the manual process, is to approximate and represent the given geometry in a clean and simple way. For this purpose, the algorithm A.6 has been developed, which simplifies the MAS while taking the given geometric boundaries into account. In this method, each point of the path is associated with the closest geometric boundaries. The association is then reordered and filtered based upon the most frequently associated curves in order to retrieve the most cardinal geometric entities. From this set, segments of points within the paths are internally combined according to their related boundaries. The critical area of each segment, in this case the point representing the section bottleneck, is identified by being the vertex with the smallest circumscribed circle, or the smallest distance to the boundary respectively. This point is assigned with the boundary curve vector direction to build a line on to which the points are projected afterwards, resulting in a new adapted segment. The projection distance of each point is also representing a potential offset parameter for later FE mesh property setup purposes. Segments containing only one associated entry are not adapted in this context and considered as mandatory to reproduce the path in less dominated regions. After all points have been processed, thus potential new line segments have been built, the recombination is taking place. Two adjacent segments are combined by extension and intersection if the extension is fulfilling the polygon boundary conditions, otherwise they are simply connected. In case of a standalone point being the neighbor to a segment, the point is projected to the extended line which triggers iterative analysis steps to check polygon and model consistency. An exemplary application on P_{major} and the overall result is shown in figure 5.15. The degree of geometry abstraction which is desired for the purpose of simplification is clearly present. Potential thickness offsets due to geometry deviating from the skeleton line can be modeled by elaborating

the present offsets with reference to the root KD-Tree T_0 and account for these in the form of shell offset properties in the 3D FEM model.

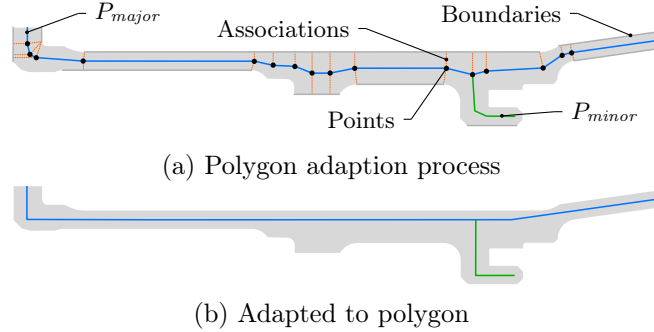


Figure 5.15.: Polygon adaption

Besides the polygon adaption, the interfaces have to be considered as well. In order to preserve interface conditions, the projected interfaces are internally re-mapped to the new paths and segments thereof. This provides additional access and so room for automated adaptations and data aggregation. An example for this is given at the flange connection of two structures. Commonly, it is desired to avoid geometric gaps in the model, which is why the flange related shells are offset to be coincident. The developed process includes an option to be automatically offset the flange related curves to the interfaces.

Afterwards, the separate paths have to be recombined in an adequate and clean way to ensure structural consistency. On account of the distinction between major and minor paths, the developed algorithms ensure that the major path topologies are not affected by the recombination, which would prove counterproductive to previous adaption procedures. The objective of simplification is still dominant here, so an optimization process aims for avoiding minor path fragments which would be affecting meshing results and introduces minor path modifications during this recombination process. Finally, the paths are extended to the section boundaries.

The last step deals with a significant missing property for the midsurface model. The curves which will be representing shell structures in the 3D domain are requiring shell thickness attributes for the involved elements. Consequently, the presented workflow interpolates the thickness of the paths and assigns concrete thickness values to the segments thereof. For this purpose, the paths are discretized to a string of points. For each of those points, the nearest neighbor is retrieved from the KD-Tree T_0 which has been set up earlier in the process and which contains an approximation of the thickness resolution over the polygon. The discretized point set \mathcal{P}_d is of dimension $\mathbb{R}^{(x,y,r)}$ with the

radii derived from the thickness distribution in T_0 . This point set is converted to a set of points of dimension $\mathbb{R}^{(z,r)}$ by substituting (x, y) with $z \in [0, l]$ with l being the relative length of the path. This allows a subsequent two-dimensional analysis as shown in figure 5.16.

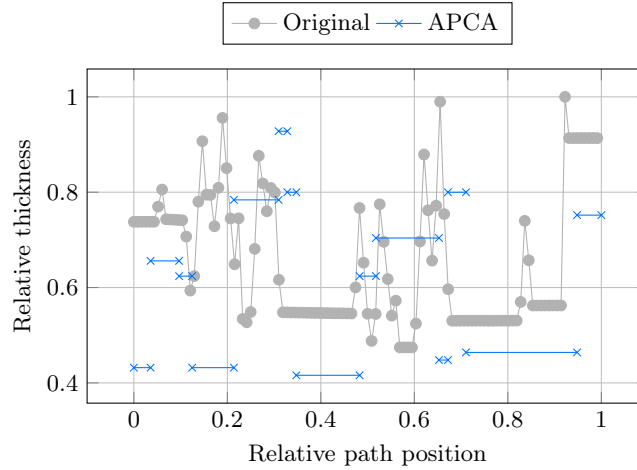


Figure 5.16.: Thickness interpolation

The reduced data is then smoothed and refurbished to a suitable representation of the thickness without introducing additional complexity. For this intention, an algorithm based on the Adaptive Piecewise Constant Approximation (APCA) technique [277–279] has been developed and implemented. Parameters for this algorithm are a minimum segment length and an error tolerance for the fitting optimization. The segment length serves the purpose of avoiding small sections which would result in small B-Rep faces implying complexity for the meshing process. The algorithm then is able to split path segments for a better approximation of the local thickness. The mapped interfaces, however, are not further subdivided since this would result in separated CAD entities which complicate the application of boundary conditions. Figure 5.16 also shows the resulting thickness levels with the segments being either defined based on topological aspects or necessary thickness adaptations.

At the end of the process, the final network of the paths is stored in a database file for data exchange reasons. The root interfaces are mapped to the new line segments representing the new related interfaces, which are stored in the database as well. This tracks the change of geometries and topologies within the automated modification process, and thus also deals with the aspect of short-term memory mentioned in 3.2. The process still contains knowledge about the objects even after modifying and abstracting the

conditions.

The resulting database is loaded in the CAD environment to transfer the result to the 3D domain again. The section is created automatically as a CAD sketch which contains the line segments and serves as basis for revolve features with the parameter identified from the original component. Each resulting face is then mapped to the respective line segment in order to transfer thickness attributes and to remap the newly created faces to the stored interfaces. The identified subtractive substructures, see section 5.1.3, can also be transferred to the created midsurface model. For additional simplification opportunities, these subtractive features can be pre-selected by a filtering algorithm to neglect for example minor holes in the model based on the identified substructure properties.

5.1.4.2. Dimensional reduction

After the main structures have been transferred to a midsurface representation, the related nodes in the substructure graph are replaced accordingly, which in turn updates the node, so geometry object, relations. The developed process afterwards turns over to these adjacent substructures and sets its focus on their conversion strategy.

Imprint representation

The first presented substructure pipeline is pursuing the goal to thicken the associated underlying shell model as a way to account for the additional local structure stiffness and mass. The interface between the main structure and the substructure therewith plays an important role. The separation of the structures reveals the interface which is used to project the separated part to the associated midsurface geometry. The projected curves are used to divide the underlying face or faces which enables direct access to the geometry later, resulting in a new set of faces $\mathcal{F}_{Imprint}$ which are visualized in example 5.12c. A bounding box and distance evaluation is conducted to compare the previous topology to the topology after the face modification has been introduced to retrieve these newly created imprint faces and associate them to the imprint representation structure, see figure 5.17. By exploiting the embedded substructure knowledge about the volume, the area of the created faces $\mathcal{F}_{Imprint}$ is used to approximate the imprint shell thickness. The resulting thickness is added to the respective thickness values of $\mathcal{F}_{Imprint}$ in their internal properties. Moreover, the substructure knowledge allows to automatically retrieve a suitable and consistent offset. For this purpose, the face normals of the imprint faces are compared to the vector of closest distance between substructure and shell body. This

reveals the offset direction, so its proper mathematical sign.

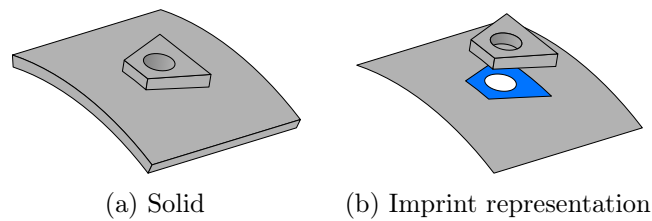


Figure 5.17.: Imprint process

In some cases, the direct imprint of the interface topology can lead to a complicated set of curves as shown in the example in figure 5.18. This can exert a notable influence on meshing complexity and as a consequence be opposed to the main simplification idea. The developed process therefore includes a second path which builds a box around the projected region boundary curves which is projected afterwards as a surrogate to the midsurface structure as shown in figure 5.18b. This box layout significantly benefits the meshing algorithm by its clear and straight boundaries. The following process is similar to the direct imprint process. The examination of the interface topology reveals attributes, which are serving as basis for the process internal logical decision between direct and approximated imprint strategies.

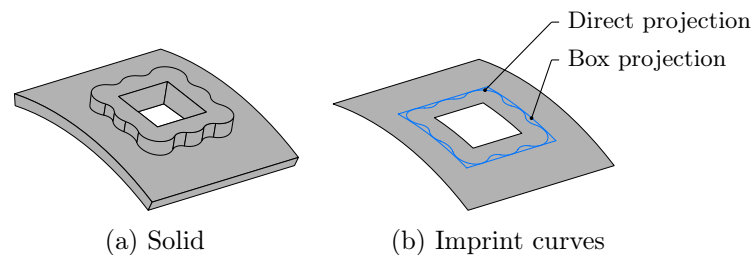


Figure 5.18.: Approximated imprint process

Similar obstacles for meshing are sliver or small faces which can occur as a result of projecting and dividing faces. To cope with this issue, an algorithm analyzes the created imprint faces considering boundaries and extension and filters the faces accordingly. The filtered faces are then merged with the original B-Rep entities to avoid unnecessary edges and faces. All gathered information about the created imprint face representations is stored internally for easy access, mapping of interfaces and additional later use.

Hybrid representation

If a higher degree of detail is requested or the structure itself indicates a complexity which makes simplification unreasonable, another transfer approach is considered. The hybrid representation, as shown in example in figure 5.19, is able to integrate 3D structures within the 2D shell model. This 3D structure allows remaining close to the geometric root design in the specified regions, which are able to replicate local stress and load conditions in a very accurate way.

To create such a hybrid structure, an automated procedure is introduced which has its first tasks with the imprint path in common. The interfaces of the substructure describe the geometry of the load path or substructure interaction. For simplicity reasons, the approximating imprint approach is used, which encloses the projected interface curves using a bounding box method. To mitigate interferences of the cutting procedure with localized stress aggregations near the considered substructure, an offset is introduced. The offset value is based on the local dominant shell thickness, as recommended by Robinson et al. in [183], and rounded to a tangible value. Each corner point of the resulting box is then projected to the underlying shell structure. The local face properties are used to access the face normal vectors at the projected point locations to build a line in both positive and negative direction. The lines are then constructing a wireframe, which is used to generate the related cutting faces, see figure 5.19c. Using the face normals ensures a perpendicular interface between shell and solid, which has proven to be most suitable as also discussed in [9].

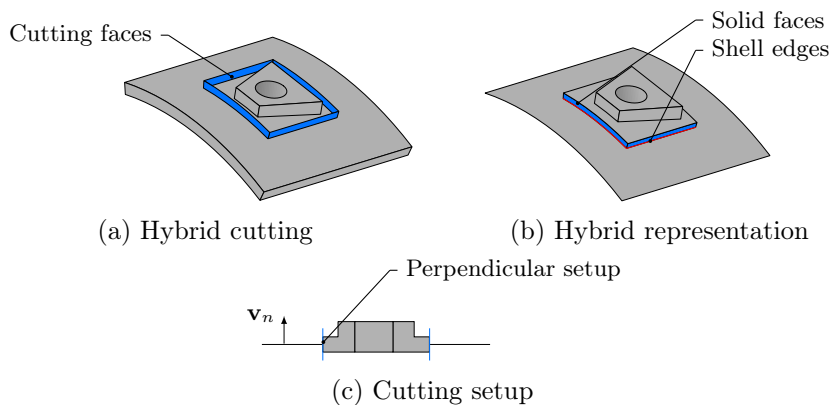


Figure 5.19.: Hybrid process

The connection of the created boundary curves results in a set of cutting faces which are used to split both related midsurface and solid geometry accordingly. The gathered

substructure information is then exploited to identify the resulting shell and solid bodies and also the created shell-solid interfaces as shown in figure 5.19b. This information is mandatory for later model setup and therefore part of the internal hybrid representation object which possesses interfaces for the downstream process.

Shell representation

The shell representation workflow pursues the reduction of substructures to shell models, but in contrast to the medial axis approach described in section 5.1.4.1, the key point is the midsurface structure being created by face-pairing methods as mentioned in section 5.1.3. The application of the methods is comparably straightforward and aligns with the midsurface evaluation which is part of the substructure analysis procedure.

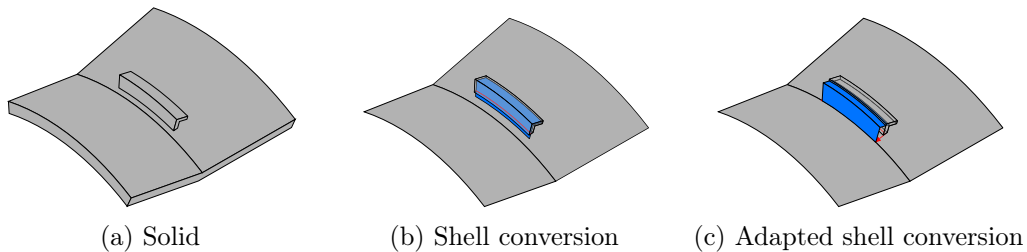


Figure 5.20.: Shell representation

With the associated main structure being replaced by the midsurface entity, a geometric gap is occurring. For this reason, the midsurface structure has to be extended to the associated shell bodies. The substructure interfaces reveal potential interface edges which are used to extend the related B-Rep objects as depicted in 5.20b. The average thickness result for each generated face is applied to the object properties and stored internally together with boundary and interface information.

The shell structure can introduce additional complexity in the form of sliver or minor geometric entities as for example smaller negligible face regions. The model from figure 5.20 is taken as exemplary case in this regard. The distance between fixing structure and the edge on the main midsurface body is identified as relatively small which means undesirable meshing boundary conditions, thus increases the probability for either a high amount of small FE elements or distorted elements. Since the process is aiming for an abstraction in combination with simplification, a potential step in the manual process would be to consider both shell edge and main structure edge to be coincident to avoid these minor meshing discrepancies. An edge mapping algorithm is introduced for this intention which searches for edges on the associated structure similar to the interface

edge regarding type, orientation and distance.

Based on shortest distances between a sampled set of points on both objects, the distance values are evaluated and the vectors of shortest distance are investigated regarding parallelism. In case of a matching pair and a distance smaller than a specified parameter, the substructure representation is automatically moved to a more robust position as shown in figure 5.20c. If the distance is not exceeding a certain distance, the gained benefit in model and meshing consistency is considered to be outweighing the minor change in volume and effect on model behavior as a consequence of this model modification. From a process related perspective, identified substructure patterns allow to focus on a single representative midsurface model, which can be patterned accordingly in the process to save resources and increase performance.

Beam representation

Decreasing the dimensionality by another magnitude leads to a beam or 1D representation. The 1D model depicts the most basic approximation of mass and stiffness in this context. The application, so 1D meshing, requires an input of basic beam properties and 1D objects like curves or splines as a meshing target. In order to automate the beam conversion, the required geometric entities have to be created in the model preparation phase first.

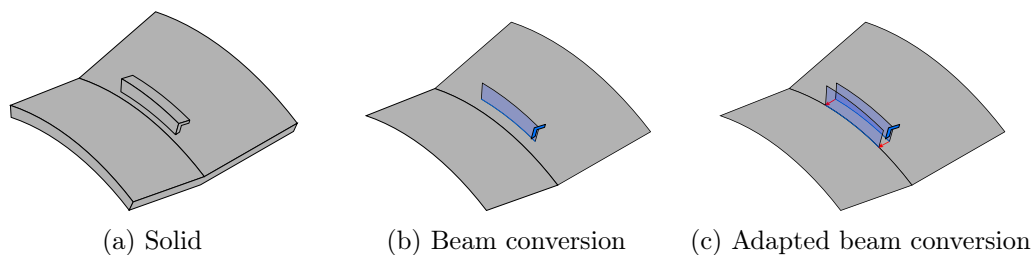


Figure 5.21.: Beam representation

To guarantee a consistent target for the beam application and consistency of the beam mesh and its underlying shell mesh, an additional edge, or edges respectively, are introduced which serve as an anchor for the substructure to be simplified. The interface edge retrieved from the shell process described in the previous section depicts an example for such edges. If no midsurface structure is available for this edge retrieval, an abstract virtual sheet is created based on the interface and the bounding box orientation as shown in figure 5.21b. This sheet then involves the artificial interface edge which is, however, not yet part of the main midsurface body. The transfer to the non-manifold environment in the CAE domain allows a so-called stitching process provided by Siemens NX which

merges the topologies of the participating objects. This ensures the geometric consistency between the target edges for the beam application and the adjacent shell bodies, hence also a consistent basis for the meshing process.

The other important aspect to be derived is referring to suitable beam properties. The process for this has already been mentioned in section 5.1.3 with regard to the evaluation of a potential cross-section. Siemens NX provides an interface to calculate adequate FE beam section properties from a face object input. The cross-section, however, is assigned to the location of application, meaning that the assigned standard cross-section is not in accordance with the real cross-section position. To cope with this, an adequate offset parameter can be introduced which artificially moves the assigned cross-section to the real cross-section position. A selected point on the interface edge is used to evaluate the vector and distance to the real CoG of the cross-section in a coordinate system defined by component center axis and a radial vector. After retrieving a suitable cross-section face and offset, all this information will be guided to the later FEM part by the means of an internal beam modeling object.

The problem of sliver or minor face regions described in previous section can also occur in this case. A similar mitigation algorithm is taking place in this approach as depicted in figure 5.21c. Accordingly, the offset has to be adapted to a new representative point on the new moved interface.

Substructure combination

More complex scenarios and arrangements often do not allow direct access to substructures, thus complicate approaches as hybrid processing due to a high probability of affecting other geometries within proximity. Similar to the manual process guided by visual input, an evaluation of potential substructure interferences is a prerequisite for a satisfactory transfer procedure. In the course of this research, an algorithm has been developed which utilizes the known bounding box techniques as an abstract approach to reveal substructure proximity and interferences. Based on projected bounding box distances and specified minimum distance parameter in the process settings, the bounding boxes can be grouped in an effective way as shown schematically in figure 5.22. If the resulting box almost covers the entire circumference or at least major parts of it, the process decides to use a full axisymmetric box instead to avoid additional topologies. The combined bounding box then can be used to initiate the hybrid process for multiple substructures at once, creating a hybrid patch enclosing these. This avoids minor or small geometries between structures which would on the one hand introduce complexity, and on the other hand

distort simulation conditions in these areas.

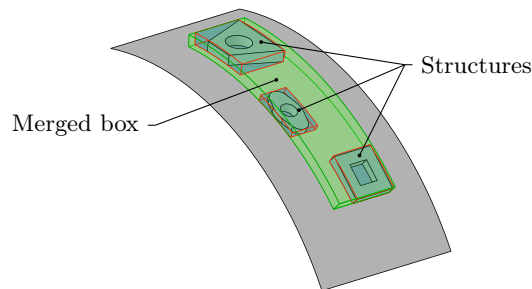


Figure 5.22.: Proximity analysis

Analogously to the analysis of substructure proximity, a check is done which deals with distances to the boundaries of the associated main structure or other geometric circumstances which could entail disadvantageous implications for process success or robustness. The assessment then filters potential representations for the enclosed structures.

5.1.4.3. Substructure representation

With previous sections having presented different approaches to introduce simplification or model modification, this section deals with methods to make the decision for a representation type. The feature vector derived from the substructure analysis in section 5.1.3 depicts the key aspect for this. A logical sequence of decisions is made based on the principles embedded in the vector and common best practices.

The volume proportion of a structure or structure group is an obvious parameter to assess the influence of a structure and also a key argument for the decision to neglect substructures. Some structures in turn are critical for the component, and thus have to be specifically considered. The entry graph position contains data related to a load path association aspect. Analogously, interfaces required for the assembly have to be maintained or mapped to new and potentially idealized entities and imply reasons for a special substructure treatment.

The decision for the beam representation for example is coupled to the detected beam potential and the resulting beam volume deviation. The beam potential is linked to dimensional properties and relations and also strongly reliant on a successful retrieval of a suitable cross-section. Similarly, the shell approach is dependent on the output of the midsurface evaluation. Moreover, the inner topologies attribute already gives hints on the applicability of beam or shell idealizations. Inner holes or boundaries can interfere with both beam curve or shell interface entities and lead to an inadequate replication of

stiffness and mass properties.

More flat structures, often referred to as bosses in the course of this work, do not guarantee a reasonable composite shell representation because the assigned shell thickness in contrast to the comparably small shell extents would not meet the basic theoretical shell assumptions. The stiffness distribution of a beam approach is relatively focused on the beam entity, and thus in many cases not reasonably able to reproduce the structure stiffness properties spanning a flat and large area. As a consequence, the imprint strategy is more suited for these sufficiently flat scenarios. The averaged height and the interface distance are assumed to reveal more information about the topology, thus about the suitability of the imprint approach.

In more complex scenarios which do not fit the described criteria, the hybrid approach is chosen if applicable (see section 5.1.4.2). If for example the center of the structure is significantly far distanced from its connection interfaces, increasing imprint shell thickness would introduce the related mass but would be contrary to the fundamental shell assumptions, and thus potentially not be able to reproduce the local component conditions. By maintaining the root 3D structure and integrating it in the surrounding shell model, the local characteristics are expected to be better replicated to a more satisfactory degree.

The described criteria are embedded in a decision tree which decides on suitable and possible representations for the modified CAD model based on experience and best practice values. If multiple representations are possible and reasonable, the process decides for the one which adds the least complexity to the model. In this context, this relates to the expected effect on element count and type. The imprint is the most desired strategy because it theoretically does not introduce additional elements and exclusively influences the associated element properties. 1D elements are highly efficient in computation and do not significantly increase the element count. Additional shell structures in turn introduce new entities to be meshed, thus also new elements, while hybrid representations include additional 3D elements and boundary conditions requirements for the connection of shell and solid bodies.

5.1.5. Simulation model setup

From a process agenda perspective, the representation transfer processes are automatically conducted after the decisions for the representations paths having decided for representation types. At the current state, however, the model is still within the CAD domain, so the FEM simulation model has to be built upon this basis next. Each representation comes

with an internal programming class which holds all required information about the related objects and properties. These classes contain an output interface which is accessed by the automated workflow in order to build a complete database containing all information about the model required for the CAE environment. The new interfaces derived from the interface mapping process are replaced its original entities in the associated files accordingly to ensure model and knowledge consistency.

In the CAE domain, the polygon model transfer and stitching of unconnected shell elements are executed automatically. The stitching feature in Siemens NX combines multiple shells which are separated in the manifold CAD modeling environment can be connected in the non-manifold CAE model. This involves that new topologies are either introduced on one or more participating shell bodies or existing boundaries are modified in order to merge the geometries. Consequently, previously identified entities which are required as interfaces or target objects have to be remapped to the newly created topologies. The process automatically copes with this aspect by bounding boxes analyses and proximity and topology checks.

The subsequent process building blocks address the automated meshing process. The meshing algorithms in detail are beyond the objectives of this thesis, so the developed strategy utilizes given state-of-the-art meshing procedures, including their provided mesh simplification options. For practical reasons, all settings related to the general algorithm parameters are outsourced to an external file for easy access within the automated process. To support a consistent, efficient and suitable mesh, adjacent similar geometry types are grouped for a simultaneous meshing. Additionally, algorithms for suitable mesh size estimation are introduced to provide a satisfactory starting point for building the mesh. The core of these algorithms and mesh size estimations are based on precedent mesh convergence studies. The objective in these studies was to discover mesh sizes which lead to satisfactory quality results on the one hand, and on the other hand contribute to the desired reduction of model complexity. The type of geometry which is stored and described in the database defines and directs the mesh type selection and property setup within this framework.

Properties as shell thickness and shell mesh offsets are attributed automatically due to the embedded knowledge about the geometric basis to the respective meshes. The beam cross-sections, offsets and orientations are set up analogously with reference to the component coordinate system described by the component center axis and radial vector and the stored parameters.

Furthermore, hybrid regions have to be connected to the surrounding shell structures to ensure consistency. The gluing approach described earlier in section 2.4 is considered

with its manifold application possibilities and convincing result quality. The gluing algorithm integrated in Siemens NX requires geometric entities as inputs. Also in this regard, the knowledge integrated in the database can prove its advantages. The hybrid container described in 5.1.4.2 contains both edge and surface entities required as connection interfaces for this algorithm, and thus support the automated application.

The result of the CAE section of the presented approach is a fully meshed FEM model which is highly connected to the basic geometric conditions and properties. In order to enable sophisticated geometry or substructure related post-processing, the developed framework includes methods which draw FEM element, node and property data from the automatically set up simulation objects and relate these to the associated substructure container in the database. Consequently, the database summarizes information about the geometry, the mesh process as well as final FEM entries, and by this provides the opportunity to create a logical connection between each geometric substructure and the related FE results as illustrated in figure 5.23.

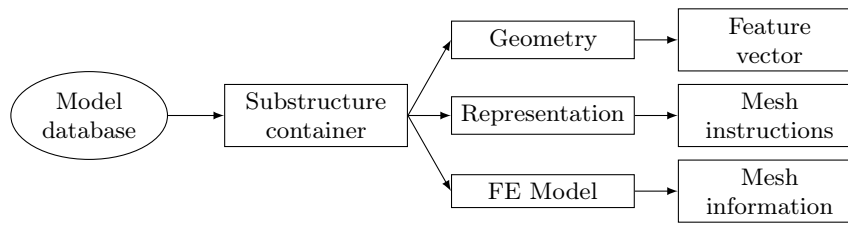


Figure 5.23.: Database information scheme

5.1.6. Application

In the following, some application cases are shown for the developed processes, algorithms and methods. This section focuses on their application on standalone components but a later section deals with the combination of the recognition framework in an assembly environment. The related processes of analyzing and building a model for simplification purposes are presented with the resulting CAE models being shown for each case. A modal study as described in section 2.4.3 is conducted to conclude a basic assessment of the quality of the idealized model. A performance and complexity validation afterwards points out the advantages of the described process.

Figure 5.24a shows the first application case C1 which represents a casing near the combustion chamber. This component has already been used as application case for the segmentation process in section 5.1.2. For the evaluation, three different simulation models have been generated: C1-A, C1-B, C1-C, see figure 5.24. These variants have

been built using the automated methods and workflows with specific variations in the selected representation paths.

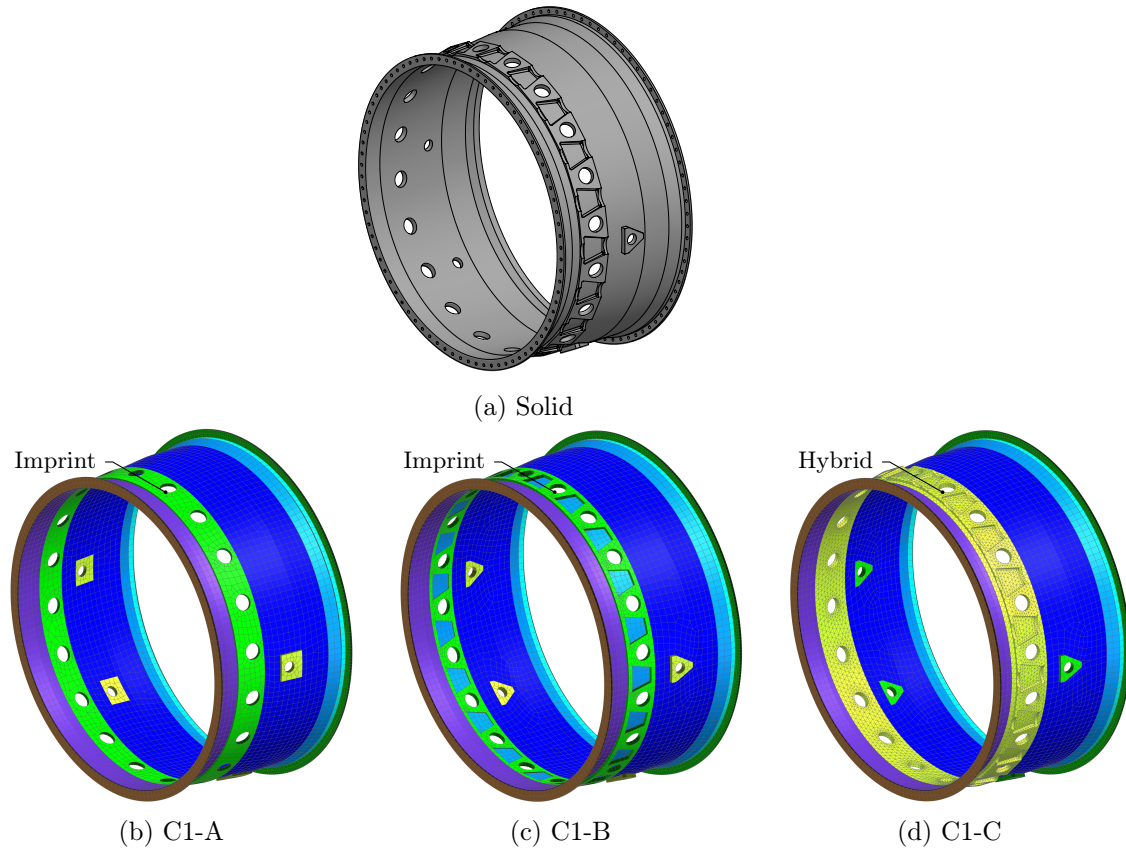


Figure 5.24.: C1 models

All three simulation models 5.24b-5.24d contain the major structure in the form of a shell midsurface representation. The differences are within the modeling approaches for the adjacent substructures which are denoted as burner ring and general bosses in this context, see figure 5.24a. In C1-A, a path involving a high degree of abstraction and simplification is chosen which applies a box imprint (see section 5.1.4.2) for both types of substructures. Increasing the level of detail by introducing exact imprints leads to the model C1-B. Some geometric circumstances are, however, cardinal for component behavior, contain decisive interfaces and cannot be simplified without having to expect drawbacks in model quality. In this regard, C1-C showcases the hybridization process for the burner ring structure.

