www.cad-journal.net
Computer-Aided Design AND APPLICATIONS

A Review of Challenges in Geometry Handling and Mesh Generation for Computer-Aided Design and Engineering

Bowen Liu [1], Yuanxing Lv [2], Kaiting Li [3], Beiyan Jia [4] and Liang Sun [5*]

[1,2,3,4,5]Dalian University of Technology, Dalian, Liaoning, 116000, China
[1]1123494295@mail.dlut.edu.cn, [2]Lv_yuanxing@mail.dlut.edu.cn,
[3]likaiting@mail.dlut.edu.cn,[4]32204229@mail.dlut.edu.cn,[5]lsun02@dlut.edu.cn

Corresponding author: Liang Sun, lsun02@dlut.edu.cn

Abstract. Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) have significantly transformed the product design and analysis workflow. Despite these advancements, the process of transitioning a complex CAD model into a suitable CAE analysis model remains highly time-consuming and labor-intensive. Challenges exist at different stages of the whole process. This paper provides a review of the key challenges of geometry handling and meshing in the current CAD and CAE processes. It intends to provide the reader with a better understanding of the issues commonly encountered, which enables the first step forward to solving them.

Keywords: Geometry parameterization; Geometry handling; Mesh generation and manipulation; Mesh to CAD; CAD/CAE integration.
DOI: https://doi.org/10.14733/cadaps.2026.123-146

1 INTRODUCTION

Finite element analysis (FEA) or computational fluid dynamics (CFD) have been widely used as effective methods to numerically validate the performance of a designed product, reducing the need for expensive physical testing. The development of computational capability has made simulation workflows more accessible, with an exponential increase in the number of analyses performed over the last few years. At the same time, to meet governmental and societal demands for cleaner, quieter, and safer transport, the level of complexity of simulations has also increased significantly. This is reflected in the demands for simulations at a larger model scale, with a higher model fidelity, and in multi-disciplines.

As the demand for high-fidelity simulation increases, the scale and complexity of the models have also grown considerably. A product such as an aeroplane, an aero-engine, or an automotive vehicle contains hundreds or thousands of key parts. With the continuous increase of computational capabilities, the level of scale of many analyses, such as fan-blade-off or bird strike analysis of an aero-engine, rises from an individual part to a component level, and eventually to the whole assembly level to achieve more accurate simulation results [1], as shown in Figure 1(a). Setting up a large-scale assembly model for simulation is an extremely time-consuming process and requires significant skills from engineers. Not only does the time and effort required increase with each additional component, but extra efforts are required to properly specify interfaces e.g., to ensure a conformal mesh at interfaces. The increase of the model scale also results in a huge increase in the number of degrees of freedom (DOF) in the analysis models, which can reach tens of millions for fan-blade-off analysis of an aero-engine for example [2], [3]. Reducing the DOF of analysis models to achieve a good balance between accuracy and computation time is still an ongoing challenge.

In addition to increased scale, simulations are also performed at a higher fidelity. The term "fidelity" has been used to describe the extent to which the computer-simulated virtual system matches the physical system. The level of fidelity required depends on the purpose of a design and its maturity, with preliminary designs using low-fidelity models to quickly determine a configuration that satisfies basic requirements, and final detail design using high-fidelity models for validation. In terms of geometry description, a high-fidelity model contains more geometry details and includes most design features (e.g., holes, blends, bosses, etc..). Figure 1(b) shows examples of a component meshed using 3D hexahedral (hex) /tetrahedral (tet) mesh elements, such as are used for detailed design, and the same model represented using 1D beam/2D shell elements as are used for earlier design stages. Applying simulation tools to more complex models while maintaining accuracy requires an increased level of fidelity, which in turn brings a significant increase in the size of the analysis model, as well as the computational time.