The automated process transfers the geometry model to simplified versions, introduces suitable meshes with geometry-based FE properties and builds the simulation model for

the modal analyses. For modal comparison purposes, a reference model is additionally generated, which is a full-featured 3D model and a 3D tetrahedral mesh with sufficiently detailed resolution while ensuring at least two elements through thickness. This reference serves as basis for the MAC for mode correlation and for performance and model investigations. The correlated mode pairs of both analysis and reference model allow a more sophisticated comparison of eigenfrequencies. This study is done for the presented simulation models C1-A, C1-B, C1-C and the results are shown in figure 5.25.

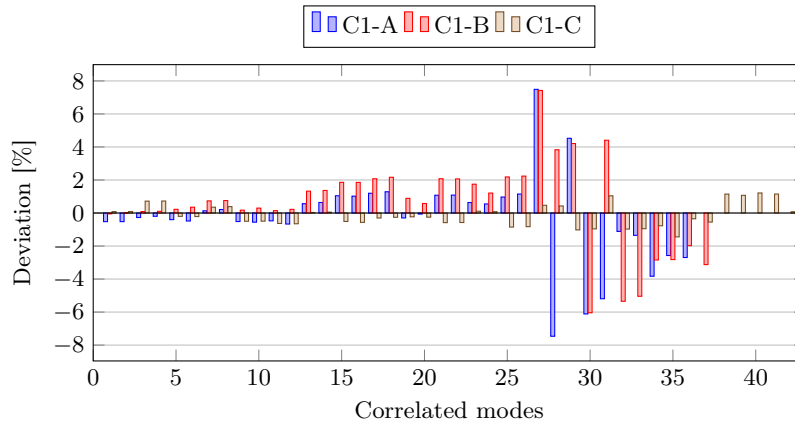


Figure 5.25.: C1 results

The value on the horizontal axis identifies an identification number of the correlated mode pairs while the vertical axis describes the relative deviation of the eigenfrequencies to the reference model. As it can be extracted from this figure, the model variants lead to a different number of correlated modes. Consequently, the models show dissimilar capabilities to replicate the exact original component behavior. The amplitude of deviations also allows conclusion about the abilities to match the local stiffness and mass properties of correlated modes. All three variants show a comparably high number of correlated modes with C1-A having the fewest mode pairs and C1-C the most. Compared to C1-C, C1-A and C1-B especially reveal higher deviations in higher frequency mode regions which means that the simplified models are less capable to reproduce more complex modal characteristics. The C1-B variant technically provides more geometry detail due to its direct imprint method. However, it is showing slightly higher deviations, which could be related to a more focused and potentially improperly allocated stiffness properties since the thickness is applied uniformly. The hybrid approach considered in C1-C is able to cope with these complexities and achieves a more desirable matching of the model behavior. This emphasizes the advantage of 3D regions for more complicated regions,

especially in scenarios where higher modes play an important role.

A similar outcome can be derived by inspecting and comparing the MAC matrices for C1-A and C1-C, see figure 5.26. Figure 5.26a shows a blurred region at a higher mode levels, while the correlation in figure 5.26b shows a more matching trend with a higher number of correlated modes. Due to the transition between the elements and regions of different dimensionality in the hybrid structure, the mapping algorithm is associating different and potentially duplicate node sets which can result in a theoretically higher mathematical similarity of modes, especially for symmetric ones as it can be derived from figure 5.26b.

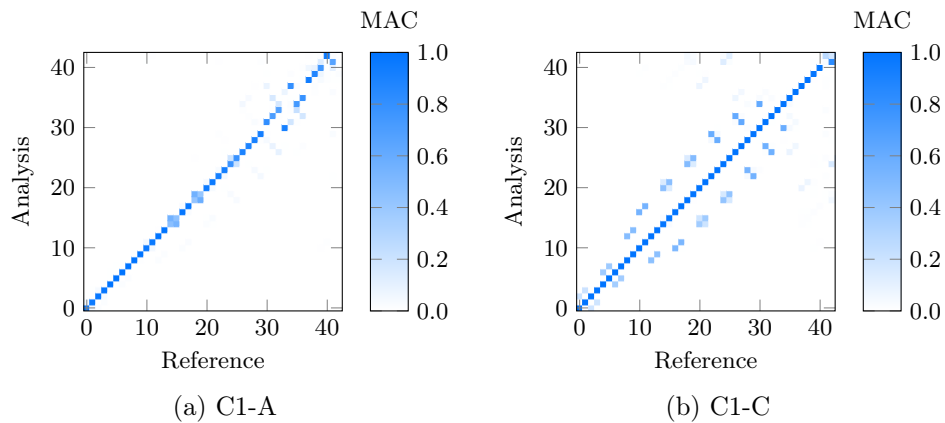


Figure 5.26.: C1 MAC evaluation

For the performance and result evaluation, a set of metrics is introduced to assess the influence of the simplification steps and the quality of the geometric approximation of the model. The calculation time required for the simulation, the number of DOF of the model as well as the result size have been selected to evaluate the simplification potential. The volume metric is describing the relative deviation in volume between reference and analysis model. The metric values resulting for the presented model variants are depicted in the following figure 5.27. In this, the values are put into relation to the reference model metrics to generate relative percentages for the complexity related outputs and a relative deviation for the volume metric.

Due to the degree of simplification, meshing algorithms encounter less problematic situations in the box imprint model C1-A, thus leading to a faster simulation and a smaller number of DOF. On the contrary, the 3D meshed hybrid section in C1-C is demonstrably influencing the complexity in the form of an increase in DOF or result size of the generated model. The required simulation time does not vary as much as for

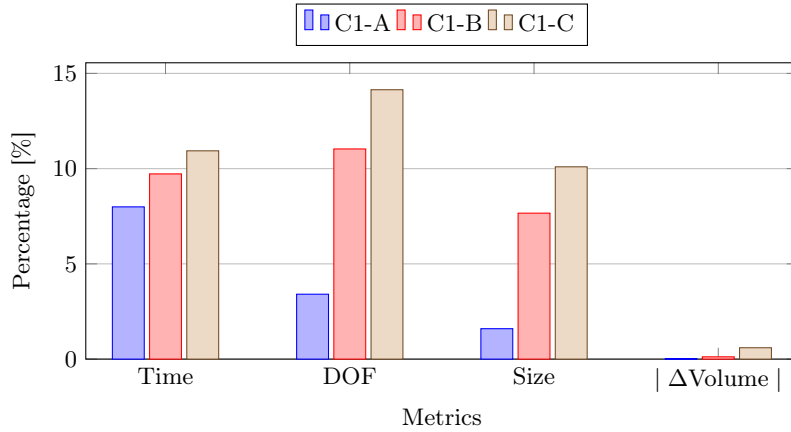


Figure 5.27.: Evaluation of metrics of model C1

example the DOF count but is expected to be significantly noticeable in larger models or assemblies due to the estimated influence of fourth power of the number of DOF on eigenvalue problem-solving time as described in [37]. When moving to the next metric, all three simplification variants show a slight increase in model volume. Investigating the shell thickness distribution does not unveil noticeable differences from a geometric perspective. A potential contribution to this deviation can be identified in simplified, in this case neglected, flange holes. Another cause for the deviations can be volume intersections at midsurface segments connected at an angle or different shells connected in general, implying volume intersections and so additional volume. Nevertheless, the advantage of reducing for example the DOF count by $\sim 85\%$ to 96% is outweighing the negligible volume increase for the objectives in this work.

The next application case C2 is representing another casing structure with comparably smaller substructures, see figure 5.28a. The main variant C2-A is depicting the automated transfer result of this model to a simplified representation. In addition to this, studies which neglect specific substructures as pointed out in figures 5.28b-5.28d are compared to evaluate the substructure influence by means of the presented automated framework. C2-B and C2-C compare for example the influence of a circumferential and segmented fixing structures on the model quality.

The figure 5.29 shows the correlated eigenfrequency deviation results. The model C2-A shows a satisfactory approximation of the model behavior especially in the lower range of modes. The fixing features neglected in C2-B appear to not significantly affect the characteristics of the simplified model, quite the opposite to the fixing feature which is suppressed in C2-C. This feature therefore seems to play a major role for the component

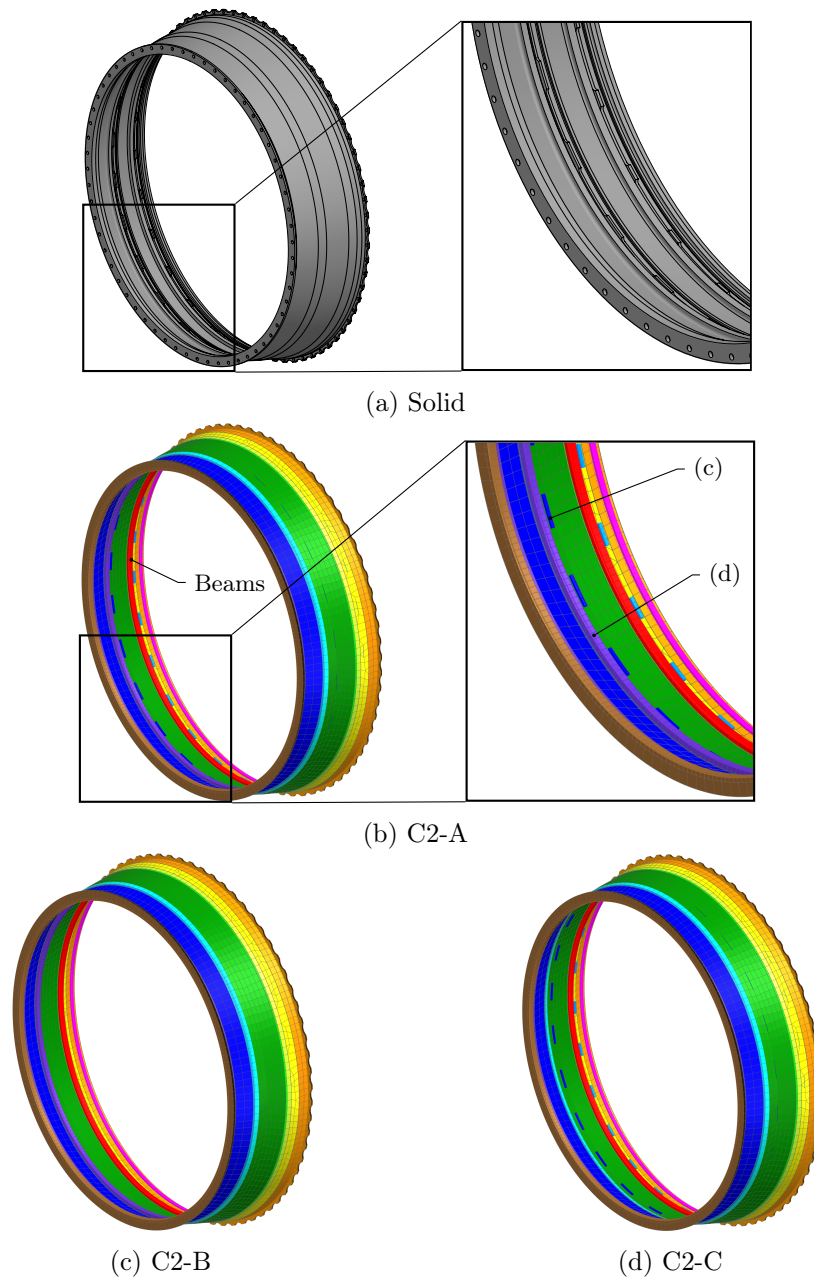


Figure 5.28.: C2 models

properties. The achieved simplification is assessed in a similar way using the described metrics with its result being shown in figure 5.30. Since the neglected substructures are converted to 1D beam elements, their suppression is not influencing the performance metrics to a considerable extent which is why only C2-A is shown in this figure.

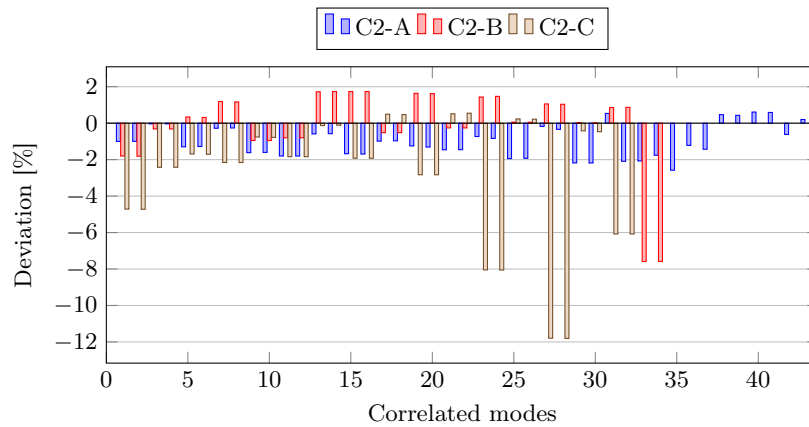


Figure 5.29.: C2 results

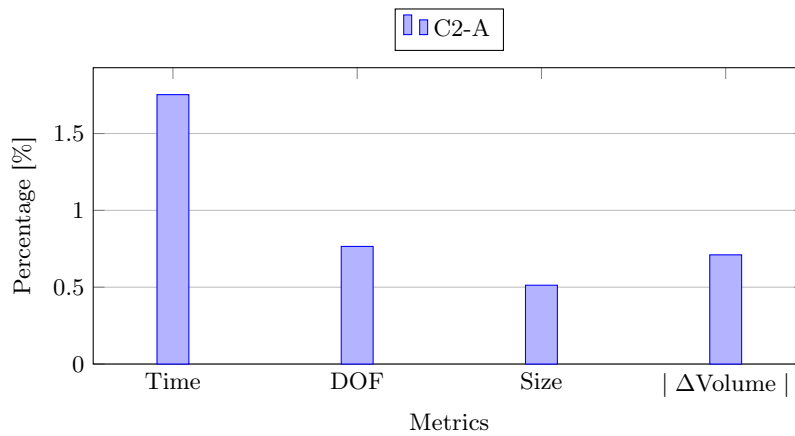


Figure 5.30.: Evaluation of metrics of model C2

The third application case C3 is a component from the casing category which consists of a multitude of substructures of various type. Four different modeling variants have been chosen and created using the developed framework methods, which are shown in figure 5.31.

The intention behind these variants is to set emphasis on the hybridization capabilities and specifically to showcase the substructure merging option described in section 5.1.4.2. Model C3-A has been transferred with a deactivated hybridization setting, which means that a 2D shell or 1D beam representation is forced by the framework. The structures in the rear part are consequently converted to imprint shell approximations, see figure 5.31b. The other models C3-B, C3-C and C3-D are set up for hybridization in turn with different merging parameters. The variant C3-B contains small hybrid regions in the area of the

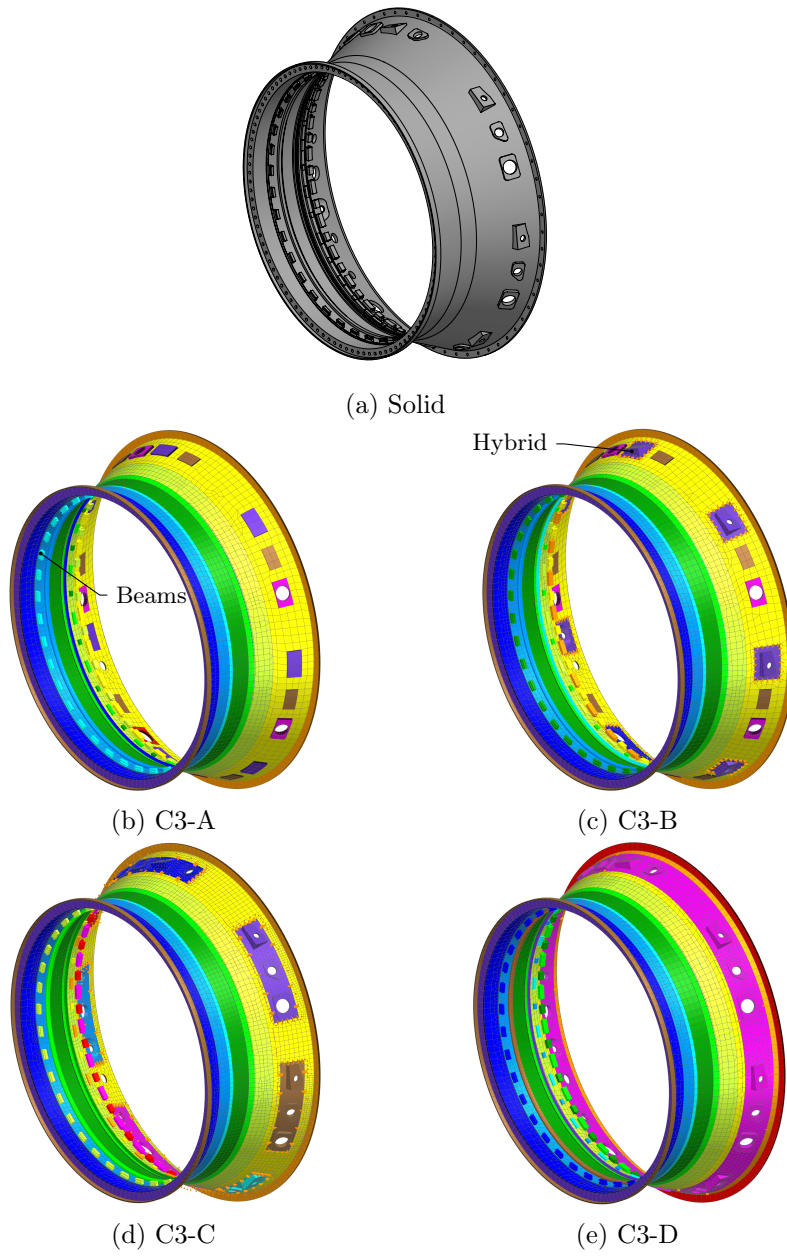


Figure 5.31.: C3 models

most complex boss resembling structures. To avoid sliver and minor areas, the algorithm has connected multiple substructures within their proximity to merged hybrid segments in C3-C. In some cases, the merging can lead to a remaining region which is not implying a significant potential for simplification anymore due to its comparably small extents. In these cases, the algorithm can decide to close regions to avoid minor regions on the one hand, and on the other hand to reduce the amount of necessary interface boundary conditions between shell and solid bodies. This intention is pursued in model variant C3-D.

The results of the modal analyses and subsequent correlation with the 3D reference model are shown in figure 5.32. As it can be derived from this graph, all four variants achieve reasonable levels of conformity with the reference model behavior. In some modes, the pure imprint version C3-A appears to have more noticeable outliers than the hybrid versions.

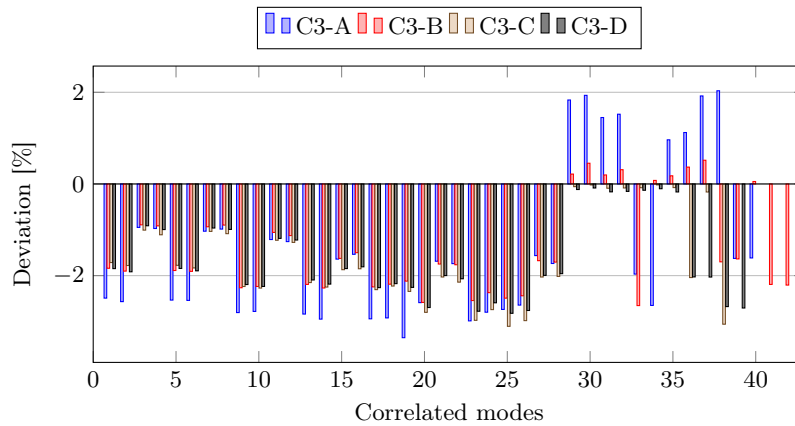


Figure 5.32.: C3 results

With regard to the metrics in 5.33, all four variants also achieve a high level of simplification and model complexity reduction with C3-A achieving a relative DOF number decrease of -98.45% . Despite C3-D containing more 3D elements than C3-C as a consequence of its extended hybrid region, its DOF count and required computational time is lower. This can be traced back to the higher geometric complexity in the neighboring regions, implying a higher number of elements. Moreover, multiple hybrid regions mean multiple necessary connecting conditions to the surrounding shells which can drive up simulation complexity, and so result in higher required computational time.

The last application case in the course of this thesis is C4 which is representing a component consisting of two major axisymmetric structures which are connected via

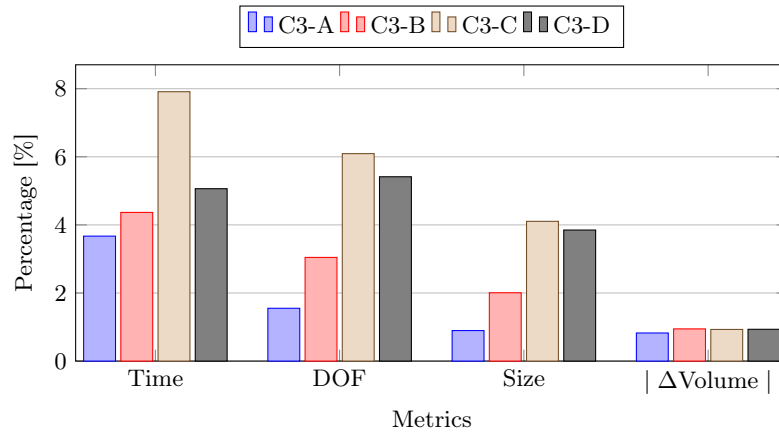


Figure 5.33.: Evaluation of metrics of model C3

structural vanes, see figure 5.34. The algorithm is able to identify these major structures as well as the included vanes as already mentioned in the section 5.1.3. Due to the recognition of the vanes, expedient methods can be applied in a strategic way. Additional algorithms are integrated in the framework to specifically handle vane geometries and structures. In the common path, the face-pairing method is used to convert the 3D vane geometry to a 2D shell representation. The more complex the vane geometry, however, the higher the possibility for robustness issues, which is why two additional methods have been developed which share a more approximating and abstracting intention. Apart from robustness considerations, especially smaller and simpler vanes recommend a reasonable simplification to avoid meshing and topological difficulties. Both methods transform the 3D vane geometry to a point cloud, which is used as basis for fitting a surface. Depending on the vane simplification settings, either a planar surface as done in C4-A, figure 5.34b, or a surface of higher order is fitted as in C4-B. Since the vane geometries in this model are comparably simple, the higher order fit surface is very similar to the standard face-pairing midsurface used for vane representation in variant C4-C, figure 5.34c.

The results of the modal investigations depicted in figure 5.35 show higher deviations than previous application cases. The average of the absolute deviations of all three variants is approximately 2%, which, nevertheless, can be within an acceptable tolerance for specific design and simulation purposes. The planar fitted vane variant C4-A reveals higher deviations in especially lower frequency modes compared to the other variants. However, it is evident that this component is more complex and a clear correlation between FE properties and potential simulation result errors is not easily derivable. The load path through these structures and especially in the connection area to adjacent structures can

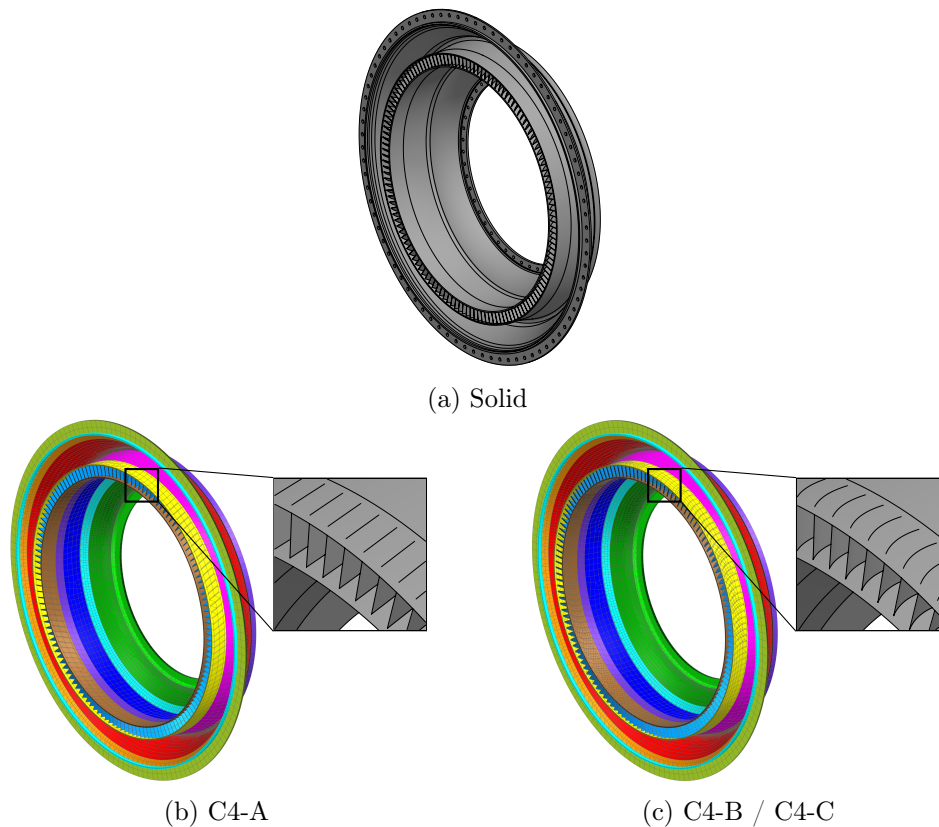


Figure 5.34.: C4 models

be difficult to rebuild using exclusively shell strategies. Furthermore, the load distribution in the area of shell connections also in the main structures is complicated to cope with in the exclusive shell domain and can introduce additional source of errors.

With the planar fitting creating a comparably simple topology, the related meshing is producing a reasonable mesh which implies a lower element count. Moreover, the interface edge topology also influences the mesh on the adjacent shell surfaces. The planar face involves linear interface edges, which also support the generation of clean quadrilateral FE elements in the adjacent regions. This influence is confirmed by the metric evaluations in figure 5.36. Since the face-pairing midsurface of the vanes of model C4 is relatively similar to the surface created by the higher-order fitting, models C4-B and C4-C are producing similar results and metrics.

In conclusion, the developed automated process pipelines and algorithms have shown and proven their advantages for the component transfer. On the one hand, they are able to reduce the model complexity in a significant way, while maintaining a reasonable

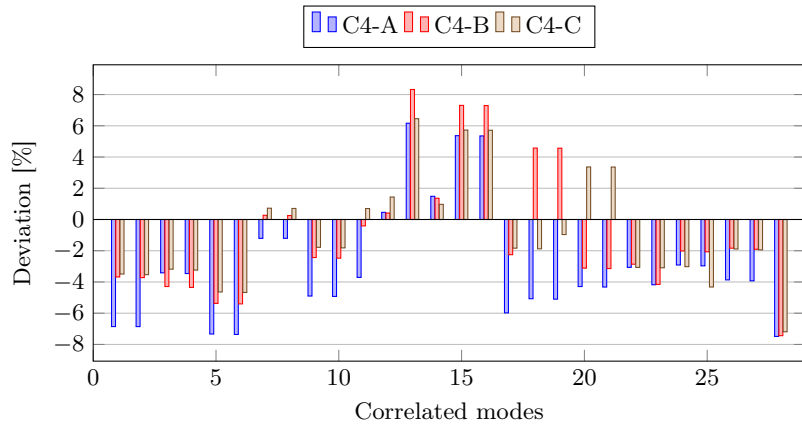


Figure 5.35.: C4 results

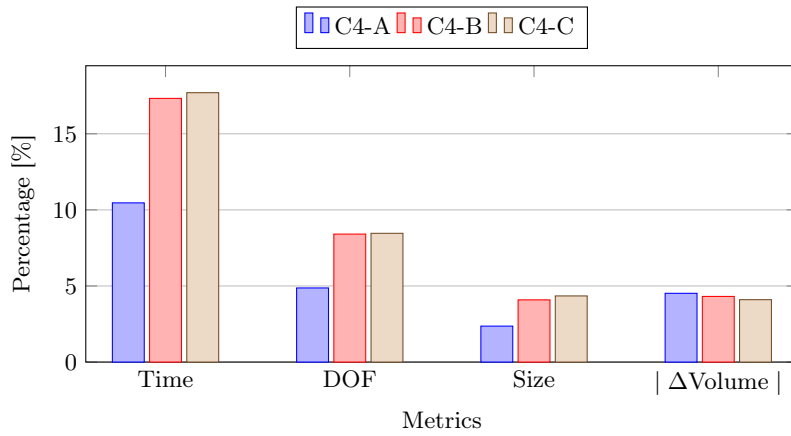


Figure 5.36.: Evaluation of metrics of model C4

model and simulation quality on the other hand. The developed strategy introduces a smart approach to mimic engineering logics in the transfer path from fully featured to simplified and abstracted simulation models for various design studies. The capability to create models of different level of detail and representations holds the potential for automatically generating a model set for different purposes from a single geometry.

For more sophisticated studies, the clean structure and setup of FEM properties in the generated simulation models allow additional interfaces for model optimization as model updating [280–283]. Approaches from this field can further trim and tune accessible FE properties to better match the characteristics of the analysis model to the reference model.

Finally, the performance of the developed algorithms and processes is put into focus on

the basis of the shown application cases C1-C4. The process itself is divided into three major steps: the segmentation of the component, the analysis of the derived substructures and the final conversion to selected simulation representations. The last step, which includes the setup of the meshing data and instructions, so the preliminary FEM model setup, is a database-related task which takes less than 0.1 s in all cases and is not depicted in the figure 5.37 for clarity reasons. The subsequent meshing process within the FEM environment is not directly part of this research, and therefore not part of the shown evaluation.

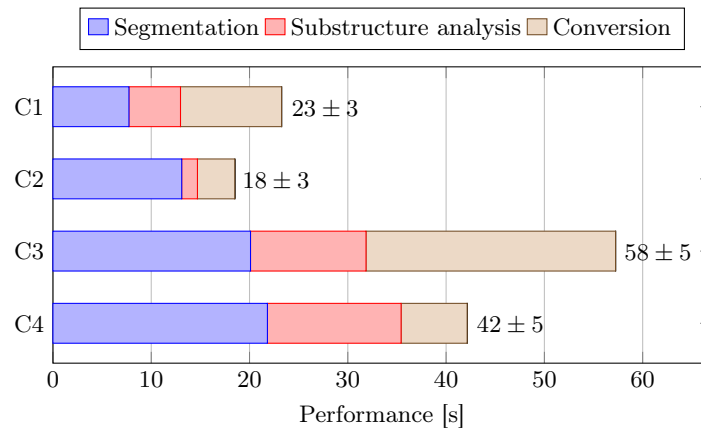


Figure 5.37.: Process performance evaluation

5.2. Beam structures

5.2.1. Process

After having presented the developed research and methods for the casing category, the process pipeline for the beam structure category is described in this section next. The basic ideas which guided the recognition of beam structures allows a more straightforward approach compared to the casing category. Moreover, the basic principles of analysis share interfaces and similarity with the beam evaluation technique in the casing substructure evaluation (section 5.1.3).

The major point of interest in this context is to retrieve suitable entities for the beam element application as curves or lines combined with associated and well-founded properties for the individual beam elements. The identification process already provides a best-fit bounding box based on the calculated component principal axes. The bounding box in turn reveals the major axis, which is used to construct the beam center line.

Discrepancies on the outer beam boundaries, so of the concrete box surface, are avoided by shortening this center line symmetrically by a specified parameter. The bounding box orientation and the center line are input to a scanning algorithm, which introduces virtual scanning planes to evaluate the respective local beam cross-sections. For this purpose, projection curves of the beam surface are created on the plane. These projection lines produce closed loops, so result in a bounded face or faces respectively.

5.2.2. Application

Two application cases have been selected for demonstration purposes in this thesis from the beam structure domain, namely B1 and B2. Analogously to the approach in the casing section, the models are presented first and modal validation is used to evaluate the quality of the transferred models.

The first model B1 and its derived variants are shown in figure 5.38. The settings which can guide the beam transfer are mainly options regarding scanning resolution and tolerances for the selection of cross-sections, as described previously. Model B1-A has been generated by choosing a coarser tolerance for the cross-section evaluation in contrast to model B1-B. Due to the simple topology of the model B1, both settings do not influence the beam generation to an extent which is visible on the first view. Nevertheless, the settings do appear to have a noticeable influence on model behavior, as it can be seen in the modal analysis results in graph 5.39.

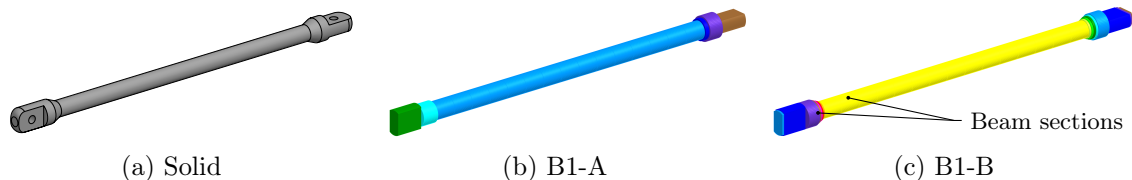


Figure 5.38.: B1 models

As expected, the performance metrics show a high degree of computational reduction, especially regarding the DOF count and data size, which improves model manageability. The relative required computational time, however, is high compared to previous investigations. This can be traced to the fact that a certain minimum amount of time is required for setting up the solver, which is not avoidable. Model B1-B also shows a better matching volume due to a finer resolution in the beam transfer process.

The model B2 depicted in figure 5.41 resembles a beam structure with a lower dimensional ratio compared to B1. The distance of the resulting beam nodes to the original reference 3D nodes is larger in such models relative to the longitudinal dimension. This

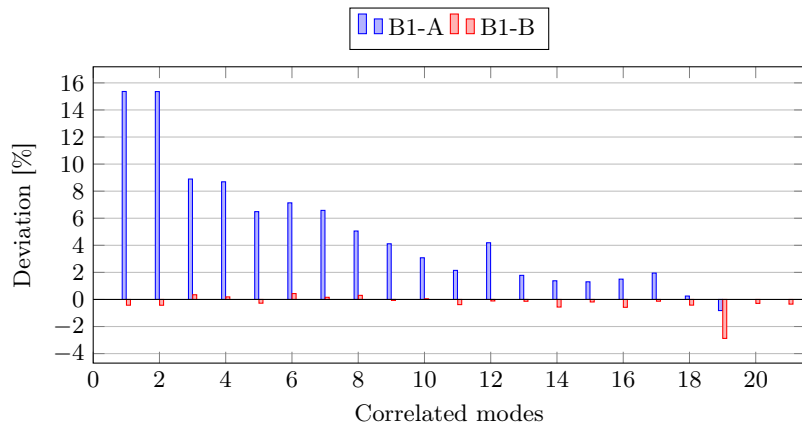


Figure 5.39.: B1 results

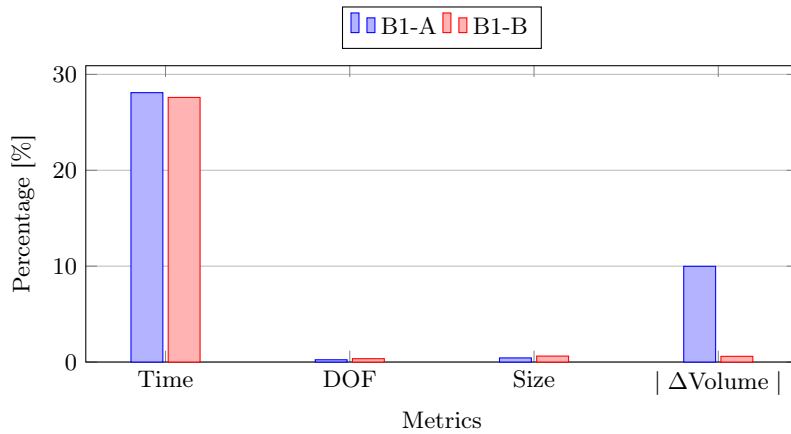


Figure 5.40.: Evaluation of metrics of model B1

impedes correlation and mode comparison because local behaviors, or more specifically displacements, cannot be replicated in a sufficient way in most of the cases using the 1D elements. This also leads to less correlated modes and higher relative deviations as visualized in figure 5.42. The conclusion which can be drawn from the metrics of these model studies in chart 5.43 is similar to the previous application case: the transfer to 1D beam elements entails a significant simplification potential for suitable application scenarios.

Compared to the casing transfer procedure, the process of converting a beam structure to a 1D representation with suitable cross-section is relatively straightforward, which is noticeable in the process performance evaluation. Depending on the specified scanning resolution and desired tolerances, the time required by the process can vary of course.

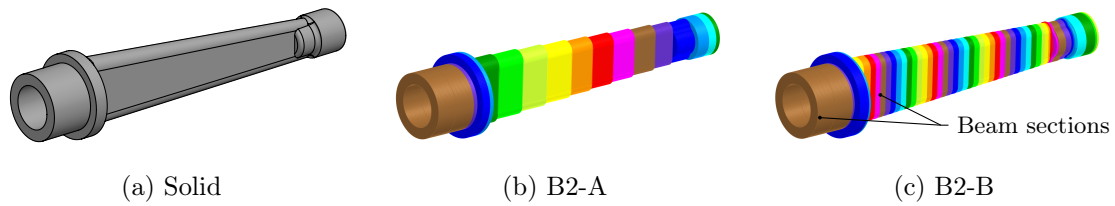


Figure 5.41.: B2 models

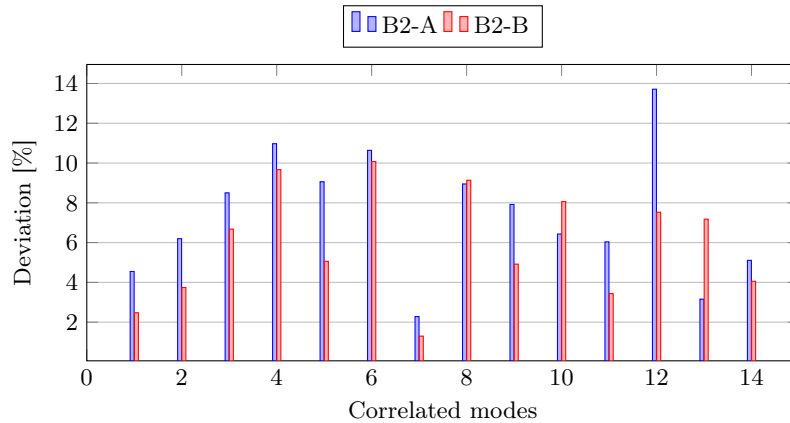


Figure 5.42.: B2 results

Despite this, the maximum time required for all shown application cases has been approximately 8 seconds with an average below 5 seconds.

5.3. Interface mapping

The relation between components in the form of interferences and interfaces has been a key point in the recognition framework. As described in the associated chapter 4, this type of information is gathered with regard to geometric entities and stored in a database. If geometry modifications are taking place, considerations about their effect on the geometry which is the basis for the mentioned component relations are required. The developed approaches have been enriched for this purpose with methods that investigate such present conditions based on the database. The model preparation steps can create new B-Rep entities, remove existing geometries or alter the internal geometry identification and access. This implies that if such a change is occurring, new, adequate and related entities have to be either created or mapped to the original ones to ensure consistency. From a process perspective, this task also shares similarities to the manual cycle in which the engineer uses existing knowledge about the original state and purpose of a simplified

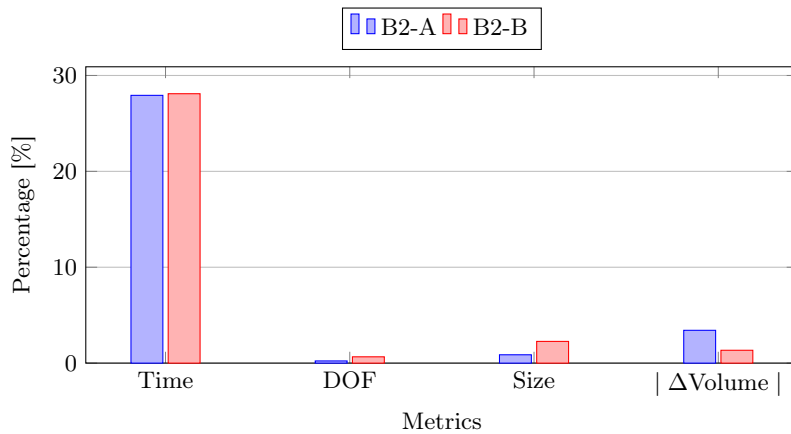


Figure 5.43.: Evaluation of metrics of model B2

geometry, what in turn resembles the short-term memory aspect. By this, the engineer is able to translate appropriate boundary conditions, even if the geometry is abstracted to such extent that it does not allow a clear conclusion about its purpose anymore.

A first technique pursuing this aspect has been mentioned in section 5.1.4.1, or more specifically in chart 5.13. Flange interfaces take a crucial role in structural analyses, which is why a suitable mapping is mandatory, which furthermore allows optional model preparation and adaption capabilities. The idea of mapping flange interfaces is described in detail in the following with reference to figure 5.44.

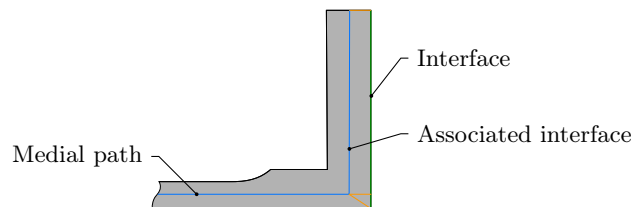


Figure 5.44.: Flange mapping

Exploiting the advantage of the 2D domain, the original flange interfaces are projected and associated to the 2D polygon. After the medial axis structure has been built, a point sampling process on the interface and on the medial axis paths is done to find the path segments which are related to the original interface. As described in the related section, the line afterwards represents the basis for a revolved sheet in the CAD domain. In summary, the 3D interface is projected to the 2D space where it is mapped to a 2D segment, and so can be re-associated to the newly generated 3D shell face or faces. Modifications as offsetting the flange faces to avoid gaps in the geometric assembly or

even to pursue a beam representation for the flanges are therefore comparably easy to introduce due to the accessible information.

Further methods to investigate general assembly interfaces as shown in figure 5.45 are to be integrated in the framework as well. Figure 5.45a depicts an exemplary interface on a substructure. Introducing a hybrid simplification, as shown in figure 5.45b, implies the circumstance that the concrete geometric entity required for the interface remains in the system. However, such operations commonly lead to a change of the internal identification in the topology, which requires a re-mapping algorithms. If the geometric entity remains similar like in the stated case, the mapping process utilizes a simple topology-focused approach which investigates face type, areas and dimensions to find the equivalent new geometry objects. In a more path implying modifications as the imprint approach shown in figure 5.45c, the original interface topology is not existing anymore. A simplification of geometry often also entails a simplification of interfaces. The approximation tool set selected in this context is the set of the developed bounding box methodologies. Each interface is abstracted to a bounding box construct which is projected to the geometric entities associated with the simplification technique. In this example, the associated topology is the imprint face, which belongs to the original boss structure. This bounding box then on the hand can provide information about all contained geometric entities from a CAD perspective, but also is transferred to the CAE domain where it can be used to find contained FE elements and nodes. The information which is gathered thereby can be accessed directly for the setup of structural connecting elements.

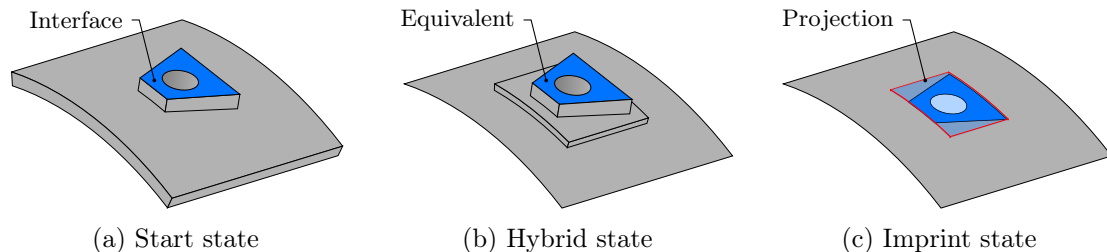


Figure 5.45.: Assembly interface mapping

A similar perspective of abstraction has to be considered in the beam simplification path. Figure 5.46 shows a segment of a beam structure with its associated system interfaces highlighted. After transferring the beam to a 1D construct, these interfaces have to be re-mapped to corresponding elements. In this context, settings in the beam process have been integrated which consider the assembly interfaces in suitable scenarios for the definition of start point or end point respectively of the beam curve. This allows to map

all interfaces to the created point object in a sophisticated way.

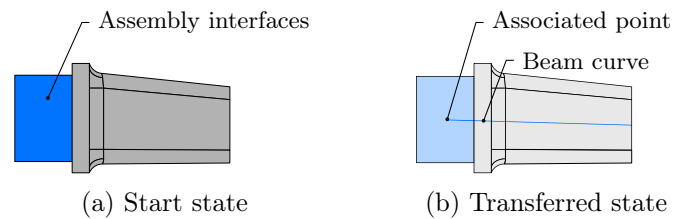


Figure 5.46.: Beam interface mapping

Re-mapping means retrieving the newly associated information and storing it as accessible data for the subsequent processes, which for example connect and build these interfaces in the CAE domain. An excerpt of the original database is created within this process, which is then altered and updated using the gathered re-mapping information. Each interface element in this database additionally contains an identifier about the type of interface to streamline the downstream tasks as for example: face, point, solid, bounding box. Finally, the altered excerpt is replacing its corresponding fields in the overall database index.

5.4. Assembly domain

5.4.1. Framework architecture

After having developed the methods and processes dedicated to the component categories, these are integrated in the top-level system framework, where connections and interfaces to the recognition framework are built. The information retrieved from the recognition steps has to be mapped to the inputs of the category process pipelines. The results thereof are in turn representing the building blocks for the final transferred assembly.

For the development of the strategy and associated framework architecture, different aspects have to be considered. The performance aspect is a cardinal contributor to the applicability and suitability and advantage of the automated process, therefore an omnipresent objective. The used CAD/CAE environment is limited by its technical foundations to single-core CPU usage, which is implying a drawback, especially in times of advantageous parallel computing architectures. In this regard, the strategy pursued in this work is to exploit this single-core limitation and convert it to an opportunity for improving the overall performance via for example cloud-computing.

The developed basic architectural principle consists of a major managing process stream which orchestrates and guides minor working threads, introducing parallel working

environments as illustrated in figure 5.47. Due to the actual single core limitation, the parallel environments are able to work independently, which also simplifies the resource management and allocation. A crucial requirement for the suitability of this architecture is an adequate setup of data stream interfaces and data aggregation procedures.

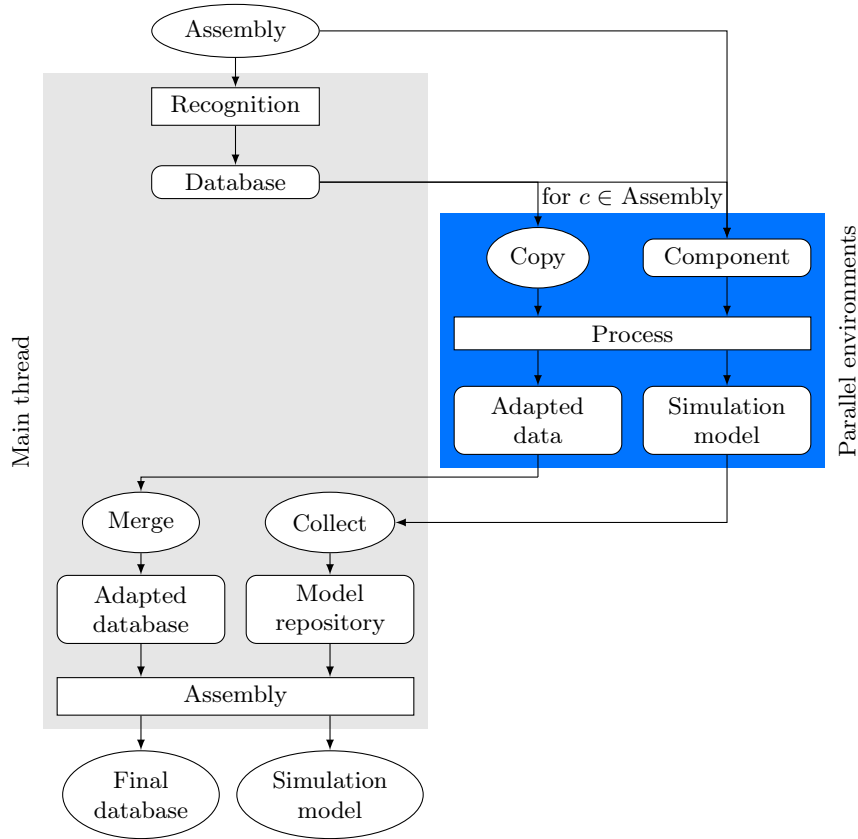


Figure 5.47.: Transfer framework process

The recognition database which has been built by the recognition framework for a present assembly serves a major organizing purpose in this regard. It defines the associated category process types and contains additional data about interfaces and component context. Starting from the database, the worker environments are initiated and assigned with a respective component and task. These are extracting required information associated to the component from the database and building a standalone partial database which maintains the fundamental hierarchy and structure of the original one. The attributes and hierarchical relations stored in the XML database are representing the target anchors for this selective extraction process and allow an efficient and explicit access. In addition to this partial database, the associated component is fetched from

the assembly and stored accordingly. The component category initiates the associated process for the part and database, which produces an adapted database and the converted simulation model. The new and replaced XML elements for the corresponding flange faces or flange hole entities when converting a 3D structure to a shell midsurface model are examples for adapted database entries.

The set of produced partial databases and transferred components are to be collected, stored and assembled in the managing thread afterwards. The components are stored in a repository and accessed from the CAE environment and mapped to the original CAD counterparts in the assembly in order to create the assembly FEM model. Due to the consistent hierarchy and unique element associations in the database, the partial databases can be merged without conflicts to a final and adapted one. This and the model repository represent the inputs for the final assembly file which consists of steps as model mapping, interface and boundary conditions setup and data association of FE entities with the corresponding geometric entities.

After having mapped the individual simulation components to the assembly FEM part, the process moves over to the introduction of boundary conditions. A detailed investigation of different forms of boundary conditions and their respective simplification techniques are not part of the objectives of this work. The aim is rather to provide a suitable interface for an automated implementation of the desired FEM surrogate constructs. Consequently, basic template connection types are used for the demonstration in this context. The idea behind the pursued strategy is that the automated framework is providing all potentially necessary interface information and the user can set up desired connection types as modules, select the desired type and apply them automatically to the related interfaces or even by a drag-and-drop way in future developments.

In this work, the glue boundary condition type is used to connect face contacts while basic NASTRAN RBE3-CBUSH-RBE3 templates (see section 2.4) are implemented as simplified and abstracted bolt connection surrogate. Beam structure interfaces are reduced to single nodes due to the 1D conversion. These commonly have to be connected to an interface face, or faces respectively, on the adjacent component. The interpolating RBE type is used to connect these beam ends to the associated surfaces. Bearing interfaces are often replaced by interpolating elements to generate suitable interfaces between the rotating and stationary parts and to be able to separate both systems. Load application or extraction from the bearing center nodes after these have been connected to the surrounding structural interfaces allows a straight access also for post-processing or sub-assembly modeling.

The geometry-oriented and smart recognition capabilities provide additional opportuni-

ties, especially with respect to post-processing. Besides steps as analyzing an assembly, transferring components to suitable simulation models and setting up a simulation assembly model, the manual process scope also can be extended to the evaluation and post-processing of results, associating simulation results to geometric, assembly or FEM conditions in order to gain understanding in the design phase. The automated tracking of recognized geometric entities through all phases of modifications, adaptations or simplifications to the implementation of associated FEM constructs allows associating the entities with the resulting FE elements, nodes and properties. This offers the opportunity of a sophisticated structure grouping and a geometry-based post-processing and access to all desired properties. On the one hand, the identified substructures as described in section 5.1.3 are associated with the respective corresponding FE entities, and on the other hand identified bolt connection groups are brought into relation with the corresponding bolt surrogate entities. This process of FE entity mapping is conducted with its results added to the adapted recognition database for potential future purposes.

5.4.2. Application

The capabilities of the developed framework are now demonstrated on exemplary application cases derived from the aero engine field, which is defining the context of this thesis. Both demo assemblies A1 and A2 have already been shown in the recognition chapter 4 and serve as examples for this application as well. The assembly A3 is a more detailed assembly of a part of an aero engine core and contains components which have already been shown in the transfer process, so serves the purpose of demonstrating the interoperability of the built methods. To showcase also more complex scenarios, assembly case A4 is derived from a Digital Mock-Up (DMU) model which is comparably highly detailed and shall put emphasis on the suitability of the developed methods for an industrial use.

Especially in the more complex cases, a comprehensive simulation and also the respective analysis of the model quality of the relatively large assemblies is demanding high computational resources. Generating and simulating a highly detailed and fully featured FEM reference model for the validation methods selected in this work is exceeding the common standard computing power. Consequently, such investigations recommend an outsourcing to high-performance computing environments or require more sophisticated validation methods. With regard to the objectives and focus of this research work, these plans are significantly exceeding its scope, which is why these are neglected in this section while setting the focus more on the application of the developed methods and the resulting

performance of the introduced strategies.

In the following figures, the beam and shell elements are visualized with their associated shell thickness or solid cross-sections for visibility purposes. Different physical, so FE, properties are indicated with different colors for a better visible information about the model.

The simplified model A1 is a demo assembly in which the structural components are designed with comparably few details. The recognition framework detects the single real casing structure and the substructure analysis recommends a hybrid representation for its boss substructure based on the set values. The flanges are introduced as glue connections while the bearings are replaced by interpolating elements referring to a respective center node, see figure 5.48. The detail view focuses on the introduced hybrid representation and shows the smooth thickness transition of the surrounding shell to the original 3D solid geometry, including the automatically applied connecting elements.

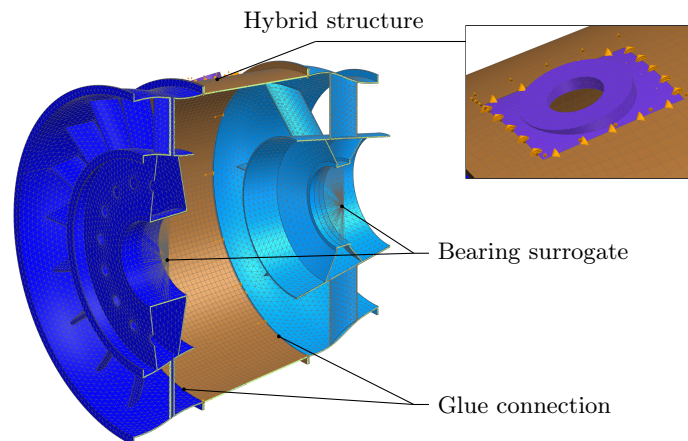


Figure 5.48.: A1 application

The larger model A2 comes with multiple casing components, and therefore depicts more targets for the developed casing transfer methods. The model created by the framework is shown in figure 5.49. Also in this case, bearing surrogates as well as contacts are visible. The strut structures in the rear part are identified by the recognition logics and transferred accordingly to suitable 1D models, as it can be seen in this figure. For clarity and visibility reasons, the beams are displayed with their associated cross-section. The three structures containing the bearing interfaces are critical components for the load transfer, and moreover depict components with a high likeliness to require additional effort and manual tuning and trimming in order to build accurate idealized models of lower dimensionality. Since these have not been considered as clear casing structures,

they are meshed by following the standard 3D path.

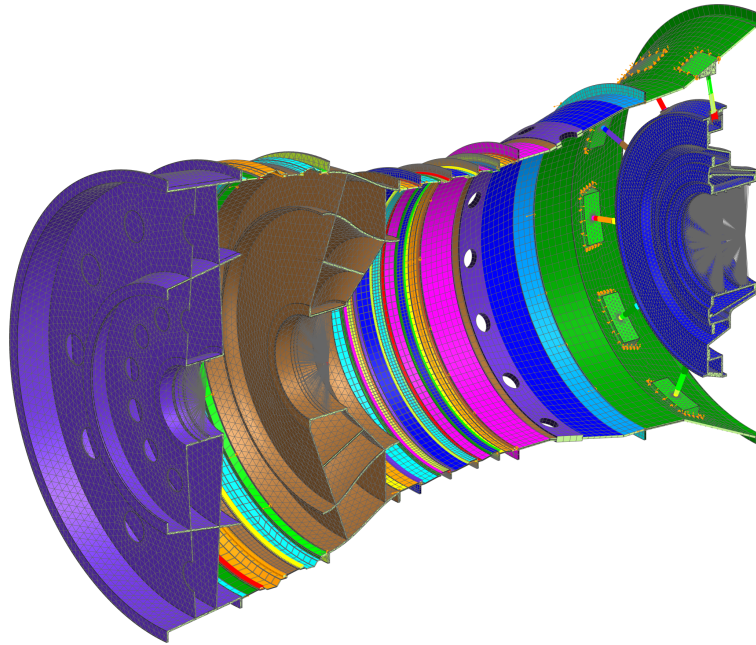


Figure 5.49.: A2 application

In order to introduce additional complexity and present more practical and industry-related models, A3 and A4 have been selected. The application of the framework on model A3, see figure 5.50a, results in the simulation model shown in the subsequent figure 5.50b. Similar to A2, the transfer result of the identified beam structures is clearly discernible. The following figures 5.50c and 5.50d show a detailed view of both bolt connection implementation and the transfer of beam interfaces to interpolating elements. For visibility reasons, the color scheme and solid property display style has been removed to shift the focus to the entities of specific interest. In this type of study, small holes as flange holes are neglected for simplification purposes, so no information about the original conditions is present in the model. Since the framework knows about the concrete bolt or flange hole positions including their respective parameter similar to an engineer who can still access knowledge about the original geometry, it is able to create exemplary interpolating elements using a virtual search radius on the exact positions. The knowledge of associated holes, or even flange pairs, allows an automated setup of bolt connections through two or multiple components. Figure 5.50d shows an isolated view of the integration of the 1D beam elements. The data about the original interfaces of the 3D beam structure allows converting this interface to interpolations for the resulting 1D

elements.

The last application case shows the entire structural core of an aero engine which is used as DMU model. This implies additional complexity due to a high number of topological entities as well as a high degree of their mutual entanglement. Due to the developed segmentation approaches, a certain level of abstraction is introduced, which can help to handle and understand structures in a well-founded way. Some casing components as the split casings are processable by the segmentation approach, but the derived substructures point to a high degree of complexity and indicate an extent spanning the full component, which is why the process decided for a more reliable 3D approach instead. Analogously, the bearing structures do not depict obvious casing components, and consequently are considered as standard 3D parts in the simulation model.

After the application cases have been presented, the last paragraphs are serving the purpose to provide information about the performance of the developed processes. For this reason, the time span required by the recognition procedure and the subsequent major steps of the assembly framework are plotted in figure 5.52.