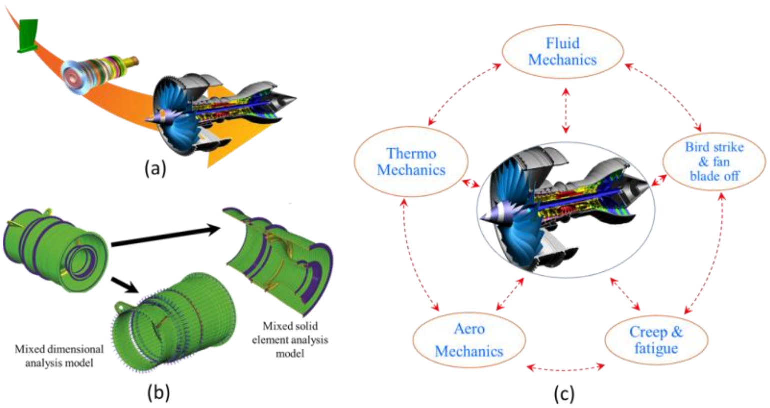

Figure 1: (a)the scale of simulations has increased from part level to component level and to the whole assembly level; (b) a low fidelity model with beam-shell elements and a high-fidelity model with mixed 3D solid elements; (c) an example of the multi-disciplinary considerations in designing an aero-engine.

Beyond scale and fidelity, another key trend in simulation is the integration of multiple physical disciplines into a unified workflow. To evaluate the in-service performance of a complex design, different disciplines need to be included in the same simulation workflow e.g., structural analysis (static, impact, vibration, fatigue, etc.), thermal analysis, fluid analysis, acoustics analysis, as shown in Figure 1(c). In a loose-coupled situation, these analyses are performed individually with the results gathered separately by experts, who then make appropriate decisions based on their experience. This approach fails to capture the interactions between the different disciplines, which makes design optimisation difficult. To obtain results that are more representative of reality, a multi-disciplinary simulation, where different analyses are coupled, is used to capture the interaction between the different physics being investigated. Multi-disciplinary simulation is identified by NASA as one of the key research areas that affect the realization of the CFD analysis of complex models and worth further investigation [4]. Some related reviews and surveys can be found in [5]–[7]. Beyond increasing the complexity of the model setup, multi-disciplinary simulation integration also requires the capability to map data between the different disciplines accurately and efficiently.

These trends toward larger, more detailed, and more integrated simulations have placed increasing demands on the underlying preprocessing steps. Among them, geometry handling and meshing have become major bottlenecks. Geometry handling and meshing are extremely time-consuming processes that involve intensive manual effort, particularly for large and complex models. In [8], it shows that a whole aero engine model contains 200M degree of freedoms and requires a preparation time of six months. According to a survey by the US Sandia National Labs [9], this process accounts for approximately 70% of the overall time. A recent study from NASA [10], as well as a subsequent conference paper [11], both highlight that geometry and mesh generation of complex models are major bottlenecks that require significant improvement. Any reduction in the time required for these processes would free up more time to focus on value-adding aspects of design, analysis, or testing.

In response to these challenges, various research efforts have been made to improve mesh generation and geometry processing techniques. Lei et al. [12] provided a comprehensive review of mainstream algorithms for generating triangular, quadrilateral, tetrahedral, and hexahedral meshes, covering both structured and unstructured methods. Pietroni et al. [13] conducted an in-depth survey of recent advances in hexahedral mesh generation and optimization, encompassing various approaches such as frame-field methods and block decomposition, and proposed hex-dominant meshing as a future trend. Zhao et al. [14] carried out an in-depth study on the integration of artificial intelligence into CAE workflows, demonstrating how AI technologies can be embedded into traditional numerical simulation processes to address challenges such as geometric complexity, low computational efficiency, and limited accuracy. While existing reviews have made significant contributions, most of them focus on a specific stage of the workflow, such as geometry processing or mesh generation. In contrast, this paper provides a broader overview that spans multiple stages from CAD to CAE, aiming to provide a more complete workflow analysis.

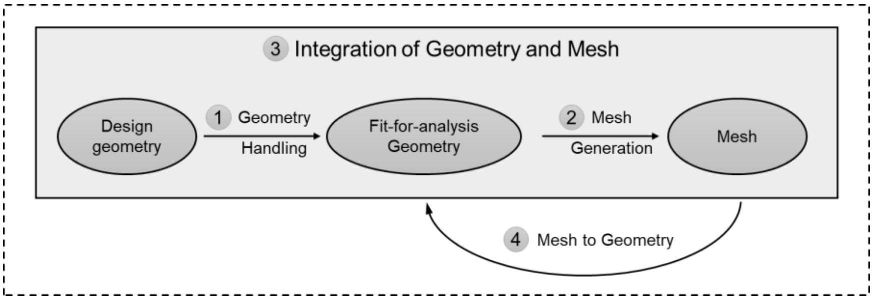

Figure 2: Four processes of geometry handling and meshing are discussed in this paper.

In this work, four major processes in CAD/CAE preprocessing are discussed: geometry handling and manipulation, mesh generation, integration of different models, and mesh-to-CAD associativity, as shown in

Figure 2. The paper serves as a technical review, aiming to summarize the key challenges related to these processes and provide the reader with a comprehensive understanding of the current issues in the field. The remainder of the paper is organized as follows: Section 2 summarizes challenges of geometry handling and manipulation. Section 3 identifies the challenges related to mesh generation and editing; Section 4 discusses challenges of integrating different models; Section 5 identifies

challenges of obtaining a CAD from a mesh; Section 6 reviews existing AI based solutions to geometry handling and mesh generation; finally, Section 7 presents the conclusions.

2 CHALLENGES OF GEOMETRY HANDLING AND MANIPULATION

In order to fully harness the benefits of FEA/CFD analysis, it is imperative to have a CAD model that accurately reflects the design intent from a simulation perspective and is well-suited for analysis. The level of detail and abstraction in this model should be carefully balanced to achieve accurate results without excessive analysis time. Geometry handling and manipulation involves several tasks, including geometry healing, geometry clean-up, de-featuring, geometry decomposition, and dimensional reduction, as shown in Figure 3. The general challenges for geometry handling include, but are not limited to, ensuring automatic, robust and high-quality geometry operations while maintaining links to original CAD model. In the following, some of the challenges associated with these operations are summarized and explained.

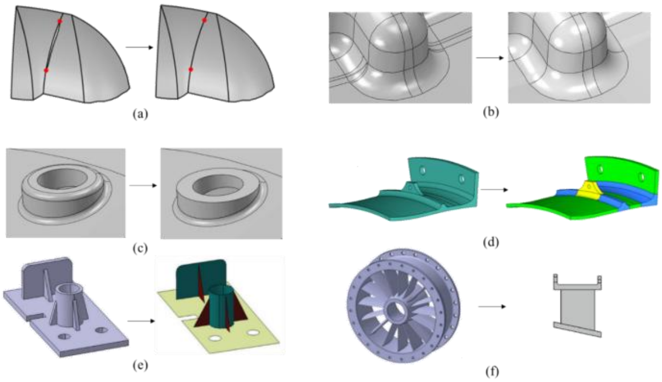

Figure 3: Examples of geometry handling and manipulation: (a) non-watertight model and healed B-Rep; (b) example of model simplification; (c) example of removal of rounded corners; (d) example of model decomposition; (e) mid-surface model of thin-wall model [15]; (f) dimensional reduction of axisymmetric 3D geometries [16].

2.1 Automatic Detection and Fix of Geometrical and Topological Issues

The purpose of geometry healing is to remove various defects to ensure that the CAD model is valid topologically and geometrically. Geometry clean-up aims to eliminate undesirable entities such as sliver faces or short edges that may hinder the mesh quality. Unintended geometrical or topological errors, such as cracks, degeneracies, duplicate/redundant entities, self-intersection and overlaps can happen, especially for a translated model in neutral format, such as .igs or .stp. Major reasons that cause these errors include:

- Errors due to bad or wrong operation by the CAD modeller
- Different CAD packages have different mathematical representations for the same model
- Translation and export between packages with different support for data exchange
- Different geometric tolerances in CAD packages

Automatic detection of issues in the model is a critical step to ensure the accuracy and consistency of geometric models, and also a key challenge that requires a balance of accuracy and efficiency. During the model conversion, the modelling history is lost, and the only information available is the dumb model. The well-known geometry fix software CADfix classifies the issues into seven groups, i.e., connectivity, unused, topology, integrity, complexity, continuity, and sloppiness. Each group contains a list of issues. Topological issues require a top-down or bottom-up topological check to make sure the bounding and bounded entities are valid. Geometrical issues normally require an input of user-defined tolerance, the misalignment within or beyond which is considered an issue. For complex geometries, it could be a challenge to get the best tolerances at the first try. The detection algorithm is expected to be intelligent enough to automatically distinguish between genuine issues and valid models that incidentally meet the defined tolerance. Efficiency is also a challenge for automatic detection. Some issues are local and needs a local analysis to identify, such as narrow region or intersection. However, perform such detection for every region could be extremely time-consuming.