The conversion label describes the amount of time required for converting all components to suitable FEM models within the parallel environments. As a consequence, this step does not show a linear behavior because of it being rather dependent on the selected number of parallel working cores. Since the architecture is programmed as such that it waits until all component pipeline processes have finished, the components which demand for the longest conversion time also define the minimum conversion duration. This aspect is software-related, with also meshing making up a large part, and so is also unavoidable in the manual procedure. In the assembly case A4, for example, the 3D meshed components occupy a high proportion of the required conversion time.

The next legend entry named “Assembly” is comprising all the tasks required for building the assembly FEM model which involve the initial setup of a simulation model, the boundary conditions integration and the extraction of FE entity information to the database. Since these processes are conducted on the major thread, they are limited anew by the existing single-core capabilities, and thus show a comparably slower progress. Components which require more assembly steps also demand for more resources as it can be seen in A2 and A4.

The last label contains miscellaneous tasks as initializing the CAE session and extracting a solver deck for the simulation model which is proportional to, inter alia, the model size.

In conclusion, it is evident that the presented methods and variables can take a comparably long time, but despite this, the relation to the manual process has to be considered in every case. Especially the setup of bolt connections in an assembly, the

implementation of beam elements including the selection of adequate beam properties or the setup of suitable shells combined with well-founded thickness and offset properties can become extraordinarily tedious in the manual path. A lot of manual inputs as for example mouse clicks, interactions or analyses are guiding the manual path of transferring a geometry assembly to a suitable and idealized simulation model. All these steps and logics have been central to the strategy and methods developed in this chapter within the automation context.

In lights of these considerations, the automated process provides benefits not only in terms of process efficiency and consistency but also regarding further-reaching opportunities. Besides the knowledge of entities derived from the recognition framework, the presented approach also offers additional information about specific identified component substructures and creates a connection between FEM objects and their respective parent geometry. This results in a smart holistic framework which provides interfaces to a plethora of more downstream purposes as the automated result post-processing based on geometry or automated and simulation-driven design optimization.

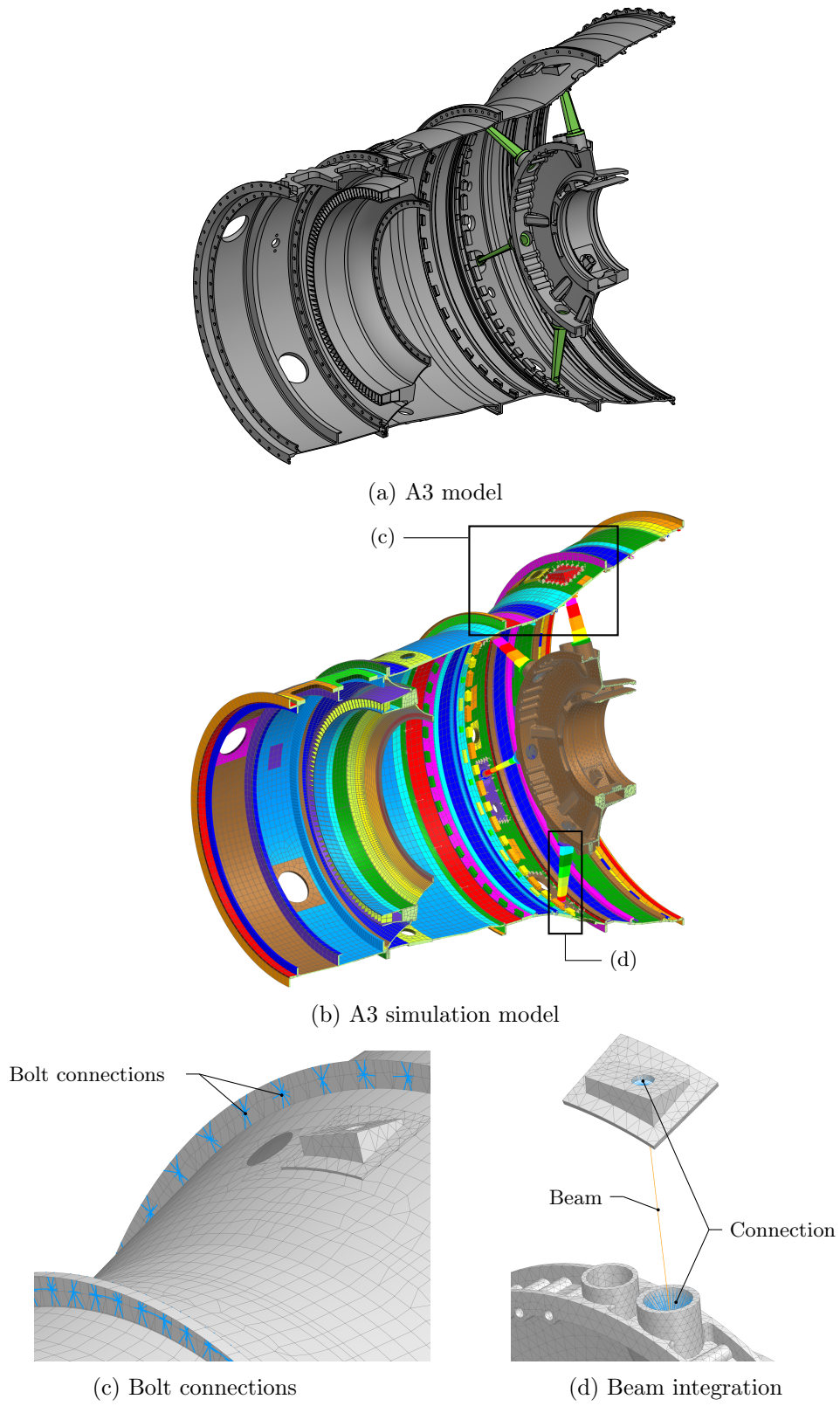
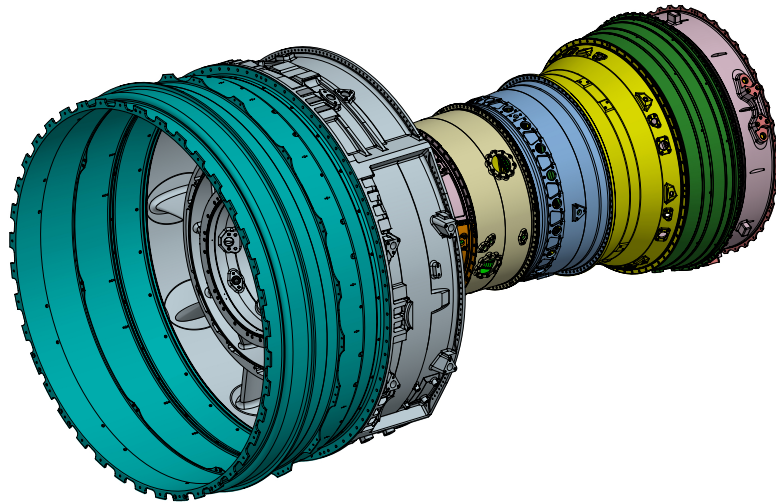
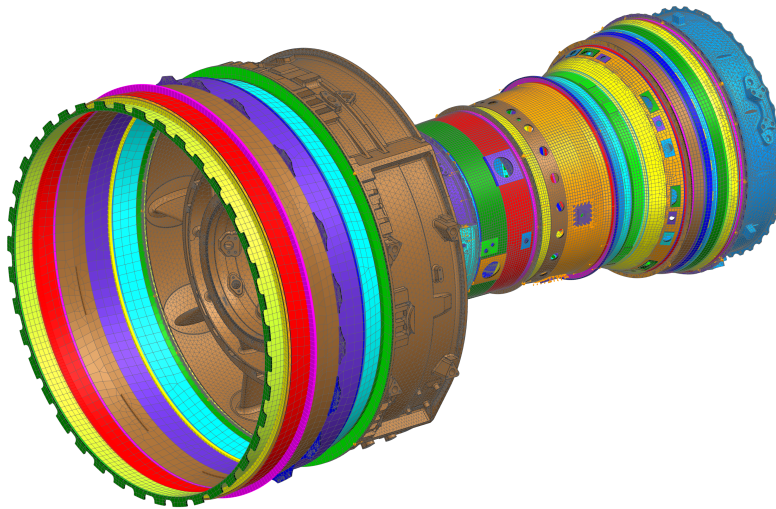


Figure 5.50.: A3 application



(a) A4 model



(b) A4 simulation model

Figure 5.51.: A4 application

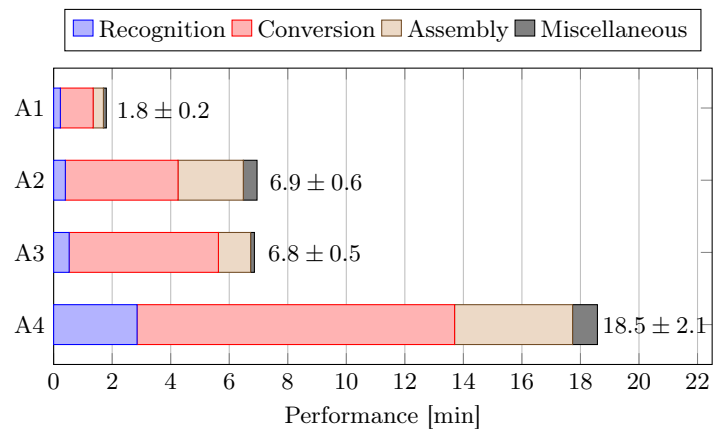


Figure 5.52.: Assembly process performance evaluation

6. Knowledge Representation

6.1. Objective

The previous chapters are dedicated to two major steps for the generation of simulation models starting from a CAD assembly. Chapter 4 presented research with its derived methods to cope with the cognitive tasks, identification and short-term memory aspect. Within the manual process, an engineer is still able to remember for example positions of flange holes from an original fully featured 3D model, even if the model is simplified to such extent that the holes are not present anymore. If the engineer started from this abstracted and idealized model, no information about the actual hole positions would be accessible. This condition also applies for an automated process and has been brought into relation with the short-term memory aspect. The subsequent processes which transfer suitable component and component categories to simplified model representations are the main point of focus of chapter 5. These create additional identification and analysis capabilities on the one hand, and on the other hand deal with the aspect of manual effort and tasks within the process. In conclusion, the analysis, short-term memory and manual effort aspects introduced in chapter 3 have been covered by this research so far.

The remaining factor on this list of generic design aspects is the long-term memory. An engineer or an engineering team or department is conducting simulations, post-processing the results, and based on that gathering experience about modeling conditions, behaviors and circumstances. Moreover, basic physical understanding is an important prerequisite for the expert role. These expert skills allow estimating or assessing the influence of certain modeling decisions prior to their application, so support the decision for a modeling strategy itself.

Since this plays a cardinal role in many design cycles, this last chapter is presenting a conceptual approach to cope with this aspect. Experience gained by simulations directly relates to a training process. This term is often referred to in the field of Artificial Intelligence (AI). Along with the process itself comes the data to learn from or train on that has to be present. Compared to other domains, data in engineering can be rare, highly unstructured and relatively hard to access or aggregate.

The process automation therefore depicts a key enabler for Artificial Intelligence applications. This allows to create training data or general data in an automated way which drives efficiency, so increases the potential use cases. From this perspective, the methods developed and presented in this work are predestined for such intentions. The purpose of this chapter is to present a concept about how the automated methods can be combined in order to create training data which shall serve as decision support for the design cycle. The subject of the decisions in this concept are related to the different types of idealization and representation variants as shown in the previous chapter. Analogously to the experienced engineer who chooses an idealization procedure based on visual inspection accompanied by knowledge, the objective is to develop a strategy which is pursuing an artificial experienced-based decision-making for substructure representations.

The starting point for the developed transfer methods is a geometry or assembly, and so is for the training process. To significantly boost efficiency and to automate the generation of training data, the first focus is set on an automated process for generating a geometry set.

6.2. Data generation

6.2.1. Geometry

In section 2.3, the term parametric modeling [4, pp. 149 sqq.] has already been presented. Parametric and feature-based models offer a lot of advantages for both adapting and creating geometry quickly and consistently. Especially in early design phases, parametric models recommend themselves for design exploration investigations or efficient and geometry-based optimization. In order to build a database for the evaluation of effects of different geometries, a set of input geometries has to be built in the first place. To generate a reasonable pool of models, the advantages of this feature-based and parametric modeling have been exploited within an automated workflow. Since this thesis is situated in the context of aero engines, the created pool is aligned to the related use cases.

The starting point for the model pool generation is the development of three different feature-based parametric CAD models. The first one is depicting a single simple quasi-axisymmetric component with few substructures attached to it. Exemplary parameters thereof are able to control the dimensions as thicknesses, lengths or radii of both main structure and substructures as well as general boolean expressions, shapes, positions, pattern parameter or distribution related values. The setup of robust parametrized models requires investigations of possible parameter design spaces and parameter correlations. A

compromise of number of parameter and limitations therefore has to be found in order to allow reasonable and efficient geometry changes without restricting design possibilities to an undesirably high extent.

All these parameters are externalized to create access via file manipulation and the API of the CAD environment. This allows to update the local geometry parameters by importing associated parameters and subsequently updating the model according to them within an automated process.

To introduce assembly context, the second base model is an assembly consisting of two structures connected via a flange connection and typical substructures. The third base model extends this aspect and introduces a third participating component.

After having set up the base models, the strategy pursued to create the model pool is presented next. The model setup provides the possibilities to automatically change the geometry. This possibility is used and integrated in an automated workflow, which is depicted in schematic form in figure 6.1.

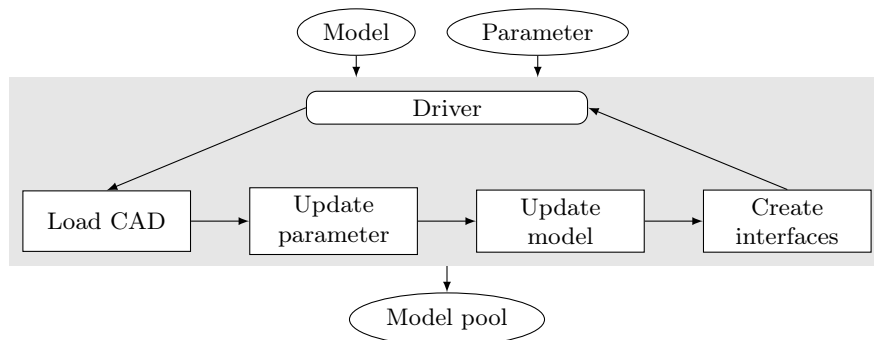


Figure 6.1.: Geometry generation workflow

This workflow has been implemented using the software Isight [284] by Dassault Systèmes, the API of Siemens NX and a set of adapting scripts using the programming language Python. The first step is to retrieve the base model and the associated parameter file and load the part in the CAD environment. Afterwards, the local parameter values are automatically overwritten by the values in the file, which results in geometrical changes. This leads to the required model update afterwards. The subsequent steps mainly focus on a reasonable setup of the model repository and to automatically create the interfaces for the next steps in this approach. To manage assembly and component dependencies as well as to generate a structured folder hierarchy for consistent data extraction purposes, additional steps for naming and storage convention have been implemented. The later process which uses the created model pools require additional inputs which are stored in

the configuration file produced by each workflow execution for the associated model.

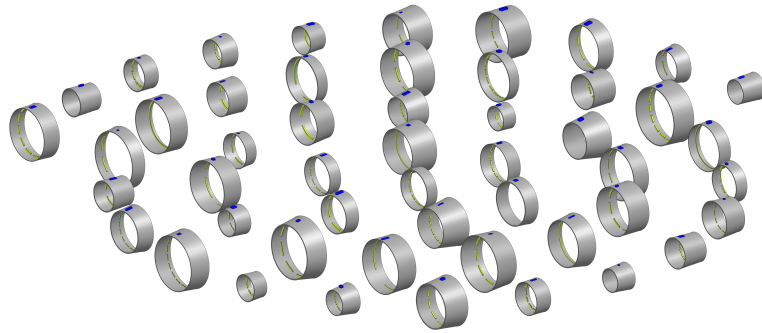
The last component of figure 6.1 to be described is the component named driver in this work. This driver block can take different roles within the automated workflow as provided by Isight as for optimization, robustness analysis and sensitivity study purposes. The aim of this part of the work is to create a set of different geometries, which is why the parameter variation aspect of the sensitivity study type is considered. This driver type allows changing a set of parameters either systematically or by introducing randomness factors. Several methods and approaches to set up design matrices for Design of Experiments (DoE) studies have emerged with Optimal Latin-Hypercube Sampling (oLHS) [284–287] being implemented in Isight and selected for the given purpose. The oLHS technique uses a statistical approach to generate samples of parameters for multidimensional distributions. These are processed by a space-filling optimization algorithm which maximizes the minimum distances between sample points in the design space. This leads to a reasonable coverage and sampling of the design space which fits the objective of creating various geometric designs. The design matrix resulting thereof contains n different combinations of parameters which represent single executions or runs in the workflow.

For the given base models including their associated parameter set, these design matrices are generated and fed into the automated geometry generation workflow. The created geometry pools for each base model are shown in figure 6.2.

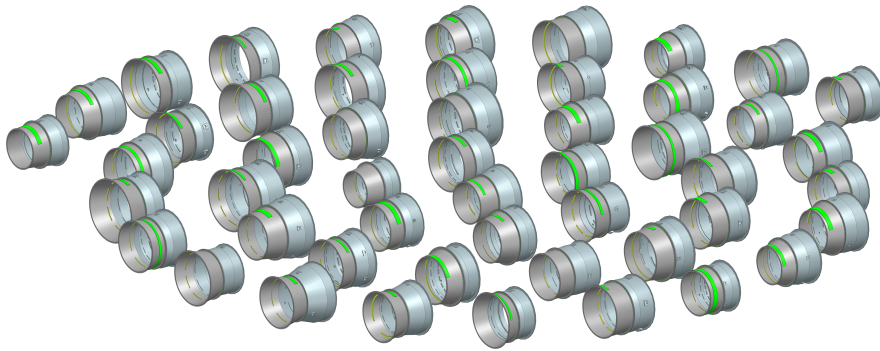
Although different model types are introduced and varied, the model variety can be described as comparably small when extending the scope also to other fields of engineering. The limitation to the aero engine context can reduce the overall application spectrum later. This can lead to the conclusion that learning from this set can become too narrow-minded with fewer capabilities to adapt to other structures. On the other hand, the aero engine core design is not expected to change to a high extent either in considered development time spans due to the fundamental working principles. Moreover, engineering experience and judgment are also often related to the respective expert domain of the engineer, and so the involved principal learning process is also comparably specialized. This approach represents a conceptual strategy which suits these scenarios in this regard.

6.2.2. Simulation models

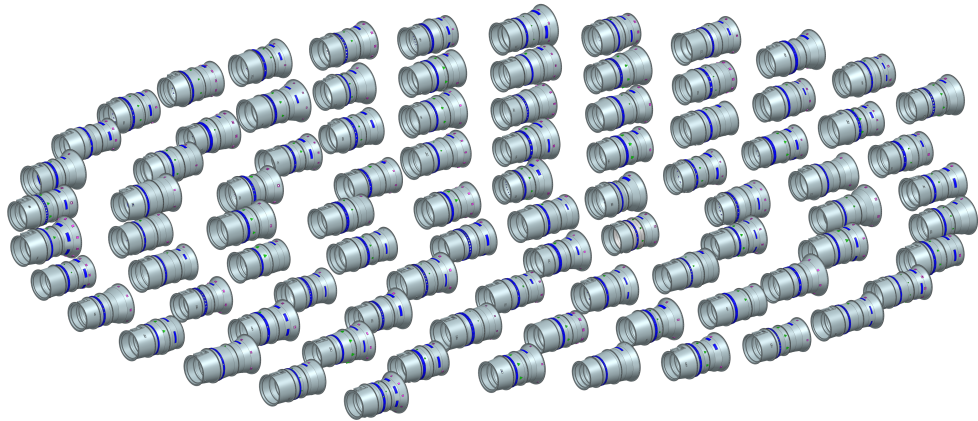
After having defined the geometry foundation, the next point on the agenda is the creation of simulation models. In order to evaluate the influence of substructures, or of different substructure idealization techniques respectively, different variants have to be studied, similar to the model variants which have been part of section 5.1.6. Consequently, the



(a) Set 1



(b) Set 2



(c) Set 3

Figure 6.2.: Model pool

aim is to create a set of simulation models with various level of detail for a base CAD model from the model pool.

Another top-level layer has been added to the transfer framework architecture illustrated in figure 5.47 in section 5.4.1 for this purpose. Especially with respect to the generation of large amounts of data, parallel and efficient computing is a crucial advantage or even prerequisite. Based on this, the parallel environments introduced earlier have been extended to deal with more tasks in this context, as shown in 6.3.

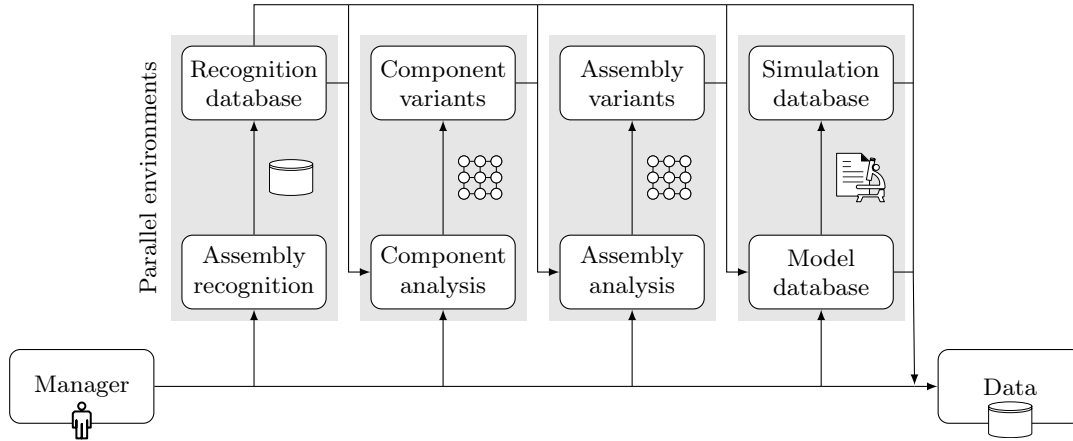


Figure 6.3.: Simulation database process

Similar to the basis architecture, a major managing thread is orchestrating the steps in the parallel environments and delegating the tasks. First, the recognition procedure is initiated to gain information about the present model, which results in the known recognition database. Based on the information contained in this database, the components are guided to the specific respective process pipeline. This is where the first process adaptation has been introduced. Additionally to the substructure analysis in the casing related process, an investigation of technically possible representation techniques is conducted. Each substructure i is therefore assigned with a feature vector \mathbf{x}_i (see table 5.2) and a set of values \mathbf{r}_i describing the possible idealization approaches.

From this set of potential simplification decisions, the full factorial DoE [284] parameter variation approach yields n_x^c combinations for the component x with

$$n_x^c = \prod_{i=1}^k |\mathbf{r}_i|, \text{ with } k = \text{Number of substructures} \quad (6.1)$$

which are stored in a design matrix \mathbf{D}_x^c where c denotes the relation to the components. The subsequent simulation and objective of this approach requires a reference model

which is represented by a fully featured and 3D variant, which is therefore added to \mathbf{D}_x^c as an additional entry. Extending the scope to assemblies containing multiple components, which can be the target for simplifications leads to the assembly design combinations n^a generated by the component variants n_x^c for $x \in \mathcal{C}^a$ with \mathcal{C}^a representing the components which are part of the assembly a . The resulting combinations are stored in \mathbf{D}^a as the assembly design matrix. Analogously to the component context, a reference row existing in \mathbf{D}^a which contains the reference fully featured versions of the participating components.

Such designs of experiments, especially the full factorial sampling, are often leading to a high number of evaluations in case of a high number of factors, thus often imply computational effort and complexity which anew emphasizes the advantages of parallel processing.

The next implemented parallel environment focuses on an efficient generation of the n_x^c simulation models based on \mathbf{D}_x^c , so the creation of the standalone component models $\mathcal{M}^x = \{M_1^x, \dots, M_j^x\}$ with j describing the respective design matrix row count and x being one component of the assembly. After the design matrices for each component have been processed, the resulting pool of the variants of all components $\mathcal{M}^c = \{\mathcal{M}^1, \dots, \mathcal{M}^l\}$ with l as number of assembly components depicts the basis for producing the assembly models according to \mathbf{D}^a if an assembly context is given. The assembly variants are a combination of the associated component representations from \mathcal{M}^c . The approach of building all component variants first to create a data pool allows the assembly process to simply pick the corresponding parts thereof. This ensures that duplicate efforts and processes are avoided, thus significantly improves performance. On account of the recognition and knowledge aspect of the embedded processes, there is also sufficient information at any time to combine and connect each participating model accordingly. This is resulting in the model database \mathcal{M}^a containing all assembly variants for a given assembly. Based on this, the complete model database \mathcal{M} is a combination of both standalone component \mathcal{M}^c and assembly models \mathcal{M}^a of each CAD model. This database is the starting point for the last group of parallel activities.

The automated process is also setting up a simulation task template for each simulation model in \mathcal{M} with regard to the goal of evaluating the model quality. In addition to this, the process creates an executable file which initializes the CAE software, outputs the simulation configuration file and directs it to the corresponding solver. This introduces comparably simple process standalone capabilities. On the one hand, this offers the opportunity to copy the folders containing simulation files and the executable for outsourcing the simulation to multiple computers, and on the other hand already provides an interface for local process automation. These execution files have been set up in an automated

resource management framework and executed to produce the data.

6.2.3. Data setup

The simulation results are often unstructured, not directly accessible and involve undesirable amounts of data which affects data manageability. An automated procedure has been developed which accesses and extracts desired information from the commonly binary result files for this purpose and stores this information in a more accessible data structure which fits the use case of this work.

The Hierarchical Data Format (HDF) structure, or HDF5 respectively, can offer significant advantages and improvements. Besides providing fast and efficient I/O operations, it allows storing n-dimensional data sets and also complex objects [288]. These data sets can be set up in a clear hierarchy which allows easy access in downstream processes or for appending information to an existing database. The advantage of common Comma-Separated Values (CSV) files is the non-binary format, which allows uncomplicated user evaluation and interaction. However, especially in big data situations, the manual processing is comparably inefficient anyway, and so binary file formats as HDF5 do not depict a significant drawback. Moreover, setting up a clear access structure or hierarchy with CSV files is not recommended or only possible via additional text manipulations. Structured Query Language (SQL) formats have advantages but contain a lot of overhead which is not required in this context, and thus the performance disadvantage outweighs its advantages. An I/O evaluation has been done to support the decision using a large matrix and several runs with its averaged results are shown in figure 6.4



Figure 6.4.: Evaluation of database formats

As a consequence of these considerations, the process extracts information required for the modal analyses, as there are eigenvectors, eigenfrequencies, node positions and

other values which could be useful for downstream use cases and stores it in separate HDF files. The modal evaluation and comparison is then conducted using the contained data. Each CAD model m is associated with $\mathcal{M}^m \subset \mathcal{M}$ which contains a single reference model and a set of different variants. In this context, m could be either a single CAD component of one assembly from the model pool or the assembly model itself. The extracted modal results are therefore processed with the respective reference results using the MAC for modal correlation and the corresponding eigenfrequency pairing and comparison. All derived information is stored in the top-level database, which follows the architectural scheme depicted in figure 6.5. In this scheme, i defines the base assembly, k a variable for the assembly variants, l the components in assembly i and j an index for the respective component variants. Similarly to previous introductions, c defines the component context and a the assembly context. The model container provides two different types of information. The simulation results, modal related values as well as the correlated results make up one part thereof. The other part is including geometric information as the assembly network and the feature vectors for each substructure of the component.

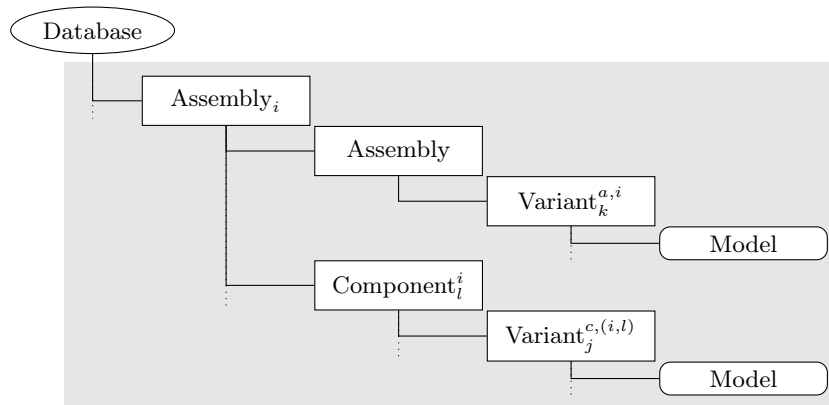


Figure 6.5.: Database structure scheme

6.3. Process development

6.3.1. Strategy

The generated data set is now to be processed with the goal of a suitable type of knowledge representation in mind. A lot of research and methods have emerged in this field of Machine Learning. The related section 2.6, or more specifically 2.6.2, in the state-of-the-

art chapter has already presented several approaches and techniques and also gave a hint on the path chosen in this work.

The data which is the basis for the reasoning process here is highly contextual and implies varying data formats and sizes. This is depicting either an obstacle for the algorithm or an aspect which is extraordinarily hard to cope with. Especially from an assembly perspective, the interconnections and relations between components and substructures are significantly influencing the model characteristics, so are also contributing to the model understanding. General and most common ML methods are built towards structured and uniformly formatted data as in- and outputs, and therefore are not suitable for the present circumstances. Sequential or recurrent algorithms and NN introduce the aspect of context consideration by for example Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) blocks [241, 242] which are additionally capable of considering neighboring data. Most of their application scenarios target linear data sequences, which does also not suit the present problem.