Three primary approaches are commonly used to prepare models for meshing: direct CAD modifications, transitioning to discrete representations, and "virtual topology"-based methods. There are also dual methods that use more than one of the above-mentioned methods together. Directly modification of the geometry involves creation of new NURBS curves or surfaces. The benefit is that it exports a geometry directly, which benefits meshing, especially in a different software package. However, this approach increases the burden on the analyst and requires familiarity with fundamental geometry operations. In some situations, within a specified tolerance, it is difficulty to generate a geometry which also satisfy the boundary continuity requirements. The link to the original CAD is also cut-down, which means repeated operations are needed when design model changes. While this approach remains widespread, it also requires significant user input and CAD modelling experience, and may produce unexpected results (invalid features, persistent naming issues [17][18], sliver entities, etc.)

Another approach is to discretize the geometry to a tessellated lightweight mesh [19]–[21]. Once discretized, facet manipulation can be used to edit the boundary of the shape, with operators available to get rid of small details as presented in [22]. Operating on discretized geometry negates the need to manipulate complex surface definitions. However, since the mesh only approximates the geometry, it needs to be careful to make sure that the result is a faithful representation of the geometry of the design. It breaks the associativity with the original CAD model making the propagation of design updates between the representations difficult. Moreover, dedicated algorithms are required for the different geometry handling operations, as algorithms based on the topology of the B-rep geometry and based on features cannot be used directly on a discretized model.

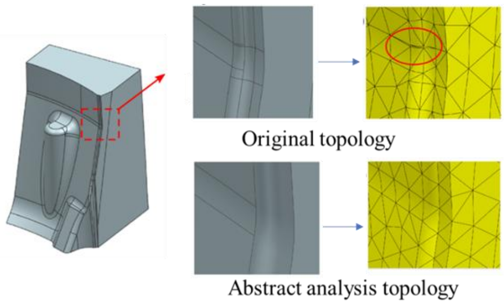

Figure 4: Geometry clean-up using virtual topology by merging faces [23].

Virtual topology enables the topological definition of the B-Rep of a model to be uncoupled from its geometrical definition [24]–[26], allowing editing the bodies, faces, and edges of a model without modifying the underlying surfaces and curves. A greater level of freedom in mesh node location is achieved after virtual topology has been used to remove topological entities that are a product of the construction of the model, but are not relevant for analysis, as shown in Figure 4. When using virtual topology, the link between a CAD design model and the equivalent analysis model prepared using virtual topology can be maintained, allowing for more automation of the pre-processing workflow. However, to benefit from virtual topology, the downstream meshing tool needs to be compatible with virtual entities and infer the required geometric information from the underlying geometry. In the absence of a standard for exchanging virtual topology operators, virtual topology implementations in commercial packages often lack flexibility. It is often impossible to access the history or edit virtual topology operations, and the user has to manually re-apply the operation from scratch after a change of design.

2.2 Intelligent Decision Making of De-feature and Automatic Operation

After the geometry healing and clean-up, it is determined which features to be removed or retained for each component in the assembly. This decision-making process is driven by the ability to capture the correct physics and the need to control the size of the analysis model. It has a considerable impact on the complexity of the final analysis model and on the overall time for model preparation and analysis. This challenge is further compounded by the intricate and detailed nature of components represented using CAD models. As highlighted in the study by [25], the transformation of these highly detailed CAD models into Finite Element (FE) models necessitates a careful balance between retaining essential geometric information and simplifying the model for efficient simulation.

Classifying features as critical or non-critical is an extremely time-consuming process, especially for multi-physics analyses [27]. Multi-disciplinary prefers to have the same idealised model that can be used for all disciplinary so that results can be easily interpolated from one model to another model. Using distinct idealized models can lead to scenarios where results must be extrapolated or where nodes from one model fall outside the mesh domain of another. Both scenarios can have a detrimental impact on the accuracy of the analysis. For these reasons compromises must be made between different disciplinary to produce a common idealised model, i.e., bosses may need to be retained for the thermal analysis while certain fillets may need to be retained for the structural analysis. It is clear that the idealisation decisions making process take more time than the actual feature removal operations.