Throughout this work, many strategies and techniques have been leading to and involving a graph representation. This aspect also suggests the use of so-called Graph Neural Networks (GNN) as prototype for the knowledge representation. Graphs inherit an explicit context information and are also able to store node or edge related attributes in a structured form. Compared to common data vectors, tables or images, which are technically matrices, or videos, which are a collection of images, graph representations depict another type of data structure. In general, the matrices or sequences considered in conventional ML approaches are fixed-size grid graphs while graph networks are often occurring without a discrete form or a constant number of unsorted nodes, see figure 6.6. Moreover, the nodes and links in the graph representation can involve a different type with different attributes, while general matrices like images only consist of uniform nodes and edges. This is one of the major advantages of GNN, and, based on this, they are selected for the concept developed in this work.



Figure 6.6.: Data structures (after [260])

6.3.2. Data stream

Consequently, the data has to be prepared and aggregated accordingly to enable a consistent data stream for GNN applications. The database described in section 6.2.3 is therefore further processed towards a graph-dedicated database.

The input data is composed of the set of component and assembly geometric variants. Similar to the recognition network graph shown in chapter 4, the assembly is transformed to a graph representation containing nodes which represent the bodies and links which describe the component interfaces or connections. The component categorization within this framework also assigns a first descriptive value to each node, introducing a node type or nodal attribute respectively.

The level of detail of the graph is further elevated by exploiting the characteristics of specific node categories. Bodies within the casing category can be further subdivided into a set of substructures, thus a single node is transformed to a set of substructure nodes which are interconnected as shown in figure 5.9 in section 5.1.3. The substructure analysis further yields a vector containing information about each of the derived nodes. This feature vector is assigned as the respective node attributes. In summary, these aspects imply a general approach to node type categorization and also a more detailed node description via attributes for separable structures within the graph.

Considering the output associated to these graphs, data for the model quality assessment has already been processed towards a structured form in section 6.2.3. The modal characteristics are central for this purpose with modes correlated by the MAC and their respective eigenfrequencies. To reduce the method complexity in this work and to fit common NN architectures, this data is to be condensed to a single expressive value as the objective output. A point distribution scheme is introduced for this purpose, which is assigning different weights and points to different criteria.

The number of correlated modes is the most crucial criterion since comparing non-correlated eigenfrequencies does not yield a meaningful conclusion. A description of this criterion is shown in the following:

$$M = |\{(i, j) \mid MAC(i, j) > \gamma, \forall i \in Modes_{Reference}, \forall j \in Modes_{Analysis}\}| \quad (6.2)$$

With some modes being similar in their nature, the MAC can yield multiple and so ambiguous potential mode correlations. An iterative approach is added to this basic criterion to find and optimize best-fit correlation pairs.

A potential drawback of this approach is that idealized representations are sometimes

not able to replicate the fully detailed modal behavior as a consequence of, for instance, missing nodes in relevant geometric regions. For example, if a substructure is neglected in a variant despite it being cardinal to a specific mode, the simplified representation is understandably not able to rebuild this kind of mode, even if other model properties are matching. This can imply that a mode which is indeed able to replicate the basic behavior is not correlated accordingly, thus reduces the number of mapped eigenfrequencies for the evaluation. Nevertheless, this effect can be attributed to the fundamentals of the approach and requires additional consideration in the manual process as well, and therefore it is not elaborated further in this concept development.

The eigenfrequencies derived for correlated modes are set in focus next. To reduce the information associated with these, two different values are generated as depicted in figure 6.7 and described as follows:

$$E_s = \max(\Delta EF) + |\min(\Delta EF)| \quad (6.3)$$

$$E_a = \overline{|\Delta EF|} \quad (6.4)$$

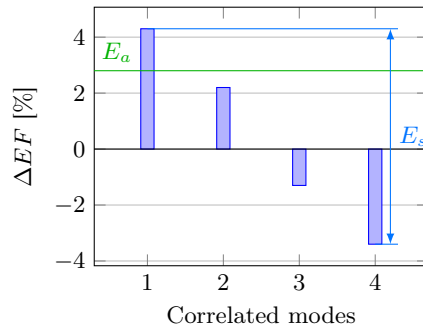


Figure 6.7.: Eigenfrequency deviation values

E_s describes the maximum span involved by the deviations to reveal outlier eigenfrequencies or inadequate model quality. E_a is a simple averaging value of the absolute deviations and serves the purpose to provide a general conclusion about model matching conditions.

These three values M , E_s and E_a are the input to the point distribution principle as shown in table 6.1. The points gathered thereof are then combined as described in equation 6.5.

$$P = \frac{(P_1 + P_2) * P_3}{(3 + 3) * 3} \quad (6.5)$$

Table 6.1.: Point distribution scheme

E_a	P_1	E_s	P_2	M	P_3
< 4 %	3	< 5 %	3	> 30	3
< 8 %	2	< 10 %	2	> 20	2
< 12 %	1	< 15 %	1	> 15	1
≥ 12 %	0	≥ 15 %	0	≤ 15	0

As shown in this equation, the number of correlated modes M is strongly controlling the objective value P . The reliability aspect of M which influences the other values E_s and E_a has been the reason for this definition. The resulting objective value P is normalized and categorized in P_c to three different classes: good (green), acceptable (yellow) and bad (red), see table 6.2.

Table 6.2.: Objective value category

P	P_c
< 0.4	●
< 0.7	●
≥ 0.7	●

Finally, each graph derived from the input data is associated with this single objective value, or class respectively. For reasons of comprehensibility, a schematic visualization of this data structure and also database is shown in figure 6.8. The final dataset created in this work contains $\approx 40\,000$ graphs derived from the executed simulations and can be extended in a straightforward way for future research.

This data stream containing and combining both input and output data is directed to a new database, which serves as direct input for subsequent applications. By using efficient data structures and file formats, the data manageability has been improved drastically, as visualized in figure 6.9. The approximated data flow of the simulation is creating a vast amount of data, which is hard to handle and even complicated to store on the most common computer configurations. A continuous data cleaning and aggregation process, as described in section 6.2.3, is extracting necessary information and storing it in a highly efficient structure and format, the simulation database, and so avoids disk space shortage. By using several steps of extracting and compressing as shown on the horizontal axis of figure 6.9, more intermediate interfaces for potential additional use cases are maintained.

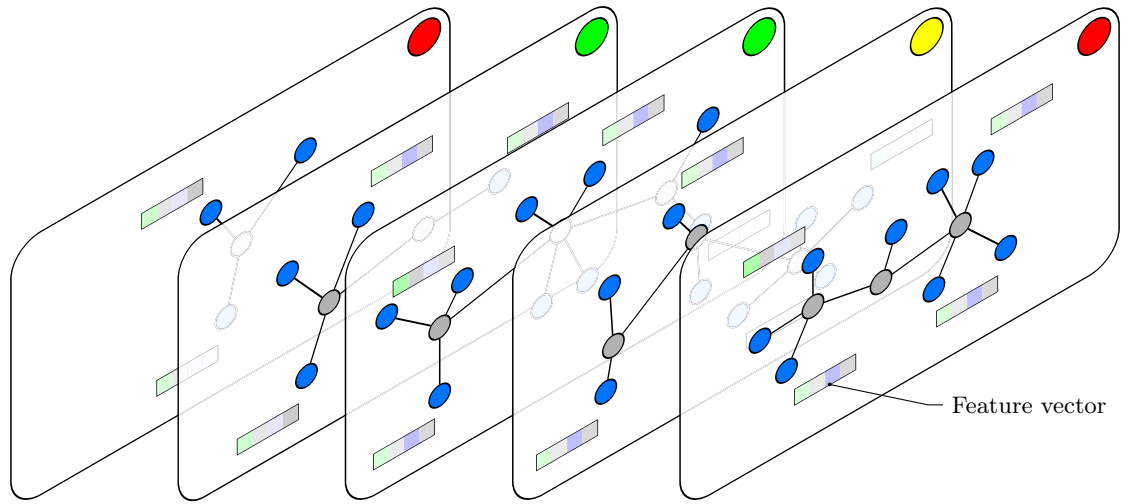


Figure 6.8.: Graph database scheme

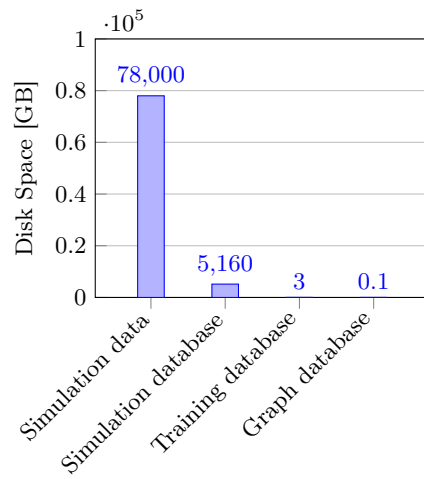


Figure 6.9.: Data extraction steps

6.4. Application

The created data stream is connected to the input interfaces of the learning algorithms. Compared to other fields of ML, Graph Neural Network open-source implementations or applications are rare. The StellarGraph package [289], however, is one of the sources to provide multiple methods for graph-based learning techniques. Most of the general graph techniques, so also the applications of StellarGraph, are oriented more towards classification tasks than regression tasks. This dominance of classification algorithms has led to the decision to create a class-based objective value as described in the previous section. The field of graph classification techniques can be further subdivided into three major subcategories as there are node, edge and graph classification algorithms with the node and edge related tasks being more widespread in the field of open-source graph applications. With regard to this, the objective value defined and presented in this approach is situated in the graph classification category.

The StellarGraph package contains two different methodologies, or neural networks, for the classification of entire graphs. First, the implemented Graph Convolutional Network (GCN) contains a model architecture which is derived from [290, 291]. The second one, the Deep Graph Convolutional Neural Network (DGCNN), depicts an architecture which stacks these GCN layers with reference to [292]. Another approach for GNN graph classification has been published by Zhang et al. in [265] which has already been mentioned in the dedicated state-of-the-art section 2.6.2. This thesis puts the focus on the application of GNN as a conceptual demonstration and the presentation of a potential path towards engineering and modeling guided by machine learning. Therefore, the presented available methods are considered as state-of-the-art algorithms for this concept without expanding the objectives of this work to the vast field of Graph Neural Networks.

The architectural setup of these three models have been adapted to fit the conditions and data formats of the present use case. The connection of the datasets, learning models and output streams is the starting point for the learning algorithms and processes.

The dataset generated in previous sections is split into three different chunks: training, validation and test set. While the algorithm is trained on the training set, a simultaneous evaluation and cross-validation is done with the validation set in each iteration to evaluate and handle overfitting (see section 2.6.2). The test set serves the purpose of finally evaluating the algorithm on so far unknown data. Additionally, dropout layers and weight decay mechanisms ([241, pp. 118 sqq., pp. 258 sqq.]) are embedded in the neural network and are proven countermeasures for overfitting issues. After an optimum is reached, the trained network is applied to the test set to derive a final estimate of its performance.

While the results of both GCN and DGCNN have not shown satisfactory capabilities to learn from the given data, the Hierarchical Graph Pooling with Structure Learning (HGP-SL) algorithm of Zhang et al. [265] indicates a convincing learning progress. In order to optimize this learning process, different approaches to tune, trim and optimize the model can be considered.

The first one is targeting the input data itself. To investigate the importance of a large amount of information on the one hand, and on the other hand to detect data which is perhaps negatively affecting the model accuracy, the nodal feature vector entries are investigated in more detail. For this purpose, the input data stream has been reduced in terms of feature vector entries, so node attributes from a graph perspective, in different combinations and variants. The results of this investigation reveal a clear conclusion, as shown in figure 6.10.

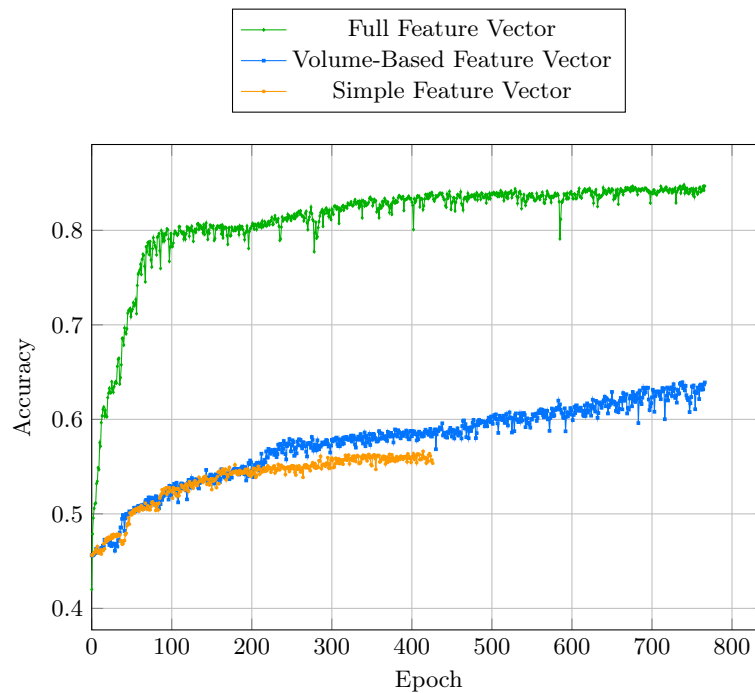


Figure 6.10.: Feature vector studies

Figure 6.10 shows the accuracy trend of the GNN model over the epochs for the three different configurations. The orange plot represents the training using only one entry of the substructure feature vector: the representation or idealization method. The input data for the blue plot additionally includes the substructure volume as an obvious factor for substructure influence. The green plot is generated on the basis of the entire feature

vector. The different ranges of the epoch numbers are occurring due to the early stopping and patience functions of the algorithm, meaning that the training stops if a plateau in accuracy is reached. A conclusion that can be drawn from this study is that the specific feature vector entries play indeed a cardinal role, and that more than a simple volume comparison is necessary for adequate estimations.

The next lever for tuning a model are the so-called hyper-parameters. Hyper-parameters typically include all parameters which are part of the model architecture as for example the number of neurons in layers, the learning rate and dropout ratios. In the HGP-SL algorithm, additional parameters like the graph pooling setting are integrated. This pooling parameter controls the pooling layer which serves the purpose to scale the input size and improve generalization and performance as further described in [265]. To optimize the model in this regard, a set of different hyper-parameter variation runs has been conducted. Despite the feature vector length and the accuracy strictly speaking not being hyper-parameters, they have been added for comprehensibility reasons to the result in figure 6.11.

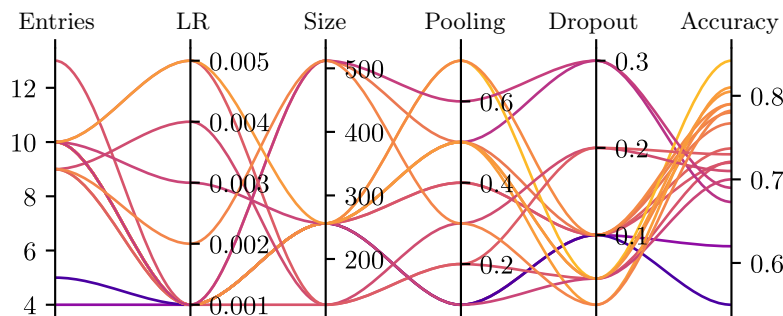


Figure 6.11.: Hyper-parameter study results

The pooling ratio indicates a high influence on the learning capabilities of the used GNN model in these studies. As mentioned previously, the pooling layer can improve generalization capabilities and is likely to support the identification of nodes and node relations in combination with the structure learning mechanism embedded in the HGP-SL algorithm. With regard to the given data structure, this relates to the analysis of relations between substructures, which is in turn matching the engineering way of thoughts.

A higher dropout ratio reveals a disadvantageous trend in the hyper-parameter studies. The dropout layer is commonly introduced as a mean of countermeasure for overfitting problems. Besides this, the used model also contains two other mechanisms to prevent overfitting. The validation set is one of these mechanisms. In the optimum runs, both

training and validation results achieve satisfactory levels even with a lower dropout parameter, which confirms that suitable countermeasures against overfitting are in place. Another integrated countermeasure is the so-called early stopping mechanism. This prevents a model overfit by specifically targeting the validation error and stopping the training process before the validation error starts to rise again [241, pp. 246 sqq.]. In some evaluation runs, this mechanism has stopped the process prior to reaching the maximum number of specified epochs. As a consequence, a higher dropout ratio appears to put too much emphasis on countering overfitting besides the other mechanisms so that necessary and significant information is lost which hinders a successful learning process.

The size of the layers has understandably a noticeable influence on computation time and also on model capabilities. The conducted runs show that a smaller layer size seems to be insufficient to properly handle the data and so the information complexity. On the contrary, the learning rate shows no clear effect on the result.

After conducting these hyper-parameter studies, the result of the optimum run is presented in table 6.3. As provided in this table, all three accuracy values show an accuracy over $\approx 82\%$.

Table 6.3.: Accuracy results

Accuracy		
Train	Validation	Test
0.848	0.820	0.821

Since the accuracy does not provide enough information to evaluate the quality of ML algorithms in some cases, other evaluation metrics have emerged which account for false positives, false negatives, or in other related terms precision and recall. The F-score presented in section 2.6.2 is among these and considered for the evaluation in the given context. This section also presented the different extensions to the F_1 score for multi-class problems: the micro and macro-oriented score. While the micro variant globally accounts for the results, the macro evaluation calculates the metrics for each label and retrieves the unweighted mean [248, 249]. The results of the evaluations using these metrics also confirm a satisfactory model quality as shown in table 6.4.

The confusion matrix is another approach to visualize the results of the classification and is often used for pointing out false positive and false negative candidates. In this representation, the actual and true labels are put into contradiction to the predicted labels. The terms false positives and false negatives are not directly applicable in the

Table 6.4.: F_1 score results

F_1 score	
Micro	Macro
0.844	0.817

present context with the problem being non-binary. Nevertheless, the confusion matrix can be extended to a multi-class domain, which is expected to create a useful qualitative statement. The resulting confusion matrix is shown in figure 6.12 and proves the quality of the model predictions in a visual way. As it can be derived from this figure, a high percentage of each class is labeled correctly by the trained graph neural network.

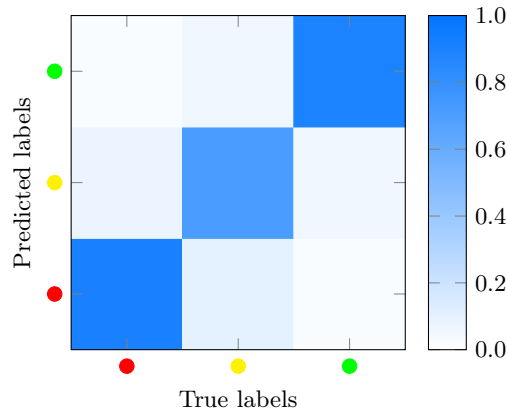


Figure 6.12.: Confusion matrix

The figure 6.13 is shown to create a more detailed overview of the simulation properties and the model quality and contains the individual simulation runs in the dataset plotted using the respective computational time and present number of DOF. The color coding, so Δ , indicates the prediction error of the trained model. Normalizing both axes reveals a correlation between computational time and the involved DOF, which can be explained by their logical dependency in terms of computational complexity. However, different types of element relations and formulations, as well as external influence factors, can affect the computational time and result in the seen variance. As a conclusion from this figure, the algorithm is able to predict the model quality to a satisfactory extent throughout the entire design space with only minor deviations in some regions.

Assessing and classifying the quality of the trained GNN, or basically a trained NN, and its results is not a straightforward task in general. A significant contributor to this problem

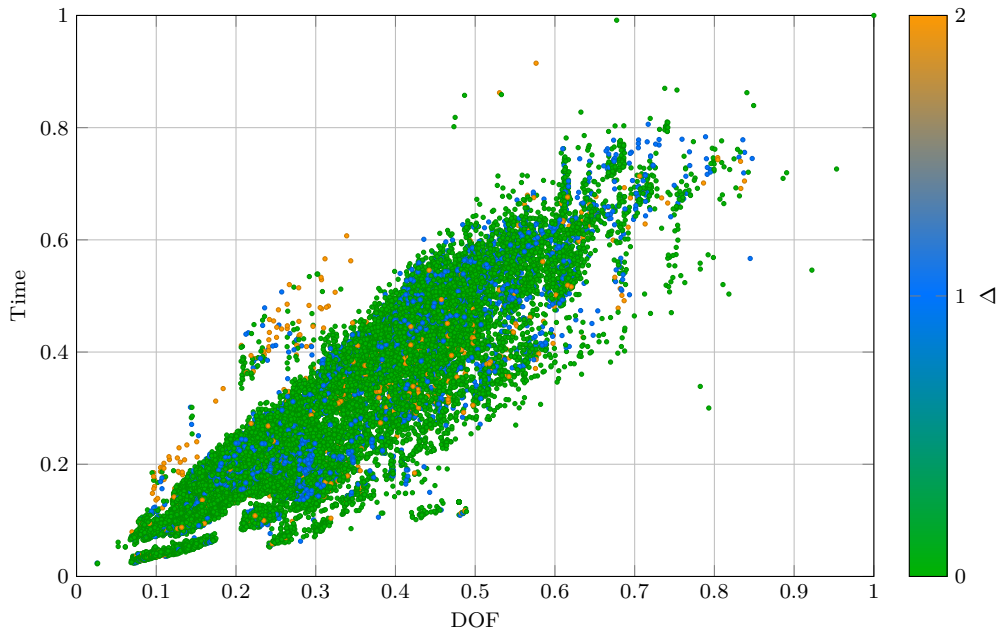


Figure 6.13.: Dataset evaluation

is that an adequate reference is often not available, especially in more complex scenarios. Despite the model achieving a comparably high accuracy of $>80\%$, the relation to assess if this accuracy level is sufficient is still missing. Other benchmarks of common GNN applications, as for example done by Zhang et al. in [265], target use cases like enzyme or protein classification on a graph data basis and range between approximately 69% to 85% in their accuracy. Consequently, the achieved accuracy is within a satisfactory range in this context.

In many cases, the so-called Human-Level Performance (HLP) [241, p. 535] is considered and used to put the result into context. The HLP is expected to achieve accuracy levels of 95% or more in for example classical and standard image recognition tasks due to the remarkable recognition abilities of the human. Nevertheless, the HLP can be assumed to be lower in more complex engineering tasks, which include highly correlated phenomena and complex physical characteristics. Regarding the problem tasks within this work, it is complicated to define a level for the HLP. Estimating the influence of substructures on complex assemblies depends on many aspects, thus depicts a difficult objective which is also hard to measure. An engineer could be able to assess the objective value for a known model, but predicting the correct label to a plethora of unknown simulation models is expected to be rather difficult. Nonetheless, the aim of this part of the research has been to showcase a conceptual approach to cope with the long-term memory in the engineering

domain. The developed model could be used as recommendation system and as first rough assessment of modeling decisions with the final decision being made by the engineer or the engineering team itself, and by this significantly support the design process. In lights of these considerations, the achieved accuracy is depicting a satisfactory result, especially for the first estimate of substructure influence.

7. Conclusion

7.1. Summary

For the conclusion of this work, a wrap-up is done first, which starts with moving back to the objectives stated in the introduction. Due to common obstacles and hurdles on the path towards more efficient processes, and by this towards a product development leading to better competitiveness and more sophisticated products, the advantages of automation also for engineering domains have been pointed out.

The scope of this work has been aligned towards this aspect by focusing on the goal of automating the generation of a simulation model from a present geometric model or system. The simulation model in this context is referring to a FEM model for structural analyses with the FE approach being the current state-of-the-art in the aero engine domain. Along with the complexity of a system comes the demand or even requirement to introduce simplification assumptions to remain within reasonable process and manageability boundaries and to provide capabilities to respond fast enough. Consequently, idealization plays an important role in the development process, and therefore has been defined as part of the goal for the automated process. To maintain model integrity, boundary conditions which are present in the geometric system have to be integrated in the simulation model as well.

Taking a more general perspective onto these goals has led to the fundamental challenges in the design process involving visual aspects, short-term and long-term memory which guide the involved manual tasks. To cope with these challenges, a strategy has been developed which serves as a common thread to the described work and represents the fundamentals for the built framework. This strategy comprises a set of approaches digitally imitating the manual model analysis and interpretation which then guide the manual model transfer, setup and assembly steps.

The major system analysis, often led by a visual inspection, results in identified components and conditions and defines the next steps in the process. This aspect is covered by the developed recognition framework, which combines different aspects and logics to identify boundary conditions and categorize components. The focus of this

framework is set on the entities whose recognition entails most benefits for structural simulations in preliminary design phases. The information about the system gathered by the recognition framework is collected in a database which serves as knowledge base for the subsequent processes, therefore contributes to the digital version of the short-term memory aspect.

Afterwards, the next steps in the developed strategy are dedicated to the transfer of geometric entities to suitable FEM representations. The simplification criterion is part of these steps, and its focus has been set on identified component categories in order to avoid excessive effort in comparison to the benefits and a negative influence on process robustness due to unreasonably forced genericness. For this reason, the focus has been set on these categories which imply a sufficient potential for simplification due to their common topology and allow specifically adapted and optimized workflows. Similar to the recognition framework, the transfer strongly depends on visual analysis, which is why this thought is also guiding the transfer process. Thin-walled casing structures imply such a potential for reducing complexity by means of shell elements. To be able to better access and work with a model, segmentation methods have been developed and presented which are able to split a component into accessible substructures. These can be analyzed separately, and, based on a set of implemented logics and values as the feature vector, decisions regarding their simplification and representation in the final FEM model are made. Similarly, a scanning process is integrated in the framework for beam-like structures. The developed and automated methods for FEM model generation can directly access the information derived from the geometric analysis, which allows well-founded property setup. Final evaluations show that the automatically transferred models can significantly decrease complexity while remaining within a reasonable model quality tolerance.

After all process chains for the components have finished, an automated assembly process is connecting the information from the single components and from the recognition database about interfaces and boundary conditions. Based on this, the framework automatically assembles the system and introduces appropriate simulation conditions. The application section in this context shows that the process can still take long for the execution of all processes, especially in complex scenarios. Setting that into contrast to the manual process which is expected to take up to months or even years [9, 10], however, emphasizes the use and benefits of the developed automated approach.

Nevertheless, increasing process efficiency is not the only advantage that is generated by automating the process. In the last chapter, a conceptual approach has been developed and presented to show the further use of automation for a wide range of newly emerging, data-

driven methods. Based on parametric feature-based models, different sets of geometry models have been created in an automated workflow. This geometry pool has been processed by the developed framework within an efficient process architecture to create simulation models with various level of detail. A data pipeline has been introduced which allows managing a structured data flow and access for multiple purposes. The graph neural network technique has been selected as a method to imitate the long-term memory aspect by learning from this generated dataset about the influence of simplifications or design in general on model quality. The data pipeline therefore creates a data pool containing the model information converted to graph representations, which are assigned with a criterion describing the achieved model quality. The results of the trained model are promising and achieve a relatively high accuracy ($\approx 82\%$) which has to be assessed in relation to the human-level performance. This emphasizes the use case and prospects for future research and applications.

In conclusion, this thesis has presented a strategy and developed methods and approaches to imitate and replicate the main common aspects of the design process which are visual analysis, manual steps and engineering expertise. These methods can significantly reduce the manual tasks, increase process efficiency and performance while also providing the capabilities to build a comprehensive knowledge base, which can be beneficial for future design iterations in the form of a, for example, recommendation system. Moreover, the process is introducing a multitude of interfaces at several process steps for extracting information and allows combining geometric to simulation entities, thus to simulation results, or modularizing specific simulation conditions and steps.

7.2. Outlook

Extending the scope of the developed framework would imply more automation potential and more interfaces for the entire development process. In lights of these considerations, the recognition framework has been set up in a modular way so that additional recognition categories and entities can be integrated easily. For example, installation and auxiliary components as oil pipes represent further potential entities of interest for the development. The CAE transfer of these structures could also be outsourced to a specifically adapted workflow, which introduces optimized adaption and modification processes. This idea has been investigated in the scope of this work what lead to topics like curve or point cloud discretization, analysis and skeleton retrieval, so also to research works as [293–301]. This also shows the usability of the developed approach and its created interfaces for interdisciplinary and multi-objective collaboration and integration.

Regarding an artificial system understanding aspect, the identification of the rotary systems in the aero engine could be the starting point for supplemental higher-level recognition tasks. The identified rotors imply information about compressor and turbine sections. Point cloud sampling and projection could reveal these two rotor sections, which could be separated by identifying both of the resulting clusters. Based on the rotor and the identified segment type, so for example the turbine section of the high-pressure rotor, temperature and pressure fields could be interpolated in reference to the associated rotor discs. This would depict an additional step towards system understanding and recognition while providing further-reaching boundary condition information which could be transferred to the opposing casings.

Methods from the field of computer science, graphics or similar which pursue the intention to identify and subdivide shapes, or specifically 3D models, as described in section 2.5 (for further reference see [62, 65–72, 74–76, 302–319]) share a similarity to the visual analysis of a CAD model done by an engineer. However, available methods provided by [67, 68, 73, 305] have been tested for this purpose in the course of this research and proven as rather unsuitable in terms of general configurations or have not led to the satisfactory results. Despite this, the general idea implied in this methodology is connected to a plethora of use cases within the context of this work, and therefore implies attractive research fields for future works.

Introducing simplification and abstraction processes and maintaining a model quality which is accurate enough for the intended purposes can become a task which is difficult to accomplish, also for experienced and skilled analysts. In many cases, trimming and tuning a model is necessary to improve the achieved quality and to match the desired model characteristics. The developed automated process provides and creates many interfaces to simulation properties which could be directly used for the so-called model updating. Hardenberg et al. [283] for example presents an approach to model updating and tuning using FE parameters which could be provided by the framework presented in this thesis in an automated way.