De-featuring techniques have been extensively researched with many approaches. If the modelling history is available, an expression-based method can be used for feature removal. Features for removal can be tagged appropriately and expressions can then be used to automate this process. Modifying the feature tree can often cause errors when parent features are removed. It is undesirable to make modifications to the feature tree and expose the analyst to the complexity of the construction history. If only the dumb geometry is available, features need to be identified first based on its geometrical and topological characteristics [28]. the accuracy of feature identification is critical and accurate algorithms need to be used to avoid misidentifying or missing key features. Removing the features normally involves operations such as surface extension and trimming. The accuracy of feature removal is a key factor to maintain a valid CAD model, avoiding problems such as overhanging vertex/edge or self-intersections. Technology such as direct modelling can be used. However, the use of these idealisation tools are currently manual operations carried out by the analyst, which counteracts the desired automation of the analysis process. Therefore, efficient algorithms need to be used to reduce pre-processing time and improve computational efficiency. The compatibility of feature removal with subsequent processing also needs to be considered to avoid any influence on the FEA results, thus ensuring the accuracy and reliability of the calculation results.

2.3 Automatic and Intelligent Decomposition and Dimensional Reduction

Decomposition refers to breaking down complex geometric shapes into simpler sub-regions, as shown in Figure 5, in order to facilitate the generation of models suitable for meshing, especially for hex meshing. Various methods for automating this process have been developed [29]–[31]. Currently, the fully automated decomposition process only applies to some specific types of geometries, such as sweepable volume and thin-walled models. More generalized methods combining multiple techniques, such as edge concavity, geometry reasoning, symmetry, etc., are still expected to accelerate the processing of complex models. The manifold representation used in most CAD packages implies that two bodies cannot share a face. Hence, no unique interface is created between adjacent sub-regions, and recovering the interface information can be difficult, especially after CAD export. It is difficult to update the decomposition to propagate a design change, as any invalid split operation during the re-computation of the B-Rep will make all the subsequent splits invalid. Split operations in CAD can be error-prone for some configurations (e.g., configuration that would create a non-manifold subset) and tolerant modelling may create some small gaps when partitioning surfaces with a complex geometric definition. As a result, recombining the different sub-regions into a single conformal mesh can be very challenging.

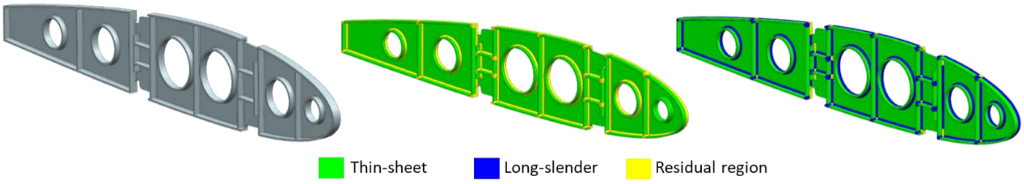

Figure 5: Example of thin-sheet and long-slender decomposition for a rib model.

Dimensional reduction aims to exploit idealization strategies to reduce the geometric representation of a model, e.g., to reduce a thin-walled model to its mid-surfaces. Mid-surfaces are created by generating a mean surface in the middle between two opposing faces of a solid model. Mid-surfaces can be used to generate meshes of 2D cells, thus reducing computational effort and memory consumption. There are various methods, including the face pair method [32], chordal axis transform method [33] and the medial axis transform-based method [34]. Automatic mid-surfacing is now available in most CAD and CAE software; however, some challenges remain for complex models, in particular on the interpretation of the junction between several mid-surfaces with varying thicknesses. In Figure 6 different treatments for a junction between 3 mid-surfaces are presented. These regions, along with regions for which local features such as bosses have been removed, are prone to underestimating or overestimating stiffness depending on the handling of the junction.

Mid-surface models are also unable to predict the normal stress variation along the intersection of two shell structures due to the zero stress assumption, along the normal direction [35]. It is therefore common practice to use mixed-dimensional models where idealized regions are used to provide accurate loadings, while the region of interest is modelled in 3D. While mixed-dimensional modelling improved the capability to model the complex features [34]. It still requires careful treatment of the 3D to 2D coupling. Medial axis/chord axis-based methods are also limited by the robust generation of the medial axis/chord axis.

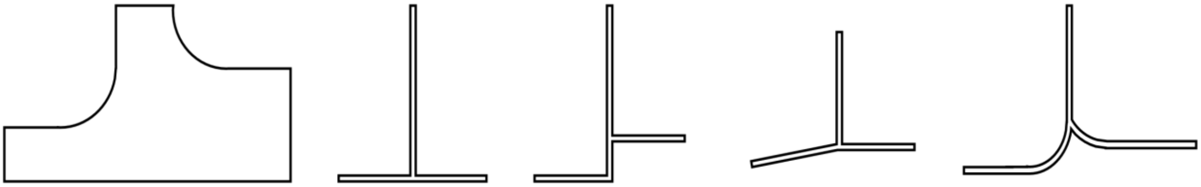

Figure 6: Different treatments of mid-surface junction. Adapted from [36].

2.4 Properly Tracing the Topology Changes of CAD Models During Optimization

With the development of computing capability, the CAD model plays a central role in the context of model-based engineering [37]. As stated in [38], high-fidelity shape optimization requires the input and output to be both CAD models to enable the integration between CAD, CAE and CAM. The multi-disciplinary analysis would benefit from having one common parameterized model to facilitate the link across different disciplines. Feature-based CAD modelling is widely used in different disciplines and is good for the integration of CAD, CAE and CAM, which makes it a great candidate for MDA or MDO.

However, there are still many limitations to the use of feature-based CAD, e.g., complex topology. Besides this, the boundary topology of the model may change during the optimization. This can occur when the underlying surface definition of the boundary is modified to accommodate a change of parameter, as shown in Figure 7(a) where a new face is introduced. There are a few identified cases where this happens, e.g., to maintain a certain level of continuity or to avoid a large curvature. This can also occur when features are clashing after a parametric perturbation. For example, in Figure 7(b), during the optimization of an aircraft, the position of the wing-fuselage intersection moves across the boundary of a face. As a result, the number and/or label of faces and edges changes and the one-to-one correspondence between these entities before and after the parametric perturbation is lost.

During an optimization process, when the CAD model's parameter values change, causing a shape change, it is necessary for the analysis mesh to be updated correspondingly. One way to achieve this is through mesh regeneration, which can be computationally expensive and may not even be possible for structured meshes. A preferable approach is for mesh deformation to be used, where the existing mesh is modified to match the shape of the updated CAD model. The above-mentioned loss of correspondence between entities before and after the parametric perturbation will block the process of mesh deformation. To keep this correspondence, it is required that the mapping of the two models can be automatically and dynamically build.

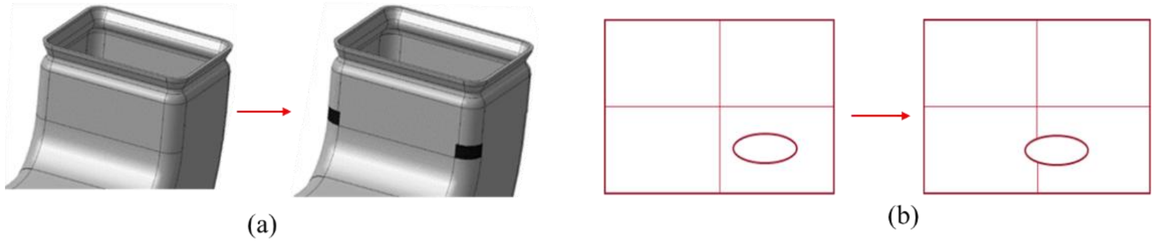

Figure 7: Examples of changes in model topology structure: (a) new faces appear after parameter change; (b) fuselage-wing intersection move across the boundary of a face.

3 CHALLENGES OF MESH GENERATION

Mesh generation plays a pivotal role in bridging the gap between CAD models and simulation. It is a crucial pre-processing step that directly impacts simulation accuracy, efficiency, and robustness. While the topic has been extensively studied for decades [13], [39], significant challenges remain—particularly in generating high-quality structured quad and hex meshes and in preserving the link between the mesh and its underlying CAD geometry.