Additionally, the field of knowledge base generation and learning can be identified as a region of interest for further research. The graph-related learning approach presented in this work can be extended for example. Apart from the node attributes which have been assigned with the feature vector, some graph methods can also account for edge attributes which could incorporate information about mutual distances, interface types or general data about the present boundary conditions as for example flange properties. Moreover, expanding the data pool for the model training is a potential improvement to almost all data-driven technologies, so also in this context. A larger model set containing a higher

7. Conclusion

variety of models will certainly influence the meaningfulness of the drawn conclusion in a positive way. A larger model set, however, would also recommend a high-performance computing environment, like cloud computing, for which the developed method would already be suitable. Regarding the evaluation of the accuracy itself, a confidence criteria would be advantageous to assess the prediction in a more production-related context. The predicted values within the last layer of the neural network which depict the input to the softmax classification could be investigated to build such a confidence value.

The presented strategy in combination with this suggested set of next steps has the potential to be of very great value for data-driven, innovative and efficient future design processes.

A. Appendix

A.1. Algorithms

Algorithm A.1 Body pairs analysis

Require: Set of assembly bodies \mathcal{B} , Distance tolerance ϵ

Ensure: Set of body pairs \mathcal{P}_B

```
1:  $\mathcal{P}_B \leftarrow \{\}$ 
2: for  $B_i, B_j \in \mathcal{B}$  with  $i \neq j$  do
3:   if BBox.Distance( $B_i, B_j$ )  $> \epsilon$  then
4:     continue
5:   if CBox.Distance( $B_i, B_j$ )  $> \epsilon$  then
6:     continue
7:   if Distance( $B_i, B_j$ )  $> \epsilon$  then
8:     continue
9:    $\mathcal{P}_B \leftarrow \mathcal{P}_B \cup \{(B_i, B_j)\}$ 
10: return  $\mathcal{P}_B$ 
```

Algorithm A.2 Rear face crawler

Require: Flange face F_F , Body B with set of faces \mathcal{F}_B , Maximum thickness τ

Ensure: Set of flange rear faces \mathcal{F}_R

```

1:  $\mathcal{F}_R \leftarrow \{\}$ 
2: for  $F \in \mathcal{F}_B$  with  $F \neq F_F$  do
3:   if  $\text{Type}(F) \neq \text{planar}$  or
4:      $\text{Normal}(F) \not\parallel \text{Normal}(F_F)$  or
5:      $\text{DistanceFiltering}(F, F_F) > \tau$  then
6:       continue
7:    $P_1, P_2 \leftarrow$  Points of closest distance
8:   if  $\overrightarrow{P_1 P_2} \not\parallel \text{Normal}(F_F)$  then
9:     continue
10:   $\mathcal{H} \leftarrow \text{RayTracing}(P_1, \overrightarrow{P_1 P_2}, B)$ 
11:  if  $|\mathcal{H}| > 2$  then
12:    continue
13:   $\mathcal{F}_R \leftarrow \mathcal{F}_R \cup \{F\}$ 
14: return  $\mathcal{F}_R$ 

```

Algorithm A.3 Casing categorization

Require: Bodies \mathcal{B} , Ratio parameter γ , Beam structures \mathcal{B}_B , Component axis \mathbf{v}_{comp}

Ensure: Set of casing components \mathcal{B}_C

```

1:  $\mathcal{B} \leftarrow \mathcal{B} - \mathcal{B}_B$ 
2:  $\mathcal{B}_C \leftarrow \{\}$ 
3: for  $B \in \mathcal{B}$  do
4:    $x, r, \phi \leftarrow \text{CBox}(B)$ 
5:   if  $\phi \leq 2\pi$  then
6:      $\mathcal{B}_P \leftarrow \{B_i \mid B_i \hat{=} \text{CounterPart}(B) \cup \text{FlangeBetween}(B, B_i) \forall B_i \in \mathcal{B}\}$ 
7:     if  $|\mathcal{B}_P| = \{\}$  then
8:       continue
9:      $A_O = 0$ 
10:    for  $F \in \text{Faces}(B)$  do
11:      if  $\text{Type}(F) \in \{\text{planar}, \text{cylindrical}, \text{conical}, \text{revolved}\}$  and
12:         $\text{Axis}(F) \parallel \mathbf{v}_{comp}$  then
13:           $A_O += \text{Area}(F)$ 
14:      if  $A_O \geq \gamma * \text{Area}(B)$  then
15:         $\mathcal{B}_C \leftarrow \mathcal{B}_C \cup \{B\}$ 
16: return  $\mathcal{B}_C$ 

```

Algorithm A.4 Silhouette filtering

Require: Faces \mathcal{F} of Body, Center axis \mathbf{v}

Ensure: Set of silhouettes \mathcal{S}

```

1:  $\mathcal{S} \leftarrow \{\}$ 
2: for all  $f \in \mathcal{F}$  do
3:   if  $\text{Type}(f) \in \{\text{cylindrical, conical, revolved, swept}\}$  then
4:     if  $\text{IsCollinear}(\text{Axis}(f), \mathbf{v})$  then
5:        $\mathcal{S} \leftarrow \mathcal{S} \cup \text{Projection}(f)$ 
6:     else if  $\text{Type}(f)$  is Planar then
7:       if  $\text{Axis}(f) \parallel \mathbf{v}$  then
8:          $c \leftarrow \text{CBox}(f)$ 
9:          $\mathcal{S} \leftarrow \mathcal{S} \cup \text{Projection}(c)$ 
10: return  $\mathcal{S}$ 

```

Algorithm A.5 Contour extension

Require: Voronoi vertices \mathcal{P}_V , Graph G , Set of contour splines \mathcal{S}

Ensure: Set of contour extensions \mathcal{L}_{ext}

```

1:  $T_V \leftarrow \text{KDTree}(\mathcal{P}_V)$ 
2:  $T_G \leftarrow \text{KDTree}(\text{Nodes}(G))$ 
3:  $\mathcal{L}_{ext} \leftarrow \{\}$ 
4: for all  $p \in \text{LeafNodes}(G)$  do
5:    $p_{start} \leftarrow p$ 
6:    $\mathcal{P}_{visited} \leftarrow \{p_{start}\}$ 
7:    $p_V \leftarrow T_V.\text{Closest}(p_{start})$ 
8:    $p_i \leftarrow T_G.\text{Closest}(p_V)$ 
9:   if  $\nexists G.\text{Path}(p_{start}, p_i)$  then
10:    continue
11:    $\mathcal{P}_{visited} \leftarrow \mathcal{P}_{visited} \cup G.\text{Path}(p_{start}, p_i)$ 
12:    $p_{start} \leftarrow p_i$ 
13:    $\mathcal{P}_n \leftarrow G.\text{Neighbors}(p_{start}) - \mathcal{P}_{visited}$ 
14:    $p_{next} \leftarrow \mathcal{P}_n(0)$ 
15:   while  $(p_{start}, p_{next}) \in \mathcal{S}$  do
16:      $\mathcal{P}_{visited} \leftarrow \mathcal{P}_{visited} \cup \{p_{next}\}$ 
17:      $p_{start} \leftarrow p_{next}$ 
18:      $\mathcal{P}_n \leftarrow G.\text{Neighbors}(p_{start}) - \mathcal{P}_{visited}$ 
19:      $p_{next} \leftarrow \mathcal{P}_n(0)$ 
20:    $\mathcal{L}_{ext} \leftarrow \mathcal{L}_{ext} \cup \{(p_{next}, p_{start})\}$ 
21: return  $\mathcal{L}_{ext}$ 

```

Algorithm A.6 Polygon adaption

Require: Boundary curves \mathcal{C}_B , Points $\mathcal{P}_{P,i}$ of path i

Ensure: Updated points $\mathcal{P}_{P,i}$ for path i

```

1:  $A \leftarrow Dictionary()$ 
2: for all  $p \in \mathcal{P}_{P,i}$  do
3:    $\mathcal{C}_p \leftarrow ClosestLines(p, \mathcal{C}_B)$ 
4:    $A(p) \leftarrow \mathcal{C}_p$ 
5:  $\mathcal{O} \leftarrow CountOccurrencesOfCurves(A, \mathcal{C}_B)$ 
6:  $\mathcal{A} \leftarrow FilterMostCommonCurves(A, \mathcal{O})$ 
7:  $\mathcal{A}_r \leftarrow \{\}$ 
8: for all  $c \in \mathcal{O}$  do
9:    $\mathcal{P}_c \leftarrow PointsAssociatedWith(c, \mathcal{A})$ 
10:   $\mathcal{S} \leftarrow Segments(\mathcal{P}_c)$ 
11:  for all  $\mathcal{S}_j \subset \mathcal{S}$  do
12:    if  $|\mathcal{S}_j| > 1$  then
13:       $\mathcal{A}_r \leftarrow \mathcal{A}_r \cup \{(c, \mathcal{S}_j)\}$ 
14: for all  $(c, \mathcal{S}) \in \mathcal{A}_r$  do
15:    $p_{min} \leftarrow p : \min_{p \in \mathcal{S}} r(p)$ 
16:    $l \leftarrow LineFromPointAndVector(p_{min}, c)$ 
17:    $\mathcal{P}_{proj} \leftarrow ProjectPointsToLine(\mathcal{S}, l)$ 
18:    $\mathcal{P}_{P,i} \leftarrow Replace(\mathcal{P}_{proj})$ 
19: return  $\mathcal{P}_P$ 

```

Bibliography

- [1] Roger Atkinson. “Project management: Cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria”. In: *International Journal of Project Management* 17.6 (1999), pp. 337–342. ISSN: 02637863. DOI: 10.1016/S0263-7863(98)00069-6.
- [2] David Ullman. *The Mechanical Design Process*. 4. Edition. McGraw-Hill, 2018, p. 415. ISBN: 9780999357804.
- [3] Thomas Bauernhansl, ten Hompel Michael, and Vogel-Heuser Birgit. *Industrie 4.0 in Produktion. Automatisierung und Logistik. Anwendung - Technologien - Migration*. 1st ed. Wiesbaden: Springer Vieweg, 2014, pp. X, 648. ISBN: 978-3-658-04681-1. DOI: 10.1007/978-3-658-04682-8.
- [4] S. Vajna et al. *CAX für Ingenieure - Eine praxisbezogene Einführung*. Springer-Verlag Berlin Heidelberg, 2018, p. 655. ISBN: 978-3-662-54623-9. DOI: 10.1007/978-3-662-54624-6.
- [5] Bernd Klein. *FEM Grundlagen und Anwendungen der Finite-Element-Methode im Maschinen- und Fahrzeugbau*. 2015, p. 391. ISBN: 978-3-658-06054-1. DOI: 10.1007/978-3-658-06054-1.
- [6] Abdelkader Benaouali and Stanisław Kachel. “An automated CAD/CAE integration system for the parametric design of aircraft wing structures”. In: *Article in Journal of Theoretical and Applied Mechanics* 55 (2017), pp. 447–459. DOI: 10.15632/jtam-pl.55.2.447. URL: <https://www.researchgate.net/publication/316717270>.
- [7] Okereke Michael and Simeon Keates. *Finite Element Applications - A Practical Guide to the FEM Process*. Springer International Publishing, 2018. ISBN: 978-3-319-67124-6. DOI: 10.1007/978-3-319-67125-3.
- [8] Trevor T. Robinson, Cecil G. Armstrong, and R. Fairey. “Automated mixed dimensional modelling from 2D and 3D CAD models”. In: *Finite Elements in Analysis and Design* 47.2 (2011), pp. 151–165. ISSN: 0168874X. DOI: 10.1016/j.finel.2010.08.010. URL: <http://dx.doi.org/10.1016/j.finel.2010.08.010>.

- [9] Cecil G. Armstrong, Trevor T. Robinson, and Hengan Ou. *Recent Advances in CAD/CAE Technologies for Thin-Walled Structures Design and Analysis*. Tech. rep. 2008.
- [10] Flavien Boussuge et al. “Capturing simulation intent in an ontology: CAD and CAE integration application”. In: *Journal of Engineering Design* 30.10-12 (2019), pp. 688–725. ISSN: 14661837. DOI: 10.1080/09544828.2019.1630806. URL: <https://doi.org/10.1080/09544828.2019.1630806>.
- [11] Liang Sun et al. “Decomposing complex thin-walled CAD models for hexahedral-dominant meshing”. In: *CAD Computer Aided Design* 103 (2018), pp. 118–131. ISSN: 00104485. DOI: 10.1016/j.cad.2017.11.004.
- [12] Okba Hamri et al. “Software environment for CAD/CAE integration”. In: *Advances in Engineering Software* 41.10-11 (2010), pp. 1211–1222. ISSN: 09659978. DOI: 10.1016/j.advengsoft.2010.07.003. URL: <http://dx.doi.org/10.1016/j.advengsoft.2010.07.003>.
- [13] Willy J.G. Bräunling. *Flugzeugtriebwerke*. 4th ed. Heidelberg: Springer Vieweg, Berlin, Heidelberg, 2015, pp. LXXXVIII, 2020. ISBN: 978-3-642-34538-8. DOI: <https://doi.org/10.1007/978-3-642-34539-5>.
- [14] Gerhard Pahl et al. *Engineering Design: A Systematic Approach*. Springer, London, 2007. ISBN: 978-1-84628-318-5. DOI: 10.1007/978-1-84628-319-2.
- [15] Jack D. Mattingly. *Elements of Gas Turbine Propulsion*. Vol. 2005. 1996, pp. 43–78. ISBN: 0-07-912196-9.
- [16] Jack D. Mattingly et al. *Aircraft Engine Design*. American Institute of Aeronautics and Astronautics, Incorporated, 2019, 2019, p. 834. ISBN: 978-1-62410-517-3.
- [17] Edris Safavi et al. “Collaborative multidisciplinary design optimization: A framework applied on aircraft conceptual system design”. In: *Concurrent Engineering Research and Applications* 23.3 (2015), pp. 236–249. ISSN: 15312003. DOI: 10.1177/1063293X15587020.
- [18] Ilan Kroo and Ilan Kroo. “Multidisciplinary optimization applications in preliminary design - Status and directions”. In: (1997). DOI: 10.2514/6.1997-1408.
- [19] Gerhard Pahl. *Konstruieren mit 3D-CAD-Systemen: Grundlagen, Arbeitstechnik, Anwendungen*. 1990, 1 Online-Ressource (X, 270 S. 129 Abb). ISBN: 978-3-642-47593-1.

- [20] José Encarnação, Wolfgang Straßer, and Reinhard Klein. *Graphische Datenverarbeitung*. De Gruyter Oldenbourg, 2019, p. 427. ISBN: 978-3-486-78924-9.
- [21] VDI-Fachbereich Produktentwicklung und Mechatronik. *VDI 2209 - 3-D-Produktmodellierung - Technische und organisatorische Voraussetzungen - Verfahren, Werkzeuge und Anwendungen - Wirtschaftlicher Einsatz in der Praxis*. Tech. rep. VDI-Gesellschaft Produkt- und Prozessgestaltung, 2009, p. 167.
- [22] Ian Stroud. *Boundary Representation Modelling Techniques*. Springer, London, 2006. ISBN: 978-1-84628-312-3. DOI: 10.1007/978-1-84628-616-2.
- [23] I. C. Braid. “Designing with volumes”. Ph.D. Dissertation. University of Cambridge, 1973, p. 181. DOI: 10.1016/0010-4485(75)90098-6.
- [24] Flavien Boussuge. “Idealization of CAD Assemblies for FE Structural Analyses”. PhD thesis. 2014. URL: <https://tel.archives-ouvertes.fr/tel-01071560>.
- [25] C. M. Hoffmann. *Geometric and Solid Modeling: An Introduction*. Morgan Kaufmann series in computer graphics and geometric modeling. Morgan Kaufmann, 1989. ISBN: 9781558600676.
- [26] M. Mantyla and M. Mäntylä. *An Introduction to Solid Modeling*. Principles of computer science series. Computer Science Press, 1988. ISBN: 9780881751086.
- [27] Tamer M.M. Shahin. “Feature-based design - An overview”. In: *Computer-Aided Design and Applications* 5.5 (2008), pp. 639–653. ISSN: 16864360. DOI: 10.3722/cadaps.2008.639-653.
- [28] J. J. Shah. “Assessment of features technology”. In: *Computer-Aided Design* 23.5 (1991), pp. 331–343. ISSN: 0010-4485. DOI: [https://doi.org/10.1016/0010-4485\(91\)90027-T](https://doi.org/10.1016/0010-4485(91)90027-T). URL: <https://www.sciencedirect.com/science/article/pii/001044859190027T>.
- [29] Jami J. Shah et al. “A discourse on geometric feature recognition from cad models”. In: *Journal of Computing and Information Science in Engineering* 1.1 (2001), pp. 41–51. ISSN: 15309827. DOI: 10.1115/1.1345522.
- [30] K. J. Bathe. *Finite Element Procedures*. 2014, p. 1037. ISBN: 0133014584. URL: <http://www.amazon.com/Finite-Element-Procedures-Part-1-2/dp/0133014584>.
- [31] S. Timoshenko. *History of Strength of Materials: With a Brief Account of the History of Theory of Elasticity and Theory of Structures*. Dover Civil and Mechanical Engineering Series. Dover Publications, 1983, p. 452. ISBN: 9780486611877.

- [32] Rao Garimella and Mark S. Shephard. “Tetrahedral Mesh Generation with Multiple Elements Through the Thickness”. In: *Proc. 4th International Meshing Roundtable* May 2013 (1995), pp. 321–334.
- [33] Cornelia Thieme. “How to Create a Good Quality FE Model”. In: *Msc Software 3* (2016). URL: https://www.mscsoftware.com/sites/default/files/whitepaper_how_to_create_a_good_fe_model.pdf.
- [34] Siemens NX. *Siemens Documentation - NX Help*. 2021. URL: https://docs.plm.automation.siemens.com/tdoc/nx/1899/nx_help/#uid:index (visited on 12/12/2021).
- [35] Siemens PLM Software Inc. “NX Nastran User’s Guide”. In: (2014), p. 886.
- [36] Siemens PLM. “NX Nastran 12 - Advanced Nonlinear Theory and Modeling Guide”. In: (2017), pp. 258–266. URL: <https://docs.plm.automation.siemens.com/tdoc/nxnastran/12/help/#uid:index>.
- [37] Robert Gasch, Klaus Knothe, and Robert Liebich. *Strukturdynamik - Diskrete Systeme und Kontinua*. Springer-Verlag Berlin Heidelberg, 2012, pp. XXVI, 670. ISBN: 978-3-540-88977-9. DOI: 10.1007/978-3-540-88977-9.
- [38] Jörg Wagner. *Einführung in die Finite Elemente Methode*. 2020.
- [39] Mats G. Larson and Fredrik Bengzon. *The Finite Element Method: Theory, Implementation, and Applications*. Springer-Verlag Berlin Heidelberg, 2013, pp. 1–726. ISBN: 978-3-642-33286-9. DOI: 10.1007/978-3-642-33287-6.
- [40] O. C. Zienkiewicz. “The Finite Element Method: Its Basis and Fundamentals”. In: *The Finite Element Method: its Basis and Fundamentals* (2013), p. iii. DOI: 10.1016/b978-1-85617-633-0.00020-4.
- [41] ASME. *V&V 10-2019 - Standard for Verification and Validation in Computational Solid Mechanics*. ASME, 2020, p. 44. ISBN: 9780791873168. URL: <https://www.asme.org/codes-standards/find-codes-standards/v-v-10-guide-verification-validation-computational-solid-mechanics#ASME-digital-books>.
- [42] David Ewins. *Modal Testing: Theory, Practice and Application*. Engineering dynamics series. Wiley, 2000. ISBN: 9780863802188.
- [43] Daniel C. Kammer, Joseph Cessna, and Andrew Kostuch. “An effective mass measure for selecting free-free target modes”. In: *Conference Proceedings of the Society for Experimental Mechanics Series* January (2005). ISSN: 21915644.

-
- [44] Daniel C. Kammer, Joseph Cessna, and Andrew Kostuch. “A generalization of effective mass for selecting free-free target modes”. In: *Journal of Vibration and Acoustics, Transactions of the ASME* 129.1 (2007), pp. 121–127. ISSN: 10489002. DOI: 10.1115/1.2424980.
- [45] R. J. Allemang, University of Cincinnati. Department of Mechanical, and Industrial Engineering. *Investigation of Some Multiple Input/Output Frequency Response Function Experimental Modal Analysis Techniques*. University Microfilms, 1980.
- [46] R. J. Allemang and D. Brown. “A Correlation Coefficient for Modal Vector Analysis”. In: *Proceedings of the 1st International Modal Analysis Conference*. Orlando, 1982, pp. 110–116.
- [47] Randall J. Allemang. “The modal assurance criterion - Twenty years of use and abuse”. In: *Sound and Vibration* 37.8 (2003), pp. 14–21. ISSN: 15410161.
- [48] D. C. Kammer and M. J. Triller. “Selection of component modes for craig-bampton substructure representations”. In: *Journal of Vibration and Acoustics, Transactions of the ASME* 118.2 (1996), pp. 264–270. ISSN: 15288927. DOI: 10.1115/1.2889657.
- [49] Alain Girard and Nicolas A. Roy. “Modal effective parameters in structural dynamics”. In: *Revue Europeenne des Elements* 6.2 (1997), pp. 233–254. ISSN: 12506559. DOI: 10.1080/12506559.1997.10511267.
- [50] Michelle Nieto, Mostafa Elsayed, and Denis Walch. “Modal Participation Factors And Their Potential Applications In Aerospace: A Review”. In: (2018), pp. 1–6. DOI: 10.25071/10315/35254.
- [51] M. Aenlle, Martin Juul, and R. Brincker. “Modal Mass and Length of Mode Shapes in Structural Dynamics”. In: *Shock and Vibration* 2020 (2020). ISSN: 10709622. DOI: 10.1155/2020/8648769.
- [52] Jacob Wijker. “Modal Effective Mass”. In: *Mechanical Vibrations in Spacecraft Design*. Springer Science & Business Media, 2008. Chap. 16, pp. 247–263. ISBN: 978-3-540-75552-4. DOI: 10.1007/978-3-540-75553-1_16.
- [53] Tom Irvine. *Effective modal mass & modal participation factors*. 2015. URL: <http://www.vibrationdata.com/tutorials2/ModalMass.pdf> (visited on 07/17/2021).
- [54] Byung Chul Kim and Duhwan Mun. “Feature-Based simplification of boundary representation models using sequential iterative volume decomposition”. In: *Computers and Graphics (Pergamon)* 38.1 (2014), pp. 97–107. ISSN: 00978493. DOI: 10.1016/j.cag.2013.10.031.

- [55] Atul Thakur, Ashis Gopal Banerjee, and Satyandra K. Gupta. *A survey of CAD model simplification techniques for physics-based simulation applications*. 2009. DOI: 10.1016/j.cad.2008.11.009.
- [56] Hugues Hoppe et al. “Mesh Optimization”. In: *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’93. New York, NY, USA: Association for Computing Machinery, 1993, pp. 19–26. ISBN: 0897916018. DOI: 10.1145/166117.166119. URL: <https://doi.org/10.1145/166117.166119>.
- [57] Hugues Hoppe. “Progressive Meshes”. In: *Computational Geometry* (1996), pp. 99–108.
- [58] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. “Decimation of triangle meshes”. In: June (1992), pp. 65–70. DOI: 10.1145/133994.134010.
- [59] Marc Soucy and Denis Laurendeau. “Multiresolution surface modeling based on hierarchical triangulation”. In: *Computer Vision and Image Understanding* 63.1 (1996), pp. 1–14. ISSN: 10773142. DOI: 10.1006/cviu.1996.0001.
- [60] Michael Garland and Paul S. Heckbert. “Surface simplification using quadric error metrics”. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997* (1997), pp. 209–216. DOI: 10.1145/258734.258849.
- [61] Shuming Gao et al. “Feature suppression based CAD mesh model simplification”. In: *CAD Computer Aided Design* 42.12 (2010), pp. 1178–1188. ISSN: 00104485. DOI: 10.1016/j.cad.2010.05.010.
- [62] Dong Xiao et al. “CAD mesh model segmentation by clustering”. In: *Computers and Graphics (Pergamon)*. Vol. 35. 3. June 2011, pp. 685–691. DOI: 10.1016/j.cag.2011.03.020.
- [63] Hong Xiao et al. “CAD mesh model simplification with assembly features preservation”. In: *Science China Information Sciences* 57.3 (2014), pp. 1–11. ISSN: 1674733X. DOI: 10.1007/s11432-013-4791-z.
- [64] William Roshan Quadros and Steven J. Owen. “Defeaturing CAD models using a geometry-based size field and facet-based reduction operators”. In: *Engineering with Computers*. Vol. 28. 3. 2012, pp. 211–224. DOI: 10.1007/s00366-011-0252-8.
- [65] Alexander Agathos et al. “3D Mesh Segmentation Methodologies for CAD applications”. In: *Computer-Aided Design and Applications* 4.6 (2007), pp. 827–841. ISSN: 16864360. DOI: 10.1080/16864360.2007.10738515.

-
- [66] Eman Ahmed et al. “A survey on Deep Learning Advances on Different 3D Data Representations”. In: vol. 1. 1. Article, 2019. arXiv: 1808.01462v2.
- [67] Rana Hanocka and Shachar Fleishman. “MeshCNN: A Network with an Edge”. In: (2019). DOI: 10.1145/nnnnnnn. arXiv: 1809.05910v2. URL: <https://doi.org/10.1145/nnnnnnn>.
- [68] Pengyu Wang et al. “3D shape segmentation via shape fully convolutional networks”. In: *Computers and Graphics (Pergamon)* 70 (Feb. 2018), pp. 128–139. ISSN: 00978493. DOI: 10.1016/j.cag.2017.07.030.
- [69] Lijun Chen and Nicolas D. Georganas. “An efficient and robust algorithm for 3D mesh segmentation”. In: *Multimedia Tools and Applications* 29.2 (June 2006), pp. 109–125. ISSN: 13807501. DOI: 10.1007/s11042-006-0002-x.
- [70] Yu Kun Lai et al. “Feature sensitive mesh segmentation”. In: *Proceedings SPM 2006 - ACM Symposium on Solid and Physical Modeling*. Vol. 2006. Association for Computing Machinery (ACM), July 2006, pp. 17–26. ISBN: 1595933581. DOI: 10.1145/1128888.1128891.
- [71] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. “Consistent mesh partitioning and skeletonisation using the shape diameter function”. In: *Visual Computer* 24.4 (2008), pp. 249–259. ISSN: 01782789. DOI: 10.1007/s00371-007-0197-5.
- [72] P. Simari et al. *Multi-objective shape segmentation and labeling*. Tech. rep. 5. 2009.
- [73] Li Hao, Rong Mo, and Binbin Wei. “CAD model segmentation algorithm using the fusion of PERT and spectral technology”. In: *Mathematical Problems in Engineering* 2019 (2019). ISSN: 15635147. DOI: 10.1155/2019/4395972.
- [74] Lakshmi Priya Muraleedharan, Shyam Sundar Kannan, and Ramanathan Muthuganapathy. “Autoencoder-based part clustering for part-in-whole retrieval of CAD models”. In: *Computers and Graphics (Pergamon)* 81 (June 2019), pp. 41–51. ISSN: 00978493. DOI: 10.1016/j.cag.2019.03.016.
- [75] Cheuk Yiu Ip and William C. Regli. *Content-Based Classification of CAD Models with Supervised Learning*. Tech. rep. 5. 2005, pp. 609–617. URL: <http://www.designrepository.org/>.
- [76] Fei Wei Qin et al. “A deep learning approach to the classification of 3D CAD models”. In: *Journal of Zhejiang University: Science C* 15.2 (2014), pp. 91–106. ISSN: 1869196X. DOI: 10.1631/jzus.C1300185.

- [77] Anton V. Mobley, Michael P. Carroll, and Scott A. Canann. “An Object Oriented Approach to Geometry Defeaturing for Finite Element Meshing”. In: *7th International Meshing Round Table, '98*. May 2000. 1998, pp. 547–563.
- [78] Milind Venkataraman, Sashikumar Sohoni and Rahul Rajadhyaksha. “Removal of blends from Boundary Representation models”. In: *Proceedings of the Symposium on Solid Modeling and Applications* (2002), pp. 83–94. DOI: 10.1145/566294.566297.
- [79] H. Zhu and C. H. Menq. “B-Rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition”. In: *CAD Computer Aided Design* 34.2 (2002), pp. 109–123. ISSN: 00104485. DOI: 10.1016/S0010-4485(01)00056-2.
- [80] Bojan Babic, Nenad Nesic, and Zoran Miljkovic. “A review of automated feature recognition with rule-based pattern recognition”. In: *Computers in Industry* 59.4 (2008), pp. 321–337. ISSN: 01663615. DOI: 10.1016/j.compind.2007.09.001.
- [81] S. Gao and J. J. Shah. “Automatic recognition of interacting machining features based on minimal condition subgraph”. In: *CAD Computer Aided Design* 30.9 (1998), pp. 727–739. ISSN: 00104485. DOI: 10.1016/S0010-4485(98)00033-5.
- [82] P. K. Venuvinod and S. Y. Wong. “A graph-based expert system approach to geometric feature recognition”. In: *Journal of Intelligent Manufacturing* 6.3 (1995), pp. 155–162. ISSN: 09565515. DOI: 10.1007/BF00171444.
- [83] C. F. Yuen and Patri K. Venuvinod. “Geometric feature recognition: Coping with the complexity and infinite variety of features”. In: *International Journal of Computer Integrated Manufacturing* 12.5 (1999), pp. 439–452. ISSN: 13623052. DOI: 10.1080/095119299130173.
- [84] Stanislav Moskalenko. “Modeling of an Automatic CAD-Based Feature Recognition and Retrieval System for Group Technology Application”. PhD thesis. Universität Stuttgart, 2014.
- [85] M. Marefat and R. L. Kashyap. “Geometric Reasoning for Recognition of Three-Dimensional Object Features”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.10 (1990), pp. 949–965. ISSN: 01628828. DOI: 10.1109/34.58868.
- [86] Qiang Ji and Michael M. Marefat. “A Dempster-Shafer approach for recognizing machine features from CAD models”. In: *Pattern Recognition* 36.6 (2003), pp. 1355–1368. ISSN: 00313203. DOI: 10.1016/S0031-3203(02)00260-1.