3.1 Automatic High-quality Quad and Hex Mesh Generation

Despite automatic algorithms being available for the 2D triangle (tri) and 3D tetrahedral (tet) mesh generation, there remains a strong demand for automatic quad and hex mesh generation due to the relatively improved efficiency and accuracy of their solutions. A quad/hex mesh can be classified as a structured mesh, an unstructured mesh or a block-structured mesh based on the internal connectivity of its elements, as shown in Figure 8. In a block-structured mesh, the domain is decomposed into a collection of simple blocks and adjacent blocks join at nodes/edges of irregular connectivity, which are referred to as singularity points/lines. A block-structured mesh usually offers a higher quality and boundary-aligned elements with a minimum number of mesh singularities, thus providing a compromise between the simplicity of a structured mesh and the flexibility of an unstructured mesh. The macro-unstructured-micro-structured layout also naturally lends itself to parallelization, where the execution of calculations on local groups of blocks is performed on parallel processors.

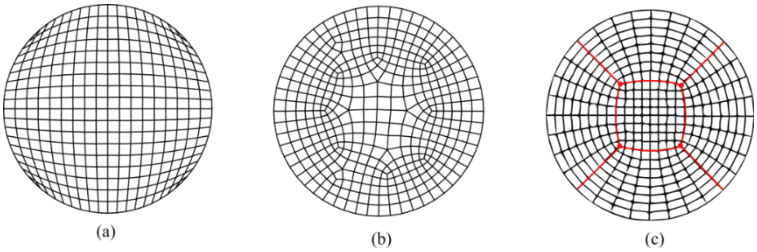

Figure 8: Quad meshes categories: (a) a structured quad mesh; (b) an unstructured quad mesh; (c) a block-structured quad mesh, the block structures are shown in red lines and the singularity points are shown in red points.

The industry-standard method of creating multi-block meshes for complex shapes is to manually create blocks using dedicated tools such as ICEM [40] and then associate block edges and vertices with the geometry. This is a time-consuming process and requires a great deal of knowledge and expertise from the analyst. The main challenge of block-structured mesh generation is to determine the number of singularities required and where to place them. There have been a few methods to achieve this using e.g. cross-field based methods and Periodic Global Parameterization [41][42]. However, these methods still have not been integrated into commercial software. One difficulty is narrow and degenerated blocks, or an infinite spiral split line, as shown in Figure 9, might be generated and need careful handling.

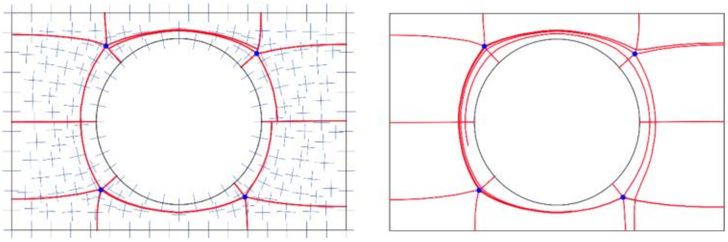

Figure 9: Block decomposition results of a symmetric example with a circular hole: (a) cross-field method where the split line stops when it first meets another split line; (b) cross-field method with the split lines shown in (a) continuing. There are degenerated blocks, infinite spiral streamlines, and the block result is not exactly symmetric.

The underlying mathematical theory of the cross-field method is conformal mapping, which keeps the angle value unchanged during the mapping, but not the length. Consequently, size constraints are not well handled. Generating a high-quality mesh robustly that adheres to the specified sizing constraints applied to the CAD model's boundary remains a challenge. Finally, a complex CAD model can contain too many details and need to be cleaned using a technique such as virtual topology during the mesh generation process of the cross-field method, as shown in Figure 10. Therefore, the mesh generator needs to support the usage of virtual topology. This means one block could step over more than one geometric surface. The relationship between the initial CAD model, the cleaned CAD model, the block model, and the mesh model needs to be built and stored so information can be transferred between the different models.

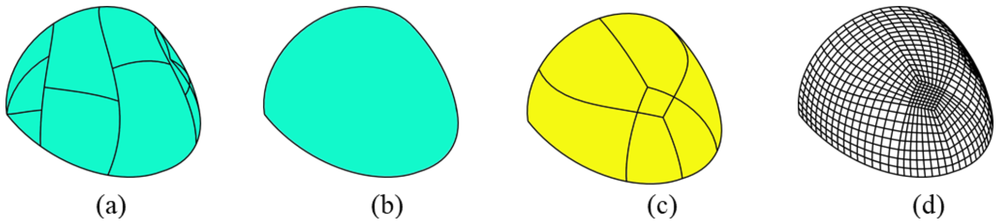

Figure 10: Block structure mesh generation using the cross-field method: (a) initial CAD model; (b) model cleaned using virtual topology; (c) block decomposition; (d) resulting quad mesh.