- [87] Z. Huang and D. Yip-Hoi. “High-level feature recognition using feature relationship graphs”. In: *CAD Computer Aided Design* 34.8 (2002), pp. 561–582. ISSN: 00104485. DOI: 10.1016/S0010-4485(01)00128-2.
- [88] Lujie Ma, Zhengdong Huang, and Yanwei Wang. “Automatic discovery of common design structures in CAD models”. In: *Computers and Graphics (Pergamon)*. 2010. ISBN: 0097-8493. DOI: 10.1016/j.cag.2010.06.002.
- [89] Paolo Di Stefano, Francesco Bianconi, and Luca Di Angelo. “An approach for feature semantics recognition in geometric models”. In: *CAD Computer Aided Design* 36.10 (Sept. 2004), pp. 993–1009. ISSN: 00104485. DOI: 10.1016/j.cad.2003.10.004.
- [90] Yan Cao et al. “Development of feature recognition and extraction system based on NX platform”. In: *Proceedings of the 2nd International Conference on Electronic and Mechanical Engineering and Information Technology, EMEIT 2012*. 2012, pp. 1154–1157. ISBN: 9789078677604. DOI: 10.2991/emeit.2012.252.
- [91] L. K. Kyprianou. “Shape Classification in Computer-aided Design”. PhD Thesis. University of Cambridge, 1980.
- [92] Tony C. Woo. “Feature extraction by volume decomposition”. In: *Proc. Conf. CAD/CAM Tech. Mech. Eng.* Vol. 76. 1982.
- [93] Yong Se Kim. “Convex Decomposition and Solid Geometric Modeling”. PhD thesis. Stanford, CA, USA, 1990.
- [94] Jung Hyun Han, Mike Pratt, and William C. Regli. “Manufacturing feature recognition from solid models: A status report”. In: *IEEE Transactions on Robotics and Automation* 16.6 (2000), pp. 782–796. ISSN: 1042296X. DOI: 10.1109/70.897789.
- [95] Y. S. Kim. “Recognition of form features using convex decomposition”. In: *Computer-Aided Design* 24.9 (1992), pp. 461–476. ISSN: 0010-4485. DOI: [https://doi.org/10.1016/0010-4485\(92\)90027-8](https://doi.org/10.1016/0010-4485(92)90027-8). URL: <https://www.sciencedirect.com/science/article/pii/0010448592900278>.
- [96] Douglas L. Waco and Yong Se Kim. “Geometric Reasoning for Machining Features Using Convex Decomposition”. In: *Proceedings on the Second ACM Symposium on Solid Modeling and Applications*. SMA '93. New York, NY, USA: Association for Computing Machinery, 1993, pp. 323–332. ISBN: 0897915844. DOI: 10.1145/164360.164460. URL: <https://doi.org/10.1145/164360.164460>.

- [97] Douglas L. Waco and Yong Se Kim. “Considerations in Positive to Negative Conversion for Machining Features Using Convex Decomposition”. In: vol. 13th Compu. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 1993, pp. 35–46. DOI: 10.1115/CIE1993-0006. URL: <https://doi.org/10.1115/CIE1993-0006>.
- [98] Eric Wang and Yong Se Kim. “Status of the Form Feature Recognition Method Using Convex Decomposition”. In: vol. Volume 5: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 1997. DOI: 10.1115/DETC97/CIE-4425. URL: <https://doi.org/10.1115/DETC97/CIE-4425>.
- [99] Frédéric Parienté and Yong Se Kim. “Incremental and localized update of convex decomposition used for form feature recognition”. In: *CAD Computer Aided Design* 28.8 (1996), pp. 589–602. ISSN: 00104485. DOI: 10.1016/0010-4485(95)00074-7.
- [100] Hiroshi Sakurai and Chia-Wei Chin. “Defining and Recognizing Cavity and Protrusion by Volumes”. In: vol. 13th Compu. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 1993, pp. 59–65. DOI: 10.1115/CIE1993-0008. URL: <https://doi.org/10.1115/CIE1993-0008>.
- [101] Hiroshi Sakurai and Chia-Wei Chin. “Definition and Recognition of Volume Features for Process Planning”. In: *Advances in Feature Based Manufacturing*. Ed. by Jami J Shah, Martti Mäntylä, and Dana S Nau. Vol. 20. Manufacturing Research and Technology. Elsevier, 1994, pp. 65–80. DOI: <https://doi.org/10.1016/B978-0-444-81600-9.50009-2>. URL: <https://www.sciencedirect.com/science/article/pii/B9780444816009500092>.
- [102] Yoonhwan Woo and Hiroshi Sakurai. “Recognition of maximal features by volume decomposition”. In: *Computer-Aided Design* 34 (2002), pp. 195–207. DOI: 10.1016/S0010-4485(01)00080-X.
- [103] Hiroshi Sakurai. “Decomposing a Delta Volume Into Maximal Convex Volumes and Sequencing Them for Machining”. In: vol. ASME 1994. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 1994, pp. 135–142. DOI: 10.1115/CIE1994-0395. URL: <https://doi.org/10.1115/CIE1994-0395>.

- [104] Hiroshi Sakurai and Parag Dave. "Volume decomposition and feature recognition, part II: Curved objects". In: *CAD Computer Aided Design* 28.6-7 (1996), pp. 519–537. ISSN: 00104485. DOI: 10.1016/0010-4485(95)00067-4.
- [105] Yoonhwan Woo. "Fast cell-based decomposition and applications to solid modeling". In: *CAD Computer Aided Design* (2003). ISSN: 00104485. DOI: 10.1016/S0010-4485(02)00144-6.
- [106] William C. Regli. "Abstract Title of Dissertation: Geometric Algorithms for Recognition of Features from Solid Models". PhD. University of Maryland, 1995.
- [107] Dr White and Lai Mingwu. "Automated hexahedral mesh generation by virtual decomposition". In: *4th International Meshing Roundtable* (1995), pp. 165–176. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.6828%5C&rep=rep1%5C&type=pdf>.
- [108] Alla Sheffer et al. "Virtual Topology Construction and Applications". In: *Geometric Modeling: Theory and Practice* (1997), pp. 247–259. DOI: 10.1007/978-3-642-60607-6_16.
- [109] Alla Sheffer et al. "Virtual Topology Operators for Meshing". In: *International Journal of Computational Geometry & Applications* 10 (2000). DOI: 10.1142/S0218195900000188.
- [110] Ted Blacker et al. "Using virtual topology to simplify the mesh generation process". In: *American Society of Mechanical Engineers, Applied Mechanics Division, AMD* 220.January (1997), pp. 45–50. ISSN: 01608835.
- [111] Alla Sheffer, Ted Blacker, and Michel Bercovier. "Clustering: Automated detail suppression using virtual topology". In: *American Society of Mechanical Engineers, Applied Mechanics Division, AMD* 220.January (1997), pp. 57–64. ISSN: 01608835.
- [112] A. Sheffer. "Model simplification for meshing using face clustering". In: *CAD Computer Aided Design* 33.13 (2001), pp. 925–934. ISSN: 00104485. DOI: 10.1016/S0010-4485(00)00116-0.
- [113] Christopher M. Tierney et al. "Using Mesh-Geometry Relationships to Transfer Analysis Models between CAE Tools". In: *Proceedings of the 22nd International Meshing Roundtable*. 2014, pp. 367–384. DOI: 10.1007/978-3-319-02335-9_21.
- [114] Christopher M. Tierney et al. "Using virtual topology operations to generate analysis topology". In: *CAD Computer Aided Design* 85 (2017), pp. 154–167. ISSN: 00104485. DOI: 10.1016/j.cad.2016.07.015.

- [115] Rafael Bidarra, Klaas Jan De Kraker, and Willem F. Bronsvooort. “Representation and management of feature information in a cellular model”. In: *CAD Computer Aided Design* 30.4 (1998), pp. 301–313. ISSN: 00104485. DOI: 10.1016/S0010-4485(97)00070-5.
- [116] R. Bidarra and W. F. Bronsvooort. “Semantic feature modelling”. In: *CAD Computer Aided Design* 32.3 (2000), pp. 201–225. ISSN: 00104485. DOI: 10.1016/S0010-4485(99)00090-1.
- [117] Cecil G. Armstrong et al. *Djinn. A Geometric Interface for Solid Modelling*. 5. 2002. ISBN: 1874728135.
- [118] K. Y. Lee et al. “A small feature suppression/unsuppression system for preparing b-rep models for analysis”. In: *ACM Symposium on Solid Modeling and Applications, SM* 1.212 (2005), pp. 113–124. ISSN: 18117783. DOI: 10.1145/1060244.1060258.
- [119] Jae Yeol Lee et al. “A cellular topology-based approach to generating progressive solid models from feature-centric models”. In: *CAD Computer Aided Design* 36.3 (Mar. 2004), pp. 217–229. ISSN: 00104485. DOI: 10.1016/S0010-4485(03)00094-0.
- [120] Declan C. Nolan et al. “Automating analysis modelling through the use of simulation intent”. In: 2013.
- [121] Declan C. Nolan et al. “Automatic dimensional reduction and meshing of stiffened thin-wall structures”. In: *Engineering with Computers* 30.4 (2013), pp. 689–701. ISSN: 14355663. DOI: 10.1007/s00366-013-0317-y.
- [122] Declan C. Nolan et al. “Defining simulation intent”. In: *CAD Computer Aided Design* 59 (2015), pp. 50–63. ISSN: 00104485. DOI: 10.1016/j.cad.2014.08.030. arXiv: 1410.7809.
- [123] M. Belaziz, A. Bouras, and J. M. Brun. “Morphological analysis for product design”. In: *CAD Computer Aided Design* 32.5 (2000), pp. 377–388. ISSN: 00104485. DOI: 10.1016/S0010-4485(00)00019-1.
- [124] Y. Lu, R. Gadh, and T. J. Tautges. “Feature based hex meshing methodology: Feature recognition and volume decomposition”. In: *CAD Computer Aided Design* (2001). ISSN: 00104485. DOI: 10.1016/S0010-4485(00)00122-6.
- [125] Baofu Li and Jin Liu. “Detail feature recognition and decomposition in solid model”. In: *CAD Computer Aided Design* 34.5 (2002), pp. 405–414. ISSN: 00104485. DOI: 10.1016/S0010-4485(01)00118-X.

- [126] C. S. Chong, A. Senthil Kumar, and K. H. Lee. “Automatic solid decomposition and reduction for non-manifold geometric model generation”. In: *CAD Computer Aided Design*. Vol. 36. 13. 2004, pp. 1357–1369. ISBN: 6568746800. DOI: 10.1016/j.cad.2004.02.005. arXiv: 1011.1669v3.
- [127] Seungbum Koo and Kunwoo Lee. “Wrap-around operation to make multi-resolution model of part and assembly”. In: *Computers and Graphics (Pergamon)* 26.5 (2002), pp. 687–700. ISSN: 00978493. DOI: 10.1016/S0097-8493(02)00124-3.
- [128] Rui Sun, Shuming Gao, and Wei Zhao. “An approach to B-rep model simplification based on region suppression”. In: *Computers and Graphics (Pergamon)* 34.5 (2010), pp. 556–564. ISSN: 00978493. DOI: 10.1016/j.cag.2010.06.007. URL: <http://dx.doi.org/10.1016/j.cag.2010.06.007>.
- [129] Huawei Zhu, Yanli Shao, and Yusheng Liu. “Automatic Hierarchical Mid-surface Abstraction of Thin-Walled Models Based on Rib Decomposition”. In: *Proceedings - 2015 14th International Conference on Computer-Aided Design and Computer Graphics, CAD/Graphics 2015*. Institute of Electrical and Electronics Engineers Inc., Apr. 2016, pp. 18–25. ISBN: 9781467380201. DOI: 10.1109/CADGRAPHICS.2015.44.
- [130] Flavien Boussuge et al. “Extraction of generative processes from B-Rep shapes and application to idealization transformations”. In: *CAD Computer Aided Design* 46.1 (2014), pp. 79–89. ISSN: 00104485. DOI: 10.1016/j.cad.2013.08.020.
- [131] Flavien Boussuge et al. “Idealized models for FEA derived from generative modeling processes based on extrusion primitives”. In: *Engineering with Computers* 31.3 (2015), pp. 513–527. ISSN: 14355663. DOI: 10.1007/s00366-014-0382-x.
- [132] C. E. Silva. *Alternative Definitions of Faces in Boundary Representatives of Solid Objects*. Tech. rep. Rochester, N.Y.: University of Rochester. Production Automation Project, 1981, p. 27. URL: <http://hdl.handle.net/1802/1184>.
- [133] Trevor T. Robinson et al. “Automated mixed dimensional modelling for the finite element analysis of swept and revolved CAD features”. In: *Proceedings SPM 2006 - ACM Symposium on Solid and Physical Modeling 2006*. May 2014 (2006), pp. 117–128. DOI: 10.1145/1128888.1128905.
- [134] Chentao L. *Preparation of CAD Model for Finite Element Analysis*. CMCE. ISBN: 9781424479566.

- [135] S. J. Tate and G. E.M. Jared. “Recognising symmetry in solid models”. In: *CAD Computer Aided Design* 35.7 (2003), pp. 673–692. ISSN: 00104485. DOI: 10.1016/S0010-4485(02)00093-3.
- [136] Flavien Boussuge et al. “Symmetry-based decomposition for meshing quasi-axisymmetric components”. In: *Procedia Engineering*. Vol. 203. 2017, pp. 375–387. DOI: 10.1016/j.proeng.2017.09.812.
- [137] Christopher M. Tierney et al. “Efficient symmetry-based decomposition for meshing quasi-axisymmetric assemblies”. In: *Computer-Aided Design and Applications* 16.3 (2019), pp. 478–495. ISSN: 16864360. DOI: 10.14733/cadaps.2019.478-495. URL: <https://www.researchgate.net/publication/327345494>.
- [138] Jorge Casero et al. “Idealizing Quasi-Axisymmetric 3D Geometries to 2D-Axisymmetric Finite Element Models”. In: *Computer-Aided Design and Applications* 16.6 (Mar. 2019), pp. 1020–1033. DOI: 10.14733/cadaps.2019.1020-1033.
- [139] Harry Blum. *A Transformation for Extracting New Descriptors of Shape*. 1967.
- [140] Harry Blum. “Biological shape and visual science (part I)”. In: *Journal of Theoretical Biology* 38.2 (1973), pp. 205–287. ISSN: 10958541. DOI: 10.1016/0022-5193(73)90175-6.
- [141] D. T. Lee. “Medial Axis Transformation of a Planar Shape”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-4.4 (1982), pp. 363–369. ISSN: 01628828. DOI: 10.1109/TPAMI.1982.4767267.
- [142] Halit Nebi Gürsoy. “Shape interrogation by medial axis transform for automated analysis”. PhD thesis. Massachusetts Institute of Technology, 1989.
- [143] Halit Nebi Gürsoy and Nicholas M. Patrikalakis. “An automatic coarse and fine surface mesh generation scheme based on medial axis transform: Part i algorithms”. In: *Engineering with Computers* 8.3 (1992), pp. 121–137. ISSN: 01770667. DOI: 10.1007/BF01200364.
- [144] Evan C. Sherbrooke, Nicholas M. Patrikalakis, and Erik Brisson. “Computation of the medial axis transform of 3-D polyhedra”. In: *Symposium on Solid Modeling and Applications - Proceedings* March 2014 (1995), pp. 187–199. DOI: 10.1145/218013.218059.
- [145] Yong Gu Lee and Kunwoo Lee. “Computing the medial surface of a 3-D boundary representation model”. In: *Advances in Engineering Software* 28.9 (1997), pp. 593–605. ISSN: 09659978. DOI: 10.1016/S0965-9978(97)00024-0.

- [146] Rajesh Ramamurthy and Rida T. Farouki. "Voronoi diagram and medial axis algorithm for planar domains with curved boundaries I. Theoretical foundations". In: *Journal of Computational and Applied Mathematics* 102.1 (1999), pp. 119–141. ISSN: 03770427. DOI: 10.1016/S0377-0427(98)00211-8.
- [147] Siavash N. Meshkat and Constantine M. Sakkas. "Voronoi diagram for multiply-connected polygonal domains II: Implementation and application". In: *IBM Journal of Research and Development* 31.3 (1987), pp. 373–381. DOI: 10.1147/rd.313.0373.
- [148] Vijay Srinivasan and Lee R. Nackman. "Voronoi diagram for multiply-connected polygonal domains I: Algorithm". In: *IBM Journal of Research and Development* 31.3 (1987), pp. 361–372. DOI: 10.1147/rd.313.0361.
- [149] Damian J. Sheehy, Cecil G. Armstrong, and Desmond. J. Robinson. "Computing the medial surface of a solid from a domain Delaunay triangulation". In: *Symposium on Solid Modeling and Applications - Proceedings*. 1995, pp. 201–212. DOI: 10.1145/218013.218062.
- [150] Y. C. Chang et al. "Medial Axis Transform (MAT) of general 2D shapes and 3D polyhedra for engineering applications". In: *IFIP Advances in Information and Communication Technology* 75 (2001), pp. 37–52. ISSN: 18684238. DOI: 10.1007/978-0-387-35490-3.
- [151] M. Ramanathan and B. Gurumoorthy. "Constructing medial axis transform of planar domains with curved boundaries". In: *CAD Computer Aided Design* 35.7 (2003), pp. 619–632. ISSN: 00104485. DOI: 10.1016/S0010-4485(02)00085-4.
- [152] Tamal K. Dey and Wulue Zhao. "Approximating the medial axis from the Voronoi diagram with a convergence guarantee". In: *Algorithmica (New York)* 38.1 (2003), pp. 179–200. ISSN: 01784617. DOI: 10.1007/s00453-003-1049-y.
- [153] Tamal K. Dey, Hyuckje Woo, and Wulue Zhao. "Approximate medial axis for CAD models". In: *Proceedings of the Symposium on Solid Modeling and Applications*. 2003, pp. 280–285. DOI: 10.1145/781650.781652.
- [154] Frédéric Chazal and André Lieutier. "The " λ -medial axis". In: *Graphical Models* (2005). ISSN: 15240703. DOI: 10.1016/j.gmod.2005.01.002.
- [155] Luzhong Yin, Xiaojuan Luo, and Mark S. Shephard. "Identifying and meshing thin sections of 3-d curved domains". In: *Proceedings of the 14th International Meshing Roundtable, IMR 2005*. 2005. ISBN: 3540251375. DOI: 10.1007/3-540-29090-7-3.

- [156] O. Aichholzer et al. “Medial axis computation for planar free-form shapes”. In: *CAD Computer Aided Design* (2009). ISSN: 00104485. DOI: 10.1016/j.cad.2008.08.008.
- [157] Hao Xia and Paul G. Tucker. “Distance solutions for medial axis transform”. In: *Proceedings of the 18th International Meshing Roundtable, IMR 2009*. 2009. ISBN: 9783642043185. DOI: 10.1007/978-3-642-04319-2_15.
- [158] Ata A. Eftekharian and Horea T. Ilieş. “Distance functions and skeletal representations of rigid and non-rigid planar shapes”. In: *CAD Computer Aided Design* (2009). ISSN: 00104485. DOI: 10.1016/j.cad.2009.05.006.
- [159] Rossi Luca. *Medial Surface Extraction for 3D Shape Representation Skeleton : Definition*. 2010.
- [160] Nataša Petrović. *Medial Axis Extraction and Thickness Measurement of Formed Sheet Metal Parts*. Tech. rep. 2010.
- [161] Feng Sun et al. “Medial Meshes for Volume Approximation”. In: (2013). ISSN: 10772626. DOI: 10.1109/TVCG.2015.2448080. arXiv: 1308.3917.
- [162] Housheng Zhu et al. “Calculating the medial axis of a CAD model by multi-CPU based parallel computation”. In: *Advances in Engineering Software* (2015). ISSN: 18735339. DOI: 10.1016/j.advengsoft.2015.03.004.
- [163] Housheng Zhu et al. “Constructive generation of the medial axis for solid models”. In: *CAD Computer Aided Design* 62 (2015), pp. 98–111. ISSN: 00104485. DOI: 10.1016/j.cad.2014.11.003.
- [164] Jacob Kresslein et al. “Automated cross-sectional shape recovery of 3D branching structures from point cloud”. In: *Journal of Computational Design and Engineering* 5.3 (July 2018), pp. 368–378. ISSN: 22885048. DOI: 10.1016/j.jcde.2017.11.010.
- [165] Shuangmin Chen et al. “Fast and robust shape diameter function”. In: *PLoS ONE* 13.1 (Jan. 2018). ISSN: 19326203. DOI: 10.1371/journal.pone.0190666.
- [166] Maurizio Kovacic et al. “Fast approximation of the shape diameter function”. In: *2nd International Workshop on Computer Graphics, Computer Vision and Mathematics, GraVisMa 2010 - Workshop Proceedings*. 2010, pp. 65–72. ISBN: 9788086943855.
- [167] Xavier Rolland-Nevière, Gwenaël Doërr, and Pierre Alliez. “Robust diameter-based thickness estimation of 3D objects”. In: *Graphical Models* 75.6 (2013), pp. 279–296. ISSN: 15240703. DOI: 10.1016/j.gmod.2013.06.001.

-
- [168] B. Delaunay. “Sur la sphere vide”. In: *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7 (1934), pp. 793–800.
- [169] Mohsen Rezayat. “Midsurface abstraction from 3D solid models: General theory and applications”. In: *CAD Computer Aided Design* 28.11 (1996), pp. 905–915. ISSN: 00104485. DOI: 10.1016/0010-4485(96)00018-8.
- [170] M. Ramanathan and B. Gurumoorthy. “Generating the Mid-Surface of a Solid using 2D MAT of its Faces”. In: *Computer-Aided Design and Applications* 1.1-4 (2004), pp. 665–674. ISSN: 16864360. DOI: 10.1080/16864360.2004.10738312.
- [171] Hanmin Lee, Yong Youn Nam, and Seong Whan Park. “Graph-based midsurface extraction for finite element analysis”. In: *Proceedings of the 2007 11th International Conference on Computer Supported Cooperative Work in Design, CSCWD* (2007), pp. 1055–1058. DOI: 10.1109/CSCWD.2007.4281585.
- [172] Dong-Pyoung Sheen et al. “Dimension Reduction of Solid Models by Mid-Surface Generation”. In: *International Journal of CAD/CAM* 7.1 (2007), pp. 71–80. ISSN: 1598-1800.
- [173] Dong Pyoung Sheen et al. “Transformation of a thin-walled solid model into a surface model via solid deflation”. In: *CAD Computer Aided Design* 42.8 (2010), pp. 720–730. ISSN: 00104485. DOI: 10.1016/j.cad.2010.01.003. URL: <http://dx.doi.org/10.1016/j.cad.2010.01.003>.
- [174] Sang Hun Lee. “Offsetting operations on non-manifold topological models”. In: *CAD Computer Aided Design* 41.11 (2009), pp. 830–846. ISSN: 00104485. DOI: 10.1016/j.cad.2009.05.001. URL: <http://dx.doi.org/10.1016/j.cad.2009.05.001>.
- [175] Felix Stanley et al. *Design optimization schemes to reduce Tip clearance uncertainties in Gas Turbine Engines UTC for Computational Engineering*. Tech. rep. URL: www.soton.ac.uk/ses/research/ced/posters.html%5C%7C.
- [176] *MATLAB version 9.10.0.1613233 (R2021a)*. The Mathworks, Inc. Natick, Massachusetts, 2021.
- [177] Leran Wang et al. “An accelerated medial object transformation for whole engine optimisation”. In: *Proceedings of the ASME Turbo Expo*. Vol. 2B. 2014. ISBN: 9780791845615. DOI: 10.1115/GT2014-26014.
- [178] Leran Wang et al. “A Whole Engine Optimization based on Medial Object Transformations”. In: *Proceedings of the 27th ISABE Conference* (2017), p. 11. URL: <https://eprints.soton.ac.uk/id/eprint/413784>.

- [179] Hau Kit Yong et al. *Multi-fidelity kriging-based optimization of engine subsystem models with medial meshes*. Tech. rep. 2018.
- [180] R. J. Donaghy et al. “Dimensional reduction of analysis models”. In: *Proc. 5th International Meshing Roundtable* October (1996).
- [181] R. J. Donaghy, Cecil G. Armstrong, and M. A. Price. “Dimensional reduction of surface models for analysis”. In: *Engineering with Computers* 16.1 (2000), pp. 24–35. ISSN: 01770667. DOI: 10.1007/s003660050034.
- [182] Cecil G. Armstrong, Declan J. Robinson, and R. M. McKeag. “Medials for meshing and more”. In: *4th International Meshing Roundtable* October (1995).
- [183] Trevor T. Robinson et al. “Automated mixed dimensional modelling with the medial object”. In: *Proceedings of the 17th International Meshing Roundtable, IMR 2008*. 2008, pp. 281–298. ISBN: 9783540879206. DOI: 10.1007/978-3-540-87921-3_17.
- [184] Jonathan E. Makem, Cecil G. Armstrong, and Trevor T. Robinson. “Automatic decomposition and efficient semi-structured meshing of complex solids”. In: *Engineering with Computers*. Vol. 30. 3. 2014, pp. 345–361. DOI: 10.1007/s00366-012-0302-x.
- [185] Yoonhwan Woo. “Abstraction of mid-surfaces from solid models of thin-walled parts: A divide-and-conquer approach”. In: *CAD Computer Aided Design* 47 (2014), pp. 1–11. ISSN: 00104485. DOI: 10.1016/j.cad.2013.08.010.
- [186] Adele Nasti. “Simulation framework for preliminary structural design of aero engines”. In: *International CAE Conference and Exhibition*. Vicenza, Italy, 2017.
- [187] Michelle Tindall, Akin Keskin, and Andrew Layton. “Industrial challenges in large thermally enabled structural whole engine models”. In: *Proceedings of the ASME Turbo Expo*. Vol. 2D-2020. 2020, pp. 1–11. ISBN: 9780791884096. DOI: 10.1115/GT2020-15207.
- [188] Ashwin Kannan et al. “Novel feature based approach for turbomachinery design and analysis”. In: *Proceedings of the ASME Turbo Expo*. Vol. 2C-2020. 2020, pp. 1–9. ISBN: 9780791884089. DOI: 10.1115/GT2020-16325.
- [189] Benoit Lecallard et al. “Automatic hexahedral-dominant meshing for decomposed geometries of complex components”. In: *Computer-Aided Design and Applications* 16.5 (2019), pp. 846–863. ISSN: 16864360. DOI: 10.14733/cadaps.2019.846-863.

- [190] Qingqing Feng, Xionghui Zhou, and Junjie Li. “A hybrid and automated approach to adapt geometry model for CAD/CAE integration”. In: *Engineering with Computers* 36.2 (2020), pp. 543–563. ISSN: 14355663. DOI: 10.1007/s00366-019-00713-4.
- [191] Sang Hun Lee. “A CAD-CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques”. In: *CAD Computer Aided Design* 37.9 (Aug. 2005), pp. 941–955. ISSN: 00104485. DOI: 10.1016/j.cad.2004.09.021.
- [192] Okba Hamri and Jean-Claude Léon. “Interoperability between CAD and Simulation Models for Cooperative Design”. In: *Methods and Tools for Co-operative and Integrated Design* (2004), pp. 451–462. DOI: 10.1007/978-94-017-2256-8_38.
- [193] G. P. Gujarathi and Y. S. Ma. “Parametric CAD/CAE integration using a common data model”. In: *Journal of Manufacturing Systems* (2011). ISSN: 02786125. DOI: 10.1016/j.jmsy.2011.01.002.
- [194] Zhaohui Xia et al. “A CAD/CAE incorporate software framework using a unified representation architecture”. In: *Advances in Engineering Software* 87 (2015), pp. 68–85. ISSN: 18735339. DOI: 10.1016/j.advengsoft.2015.05.005. URL: <http://dx.doi.org/10.1016/j.advengsoft.2015.05.005>.
- [195] Zhaohui Xia et al. “A novel approach for automatic reconstruction of boundary condition in structure analysis”. In: *Advances in Engineering Software* 96 (2016), pp. 38–57. ISSN: 18735339. DOI: 10.1016/j.advengsoft.2016.02.001.
- [196] Y. S. Ma and T. Tong. “Associative feature modeling for concurrent engineering integration”. In: *Computers in Industry* 51.1 (2003), pp. 51–71. DOI: 10.1016/S0166-3615(03)00025-3.
- [197] Hamidullah, E. Bohez, and M. A. Irfan. “Assembly features: Definition, classification, and instantiation”. In: *Proceedings - 2nd International Conference on Emerging Technologies 2006, ICET 2006* November (2006), pp. 617–623. DOI: 10.1109/ICET.2006.335941.
- [198] Y. S. Ma et al. “Associative assembly design features: Concept, implementation and application”. In: *International Journal of Advanced Manufacturing Technology* 32.5-6 (2007), pp. 434–444. ISSN: 02683768. DOI: 10.1007/s00170-005-0371-8.
- [199] Adam Dixon and Jami J. Shah. “Assembly feature tutor and recognition algorithms based on mating face Pairs”. In: *Computer-Aided Design and Applications* 7.3 (2010), pp. 319–333. ISSN: 16864360. DOI: 10.3722/cadaps.2010.319-333.

- [200] Robert Iacob, Peter Mitrouchev, and Jean Claude Léon. “Assembly simulation incorporating component mobility modelling based on functional surfaces”. In: *International Journal on Interactive Design and Manufacturing* 5.2 (2011), pp. 119–132. ISSN: 19552513. DOI: 10.1007/s12008-011-0120-1.
- [201] Ahmad Shahwan et al. “Qualitative behavioral reasoning from components’ interfaces to components’ functions for DMU adaption to FE analyses”. In: *CAD Computer Aided Design* 45.2 (2013), pp. 383–394. ISSN: 00104485. DOI: 10.1016/j.cad.2012.10.021. URL: <http://dx.doi.org/10.1016/j.cad.2012.10.021>.
- [202] Diana Popescu and Robert Iacob. “Disassembly method based on connection interface and mobility operator concepts”. In: *International Journal of Advanced Manufacturing Technology* 69.5-8 (2013), pp. 1511–1525. ISSN: 14333015. DOI: 10.1007/s00170-013-5092-9.
- [203] François Jourdes et al. “Computation of components’ interfaces in highly complex assemblies”. In: *CAD Computer Aided Design* 46.1 (2014), pp. 170–178. ISSN: 00104485. DOI: 10.1016/j.cad.2013.08.029. URL: <http://dx.doi.org/10.1016/j.cad.2013.08.029>.
- [204] Baha A. Hasan, Jan Wikander, and Mauro Onori. “Assembly design semantic recognition using solid works-APP”. In: *International Journal of Mechanical Engineering and Robotics Research* 5.4 (2016), pp. 280–287. ISSN: 22780149. DOI: 10.18178/ijmerr.5.4.280-287.
- [205] Jie Zhang et al. “Generic face adjacency graph for automatic common design structure discovery in assembly models”. In: *CAD Computer Aided Design* 45.8-9 (2013), pp. 1138–1151. ISSN: 00104485. DOI: 10.1016/j.cad.2013.04.003. URL: <http://dx.doi.org/10.1016/j.cad.2013.04.003>.
- [206] Abhijit S. Deshmukh et al. “Content-based assembly search: A step towards assembly reuse”. In: *CAD Computer Aided Design* 40.2 (Feb. 2008), pp. 244–261. ISSN: 00104485. DOI: 10.1016/j.cad.2007.10.012.
- [207] Xiang Chen et al. “A flexible assembly retrieval approach for model reuse”. In: *CAD Computer Aided Design* 44.6 (June 2012), pp. 554–574. ISSN: 00104485. DOI: 10.1016/j.cad.2012.02.001.
- [208] Katia Lupinetti et al. “CAD assembly descriptors for knowledge capitalization and model retrieval”. In: *Tools and Methods for Competitive Engineering* December 2019 (2016).

- [209] Katia Lupinetti et al. “Multi-criteria retrieval of CAD assembly models”. In: *Journal of Computational Design and Engineering* 5.1 (2018), pp. 41–53. ISSN: 22885048. DOI: 10.1016/j.jcde.2017.11.003.
- [210] Katia Lupinetti et al. “Content-based multi-criteria similarity assessment of CAD assembly models”. In: *Computers in Industry* 112 (Nov. 2019). ISSN: 01663615. DOI: 10.1016/j.compind.2019.07.001.
- [211] Johan W.H. Tangelder and Remco C. Veltkamp. “A survey of content based 3D shape retrieval methods”. In: *Multimedia Tools and Applications* 39.3 (Sept. 2008), pp. 441–471. ISSN: 13807501. DOI: 10.1007/s11042-007-0181-0.
- [212] Adem Çiçek and Mahmut Gülesin. “A part recognition based computer aided assembly system”. In: *Computers in Industry* 58.8-9 (Dec. 2007), pp. 733–746. ISSN: 01663615. DOI: 10.1016/j.compind.2007.02.007.
- [213] Jing Bai et al. “Design reuse oriented partial retrieval of CAD models”. In: *CAD Computer Aided Design* 42.12 (Dec. 2010), pp. 1069–1084. ISSN: 00104485. DOI: 10.1016/j.cad.2010.07.002.
- [214] Zhansong Wang, Ling Tian, and Wenrui Duan. “Annotation and retrieval system of CAD models based on functional semantics”. In: *Chinese Journal of Mechanical Engineering* 27.6 (Nov. 2014), pp. 1112–1124. ISSN: 1000-9345. DOI: 10.3901/cjme.2014.0815.134.
- [215] Zhoupeng Han, Rong Mo, and Li Hao. “Clustering and retrieval of mechanical CAD assembly models based on multi-source attributes information”. In: *Robotics and Computer-Integrated Manufacturing* 58.March (2019), pp. 220–229. ISSN: 07365845. DOI: 10.1016/j.rcim.2019.01.003. URL: <https://doi.org/10.1016/j.rcim.2019.01.003>.
- [216] Pan Wang et al. “An assembly retrieval approach based on shape distributions and Earth Mover’s Distance”. In: *International Journal of Advanced Manufacturing Technology* 86.9-12 (2016), pp. 2635–2651. ISSN: 14333015. DOI: 10.1007/s00170-016-8368-z. URL: <http://dx.doi.org/10.1007/s00170-016-8368-z>.
- [217] Ahmad Shahwan and Lionel Fine. “Deriving Functional Properties of Components From the Analysis of Digital Mock-Ups”. In: *Proceedings of TMCE*. January. 2012, pp. 1–15.

- [218] Harold Vilmart, Jean Claude Léon, and Federico Ulliana. “From CAD assemblies toward knowledge-based assemblies using an intrinsic knowledge-based assembly model”. In: *Computer-Aided Design and Applications* 15.3 (2018), pp. 300–317. ISSN: 16864360. DOI: 10.1080/16864360.2017.1397882.
- [219] S. Krishna. *Introduction to Database and Knowledge-base Systems*. Computer Science Series. World Scientific, 1992, p. 328. ISBN: 9789810206208.
- [220] Frederick Hayes-Roth, Donald A. Waterman, and Douglas B. Lenat. *Building Expert Systems*. USA: Addison-Wesley Longman Publishing Co., Inc., 1983. ISBN: 0201106868.
- [221] Aviv Segev, Moshe Leshno, and Moshe Zviran. “Context recognition using internet as a knowledge base”. In: *Journal of Intelligent Information Systems* 29.3 (2007), pp. 305–327. ISSN: 09259902. DOI: 10.1007/s10844-006-0015-y.
- [222] Simon Kendal and Malcolm Creen. *An introduction to knowledge engineering*. London: Springer, London, 2007, pp. X, 290. ISBN: 978-1-84628-475-5. DOI: 10.1007/978-1-84628-667-4. URL: http://www.computingreviews.com/review/review_review.cfm?review_id=133988.
- [223] R. Akerkar and P. Sajja. *Knowledge-Based Systems*. Jones & Bartlett Learning, 2009. ISBN: 9781449662707.
- [224] C. B. Chapman and M. Pinfold. “Design engineering - a need to rethink the solution using knowledge based engineering”. In: *Knowledge-Based Systems* 12.5-6 (1999), pp. 257–267. ISSN: 09507051. DOI: 10.1016/S0950-7051(99)00013-1.
- [225] Craig B. Chapman and Martyn Pinfold. “The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure”. In: *Advances in Engineering Software* 32.12 (2001), pp. 903–912. ISSN: 09659978. DOI: 10.1016/S0965-9978(01)00041-2.
- [226] Stephen Cooper, Ip-shing Fan, and Guihua Li. “Achieving Competitive Advantage Through Knowledge-Based Engineering A Best Practice Guide”. In: *Department of Trade and Industry* (2001), p. 21.
- [227] Bernal Z.A. Fabio, Rojo V. Alejandro, and Tornero M. Josep. “Application of NX knowledge fusion module for the design automation of an automotive painting defects inspection tunnel”. In: *Computer-Aided Design and Applications* 9.5 (2012), pp. 655–664. ISSN: 16864360. DOI: 10.3722/cadaps.2012.655-664.

- [228] Pablo Bermell García and Fan Ip-Shing. “A KBE System for the design of wind tunnel models using reusable knowledge components”. In: Barcelona: Asociación Española de Dirección e Ingeniería de Proyectos (AEIPRO), 2002, pp. 0594–0604.
- [229] Michael Negnevitsky. *Artificial Intelligence: A Guide to Intelligent Systems*. 1st. USA: Addison-Wesley Longman Publishing Co., Inc., 2001, p. 394. ISBN: 978-0-201-71159-2.
- [230] Gianfranco La Rocca. “Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design”. In: *Advanced Engineering Informatics* 26.2 (2012), pp. 159–179. ISSN: 14740346. DOI: 10.1016/j.aei.2012.02.002. URL: <http://dx.doi.org/10.1016/j.aei.2012.02.002>.
- [231] Siemens Product Lifecycle Management Software Inc. *NX™ Knowledge Fusion Knowledge Based Engineering*. 2008. URL: https://www.plm.automation.siemens.com/de_de/Images/knowledge_fusion_tcm73-62405.pdf.
- [232] William D. O’Brien et al. “Using Knowledge-based Solid Modeling Techniques and Airfoil Design Data: A Case Study in Developing an Airfoil Seed Part Generator”. In: *Proceedings of The 2006 IJME - INTERTECH Conference*. January. Union, NJ, 2006. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.1360%5C&rep=rep1%5C&type=pdf>.
- [233] Sai Zeng et al. “ZAP: A knowledge-based FEA modeling method for highly coupled variable topology multi-body problems”. In: *Engineering with Computers* 24.4 (2008), pp. 359–381. ISSN: 01770667. DOI: 10.1007/s00366-007-0086-6.
- [234] Pei Zhan et al. “Knowledge representation and ontology mapping methods for product data in engineering applications”. In: *Journal of Computing and Information Science in Engineering* 10.2 (2010), pp. 1–11. ISSN: 15309827. DOI: 10.1115/1.3330432.
- [235] Bojan Dolšak, Ivan Bratko, and Anton Jezernik. “Knowledge base for finite-element mesh design learned by inductive logic programming”. In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM* 12.2 (1998), pp. 95–106. ISSN: 08900604. DOI: 10.1017/s0890060498122023.
- [236] Sebastian Katona, Michael Koch, and Sandro Wartzack. “An Approach of a Knowledge-based Process to Integrate Real Geometry Models in Product Simulations”. In: *Procedia CIRP* 50 (2016), pp. 813–818. ISSN: 22128271. DOI: 10.1016/j.procir.2016.04.176. URL: <http://dx.doi.org/10.1016/j.procir.2016.04.176>.

- [237] Sebastian Katona, Michael Koch, and Sandro Wartzack. “Integrating real geometry models into product simulations: an approach of knowledge-based process”. In: *Proceedings of the 14th International Design Conference DESIGN 2016*. Ed. by Pavkovic N Bojctetic N Skec S Marjanovic D Storga M. 2016, pp. 689–696.
- [238] Sandra Vieira, Walter Hugo Lopez Pinaya, and Andrea Mechelli. *Introduction to machine learning*. 2019, pp. 1–20. ISBN: 9780128157398. DOI: 10.1016/B978-0-12-815739-8.00001-8.
- [239] François Chollet. *Deep Learning with Python*. Manning, Nov. 2017. ISBN: 9781617294433.
- [240] Wolfgang Ertel. *Introduction to Artificial Intelligence*. 2011. ISBN: 9783319584867.
- [241] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [242] Charu C Aggarwal. *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing, 2018, p. 497. ISBN: 9783319944630.
- [243] Gopinath Rebala, Ajay Ravi, and Sanjay Churiwala. *An Introduction to Machine Learning*. 2019. ISBN: 978-3-030-15728-9. DOI: 10.1007/978-3-030-15729-6.
- [244] Ameet Joshi. *Machine Learning and Artificial Intelligence*. Vol. 1. 1. Springer International Publishing, 2020. ISBN: 978-3-030-26621-9. DOI: 10.1007/978-3-030-26622-6.
- [245] Kai Ming Ting. “Confusion Matrix”. In: *Encyclopedia of Machine Learning and Data Mining*. Ed. by Claude Sammut and Geoffrey I Webb. Boston, MA: Springer US, 2017, p. 260. ISBN: 978-1-4899-7687-1. DOI: 10.1007/978-1-4899-7687-1_50. URL: https://doi.org/10.1007/978-1-4899-7687-1_50.
- [246] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979. ISBN: 9780408709293.
- [247] C. J. van Rijsbergen, F. Crestani, and M. Lalmas. *Information Retrieval: Uncertainty and Logics: Advanced Models for the Representation and Retrieval of Information*. The Information Retrieval Series. Springer US, 2012. ISBN: 9781461556176.
- [248] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [249] C. Sammut and G. I. Webb. *Encyclopedia of Machine Learning*. Encyclopedia of Machine Learning. Springer US, 2011. ISBN: 9780387307688.

- [250] Sebastian Koch et al. “ABC: A big cad model dataset for geometric deep learning”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2019-June. 2019, pp. 9593–9603. ISBN: 9781728132938. DOI: 10.1109/CVPR.2019.00983. arXiv: 1812.06216. URL: <https://deep-geometry.github.io/abc-dataset>.
- [251] Liang Liang et al. “A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis”. In: *Journal of the Royal Society Interface* 15.138 (Jan. 2018). ISSN: 17425662. DOI: 10.1098/rsif.2017.0844.
- [252] Francois Roewer-Despres, Najeeb Khan, and Ian Stavness. “Towards Finite-Element Simulation Using Deep Learning”. In: *15th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering* (2018), p. 2018. URL: http://cmbbe2018.tecnico.ulisboa.pt/pen_cmbbe2018/pdf/WEB_PAPERS/CMBBE2018_paper_89.pdf.
- [253] German Capuano and Julian Rimoli. “Smart finite elements: A novel machine learning application”. In: *Computer Methods in Applied Mechanics and Engineering* 345 (2018). DOI: 10.1016/j.cma.2018.10.046.
- [254] German Capuano. “Smart Finite Elements: An Application Of Machine Learning To Reduced-Order Modeling Of Multi-Scale Problems”. PhD. Georgia Institute of Technology, 2019. URL: <http://hdl.handle.net/1853/61722>.
- [255] Larry Manevitz, Malik Yousef, and Dan Givoli. “Finite-element mesh generation using self-organizing neural networks”. In: *Computer-Aided Civil and Infrastructure Engineering* 12.4 (1997), pp. 233–250. ISSN: 10939687. DOI: 10.1111/0885-9507.00060.
- [256] Florence Danglade, Jean Philippe Pernot, and Philippe Véron. “On the use of Machine Learning to Defeature CAD Models for Simulation”. In: *Computer-Aided Design and Applications* (2014). ISSN: 16864360. DOI: 10.1080/16864360.2013.863510.
- [257] Florence Danglade et al. “A priori evaluation of simulation models preparation processes using artificial intelligence techniques”. In: *Computers in Industry* 91 (2017), pp. 45–61. ISSN: 01663615. DOI: 10.1016/j.compind.2017.06.001. URL: <http://dx.doi.org/10.1016/j.compind.2017.06.001>.
- [258] Philipp Kestel et al. “Konzept zur Umsetzung einer zentralen Wissensbasis für Simulationswissen in ANSYS Engineering Knowledge Manager”. In: *ANSYS Conference & 33th CADFEM Users’ Meeting*. 2015.

- [259] Philipp Kestel, T. Schneyer, and Sandro Wartzack. “Feature-based approach for the automated setup of accurate, design accompanying finite element analyses”. In: *Proceedings of International Design Conference, DESIGN DS 84* (2016), pp. 697–706. ISSN: 18479073.
- [260] Jie Zhou et al. “Graph Neural Networks: A Review of Methods and Applications”. In: *arXiv* (2018), pp. 1–22. ISSN: 23318422. arXiv: 1812.08434.
- [261] Federico Errica et al. “A Fair Comparison of GNN”. In: *Proceedings of the International Conference on Learning Representations* (2020), pp. 1–14. arXiv: 1912.09893. URL: <http://arxiv.org/abs/1912.09893>.
- [262] Zahra Jandaghi and Liming Cai. “On Graph Learning with Neural Networks”. In: *Machine Learning, Optimization, and Data Science*. Ed. by Giuseppe Nicosia et al. Cham: Springer International Publishing, 2020, pp. 516–528. ISBN: 978-3-030-64580-9.
- [263] Peter Battaglia et al. “Interaction networks for learning about objects, relations and physics”. In: *Advances in Neural Information Processing Systems* (2016), pp. 4509–4517. ISSN: 10495258. arXiv: 1612.00222.
- [264] Junyoung Park and Jinkyoo Park. “Physics-induced graph neural network: An application to wind-farm power estimation”. In: *Energy* 187 (2019), p. 115883. ISSN: 03605442. DOI: 10.1016/j.energy.2019.115883. URL: <https://doi.org/10.1016/j.energy.2019.115883>.
- [265] Zhen Zhang et al. “Hierarchical graph pooling with structure learning”. In: *arXiv* (2019). ISSN: 23318422. arXiv: 1911.05954.
- [266] Ivan E. Sutherland, Robert F. Sproull, and Robert A. Schumacker. “A Characterization of Ten Hidden-Surface Algorithms”. In: *ACM Computing Surveys (CSUR)* 6.1 (1974), pp. 1–55. ISSN: 15577341. DOI: 10.1145/356625.356626.
- [267] Jami J. Shah and Martti Mäntylä. “Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications”. In: 1995.
- [268] Sean Gillies et al. *Shapely: manipulation and analysis of geometric objects*. URL: <https://github.com/Toblerity/Shapely>.
- [269] Jon Louis Bentley. “Multidimensional Binary Search Trees Used for Associative Searching”. In: *Commun. ACM* 18.9 (Sept. 1975), pp. 509–517. ISSN: 0001-0782. DOI: 10.1145/361002.361007. URL: <https://doi.org/10.1145/361002.361007>.

- [270] Jon Louis Bentley. “K-d Trees for Semidynamic Point Sets”. In: *Proceedings of the Sixth Annual Symposium on Computational Geometry*. SCG '90. New York, NY, USA: Association for Computing Machinery, 1990, pp. 187–197. ISBN: 0897913620. DOI: 10.1145/98524.98564. URL: <https://doi.org/10.1145/98524.98564>.
- [271] Rolf Klein. *Algorithmische Geometrie - Grundlagen, Methoden, Anwendungen*. Springer-Verlag Berlin Heidelberg, 2005. ISBN: 978-3-540-27619-7. DOI: 10.1007/3-540-27619-X.
- [272] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.
- [273] Benjamin Spiess, Klaus Höschler, and Maren Fanter. “An Automated Silhouette-based Segmentation and Semi-parametric Geometry Reconstruction of Quasi-axisymmetric Aero Engine Structures”. In: NAFEMS World Congress 2021, Oct. 2021. URL: https://www.nafems.org/publications/resource_center/nwc21-235/.
- [274] Dexter C. Kozen and Dexter C. Kozen. “Depth-First and Breadth-First Search”. In: *The Design and Analysis of Algorithms*. 1992. DOI: 10.1007/978-1-4612-4400-4_4.
- [275] David H. Douglas and Thomas K. Peucker. “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature”. In: *Cartographica: The International Journal for Geographic Information and Geovisualization* 10.2 (1973). DOI: 10.3138/fm57-6770-u75u-7727.
- [276] Urs Ramer. “An iterative procedure for the polygonal approximation of plane curves”. In: *Computer Graphics and Image Processing* 1.3 (1972), pp. 244–256. ISSN: 0146-664X. DOI: [https://doi.org/10.1016/S0146-664X\(72\)80017-0](https://doi.org/10.1016/S0146-664X(72)80017-0). URL: <https://www.sciencedirect.com/science/article/pii/S0146664X72800170>.
- [277] Chin Shyurng Fahn, Jhing Fa Wang, and Jau Yien Lee. “An Adaptive Reduction Procedure for the Piecewise Linear Approximation of Digitized Curves”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11.9 (1989). ISSN: 01628828. DOI: 10.1109/34.35499.
- [278] Eamonn Keogh et al. *Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases*. Tech. rep. 2001.

- [279] Leif Bergerhoff, Joachim Weickert, and Yehuda Dar. “Algorithms for piecewise constant signal approximations”. In: *European Signal Processing Conference 2019-Septe*.741215 (2019). ISSN: 22195491. DOI: 10.23919/EUSIPCO.2019.8902559. arXiv: 1903.01320.
- [280] M. I. Friswell and J. E. Mottershead. *Finite Element Model Updating in Structural Dynamics*. Vol. 38. Springer, Dordrecht, 1995. ISBN: 978-94-015-8508-8. DOI: <https://doi.org/10.1007/978-94-015-8508-8>. URL: <http://www.ncbi.nlm.nih.gov/pubmed/4450526>.
- [281] Tshilidzi Marwala. “Finite Element Model Updating Using Wavelet Data and Genetic Algorithm”. In: *Journal of Aircraft - J AIRCRAFT* 39 (2002), pp. 709–711. DOI: 10.2514/2.2985.
- [282] Tshilidzi Marwala. *Finite Element Model Updating Using Computational Intelligence Techniques: Applications to Structural Dynamics*. 2010. ISBN: 978-1-84996-322-0. DOI: 10.1007/978-1-84996-323-7.
- [283] Alexander Hardenberg, Arnold Kühhorn, and Maren Fanter. “Correlating and Updating Finite Element Models of Different Fidelity Using an Energy-Based Approach”. In: vol. Volume 10B. Turbo Expo: Power for Land, Sea, and Air. 2020. DOI: 10.1115/GT2020-14509. URL: <https://doi.org/10.1115/GT2020-14509>.
- [284] Dassault Systèmes. *Isight 2016: User’s Guide*. 2015.
- [285] Jeong-soo Park. “Optimal Latin-hypercube experiments”. In: *Journal of Statistical Planning and Inference* 39.1 (1994), pp. 95–111.
- [286] R. C. Jin, Wei Chen, and Agus Sudjianto. “An Efficient Algorithm for Constructing Optimal Design of Computer Experiments”. In: *J. Stat. Plan. Inference* 134 (2005), pp. 268–287. DOI: 10.1016/j.jspi.2004.02.014.
- [287] Bart Husslage et al. “Space-filling Latin hypercube designs for computer experiments”. In: *Optimization and Engineering* 12 (2006), pp. 611–630. DOI: 10.1007/s11081-010-9129-8.
- [288] The HDF Group. *Hierarchical data format version 5*. URL: <http://www.hdfgroup.org/HDF5> (visited on 06/21/2021).
- [289] CSIRO’s Data61. *StellarGraph Machine Learning Library*. <https://github.com/stellargraph/stellargraph>. 2018.

- [290] Federico Monti et al. “Fake News Detection on Social Media using Geometric Deep Learning”. In: (2019), pp. 1–15. ISSN: 2331-8422. arXiv: 1902.06673. URL: <http://arxiv.org/abs/1902.06673>.
- [291] Thomas N. Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (2017), pp. 1–14. arXiv: 1609.02907.
- [292] Amel Ben Mahjoub and Mohamed Atri. “An efficient end-to-end deep learning architecture for activity classification”. In: *Analog Integrated Circuits and Signal Processing* 99.1 (2019), pp. 23–32. ISSN: 15731979. DOI: 10.1007/s10470-018-1306-2.
- [293] In Kwon Lee and Ku Jin Kim. “Shrinking: Another method for surface reconstruction”. In: *Proceedings - Geometric Modeling and Processing 2004* 2 (2004), pp. 259–266. DOI: 10.1109/gmap.2004.1290047.
- [294] Yang Liu, Huaiping Yang, and Wenping Wang. “Reconstructing B-spline curves from point clouds - A tangential flow approach using least squares minimization”. In: *Proceedings - International Conference on Shape Modeling and Applications, SMI'05* 2005 (2005), pp. 4–12. DOI: 10.1109/SMI.2005.39.
- [295] B. I.N. Dong and Zuwei Shen. “Curve Reconstruction from Unorganized Points”. In: (), pp. 1–14.
- [296] Wenping Wang, Helmut Pottmann, and Yang Liu. “Fitting B-spline curves to point clouds by curvature-based squared distance minimization”. In: *ACM Transactions on Graphics* 25.2 (2006), pp. 214–238. ISSN: 07300301. DOI: 10.1145/1138450.1138453.
- [297] Ulrich Bauer and Konrad Polthier. “Parametric Reconstruction of Bent Tube Surfaces”. In: November 2007 (2008), pp. 465–474. DOI: 10.1109/cw.2007.59.
- [298] Ulrich Bauer and Konrad Polthier. “Generating parametric models of tubes from laser scans”. In: *CAD Computer Aided Design* 41.10 (2009), pp. 719–729. ISSN: 00104485. DOI: 10.1016/j.cad.2009.01.002. URL: <http://dx.doi.org/10.1016/j.cad.2009.01.002>.
- [299] Amal Dev Parakkat and Ramanathan Muthuganapathy. “Crawl through neighbors: A simple curve reconstruction algorithm”. In: *Eurographics Symposium on Geometry Processing* 35.5 (2016), pp. 177–186. ISSN: 17278384. DOI: 10.1111/cgf.12974.

- [300] Pengcheng Jia. “Fitting a Parametric Model To a Cloud of Points Via Optimization Methods”. In: *Dissertations - ALL* June (2017). URL: <https://surface.syr.edu/etd/673>.
- [301] Lili Cheng et al. “DeepPipes: Learning 3D pipelines reconstruction from point clouds”. In: *Graphical Models* 111 (2020). ISSN: 15240703. DOI: 10.1016/j.gmod.2020.101079.
- [302] Sagi Katz and Ayellet Tal. “Hierarchical mesh decomposition using fuzzy clustering and cuts”. In: *ACM Transactions on Graphics* 22.3 (Oct. 2003), p. 954. ISSN: 07300301. DOI: 10.1145/882262.882369.
- [303] Rong Liu and Computing Science. “Spectral mesh segmentation”. In: *Techniques* (2009). URL: <http://ir.lib.sfu.ca/handle/1892/112>.
- [304] Ariel Shamir. *Segmentation and Shape Extraction of 3D Boundary Meshes*. Tech. rep. 2006. URL: <https://www.researchgate.net/publication/266493778>.
- [305] Evangelos Kalogerakis et al. “3D Shape segmentation with projective convolutional networks”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. Vol. 2017-Janua. 2017, pp. 6630–6639. ISBN: 9781538604571. DOI: 10.1109/CVPR.2017.702. arXiv: 1612.02808.
- [306] Aleksey Golovinskiy and Thomas Funkhouser. “Randomized cuts for 3D mesh analysis”. In: *ACM SIGGRAPH Asia 2008 papers on - SIGGRAPH Asia '08*. New York, New York, USA: ACM Press, 2008, p. 1. ISBN: 9781450318310. DOI: 10.1145/1457515.1409098. URL: <http://dl.acm.org/citation.cfm?doid=1457515.1409098>.
- [307] Halim Benhabiles. “3D-mesh segmentation: automatic evaluation and a new learning-based method”. PhD thesis. Université des Sciences et Technologie de Lille, 2013, p. 160. URL: <https://tel.archives-ouvertes.fr/tel-00834344>.
- [308] Yu Kun Lai et al. “Rapid and effective segmentation of 3D models using random walks”. In: *Computer Aided Geometric Design* 26.6 (Aug. 2009), pp. 665–679. ISSN: 01678396. DOI: 10.1016/j.cagd.2008.09.007.
- [309] Martin Reuter. “Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions”. In: *International Journal of Computer Vision* 89.2-3 (Sept. 2010), pp. 287–308. ISSN: 09205691. DOI: 10.1007/s11263-009-0278-1.

- [310] Oscar Kin Chung Au et al. “Mesh segmentation with concavity-aware fields”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.7 (2012), pp. 1125–1134. ISSN: 10772626. DOI: 10.1109/TVCG.2011.131.
- [311] Mukulika Ghosh et al. “Fast approximate convex decomposition using relative concavity”. In: *CAD Computer Aided Design*. Vol. 45. 2. Feb. 2013, pp. 494–504. DOI: 10.1016/j.cad.2012.10.032.
- [312] Zhige Xie et al. “3D shape segmentation and labeling via extreme learning machine”. In: *Eurographics Symposium on Geometry Processing* 33.5 (2014), pp. 85–95. ISSN: 17278384. DOI: 10.1111/cgf.12434.
- [313] Simon Christoph Stein et al. “Object partitioning using local convexity”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Sept. 2014, pp. 304–311. ISBN: 9781479951178. DOI: 10.1109/CVPR.2014.46.
- [314] Huijuan Zhang et al. “Shape segmentation by hierarchical splat clustering”. In: *Computers and Graphics (Pergamon)* 51 (July 2015), pp. 136–145. ISSN: 00978493. DOI: 10.1016/j.cag.2015.05.012.
- [315] Minghui Gu et al. “An improved approach of mesh segmentation to extract feature regions”. In: *PLoS ONE* 10.10 (Oct. 2015). ISSN: 19326203. DOI: 10.1371/journal.pone.0139488.
- [316] Azza Ouled Zaid, Meha Hachani, and Raoua Khwildi. *Segmentation of 3D articulated meshes using the shape diameter function and curvature information*. Tech. rep. 2016. DOI: 10.1109/ICME.2016.7552943.
- [317] Peng-Shuai Wang et al. “O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis”. In: *ACM Transactions on Graphics* 36.4 (July 2017), pp. 1–11. DOI: 10.1145/3072959.3073608.
- [318] Li Yi et al. “SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. Institute of Electrical and Electronics Engineers Inc., Nov. 2017, pp. 6584–6592. ISBN: 9781538604571. DOI: 10.1109/CVPR.2017.697.
- [319] Weihua Tong et al. “Spectral Mesh Segmentation via l0 Gradient Minimization”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.4 (2020), pp. 1807–1820. ISSN: 19410506. DOI: 10.1109/TVCG.2018.2882212.

- [320] Benjamin Spiess, Klaus Höschler, and Maren Fanter. “Development of a Simulation-based Knowledge Representation for the Simplification of Structural Aero Engine Components”. In: NAFEMS World Congress 2021, Oct. 2021. URL: https://www.nafems.org/publications/resource_center/nwc21-234/.
- [321] Benjamin Spiess, Klaus Höschler, and Maren Fanter. “A Feature-Based Approach for an Automated Simplification of Structural Aero Engine Components”. In: *Deutscher Luft- und Raumfahrtkongress 2020*. Online: Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V., 2020. DOI: 10.25967/530050.
- [322] Benjamin Spiess, Klaus Höschler, and Maren Fanter. “Using a Smart Recognition Framework for the Automated Transfer of Geometry to Whole Engine Mechanical Models”. In: *WCCM-ECCOMAS2020*. Scipedia, 2021. DOI: 10.23967/wccm-eccomas.2020.354.