Hex mesh generation has been claimed to be the holy grail in the mesh generation field [43]. It shares a lot of similarities with quad mesh generation. However, none of the attempts made to extend quad meshing algorithms in 3D have resulted in a robust and general solution. The level of difficulty for hex mesh generation is much higher than quad mesh generation, and automatic hex mesh generation is still limited to a very narrow range of simple geometries. Recently, frame-field methods, which are an extension of 2D cross-field methods [44] [45], have been applied for hex meshing. However, additional theoretical understanding is still needed to provide sufficient support to obtain hex meshes from the frame field. Currently, there are no guarantees that the generated frame field will lead to a valid hex mesh structure. A commonly seen example is the merging of a 3-valence singularity line and a 5-valence singularity line, as shown in Figure 11, which will result in an undesired and invalid mesh topology. Although there are attempts to solve this, they are all heuristic-based methods [46].

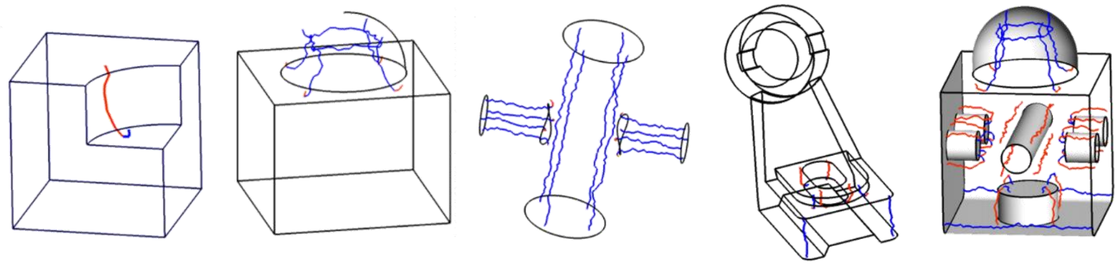

Figure 11: Examples of non-meshable 3-5 singular curves [46].

Complex CAD models used in industry also feature geometries that are difficult to capture with frames, such as an edge with a varying dihedral angle, or a vertex with a high valence (>3 edges connected), as shown in Figure 12. Dirty geometry and complex features (e.g., the intersection of blends) increase the difficulty of hex mesh generation and require geometric handling to reach a good balance between geometric accuracy and mesh quality. Finally, the computational efficiency of the method has to be taken into account, as frame fields rely on intermediary constructs such as a tetrahedral mesh for identifying frame orientations, adding to the cost of the method. Considering features at different sizing scales, it is necessary to generate an adaptive mesh, which further adds to the computational demands.

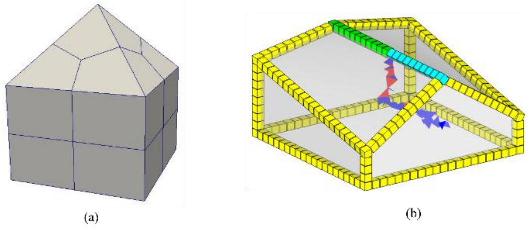

Figure 12: Example with hard-to-capture features: (a) a vertex of a high valence; (b) an edge with a varying dihedral angle.

While significant efforts are being made to improve automatic quad and hex mesh generation, it is equally important to ensure that the generated meshes remain tightly associated with their geometric origins. The next section discusses the challenges and requirements related to maintaining this mesh-CAD associativity.

3.2 Maintaining The Mesh and CAD Associativity

The associativity between the mesh and the CAD model, e.g., which mesh node/element lies on which geometric entity and its position, is usually lost after the mesh is generated, e.g., information such as which mesh node/element lies on which geometry entity and its location on it. This associativity is, however, required in many cases, such as:

- High-order curved elements offer a better approximation of the geometry compared to the polynomial linear elements. When generating a high-order curved mesh, it is necessary to obtain geometric information such as curve slopes, surface curvatures and surface derivatives at a point [47].
- When performing mesh adaption to improve the quality of the elements, e.g., h-type refinement, geometric information from the CAD system is required at each iteration. Hughes suggested that the lack of the link between the mesh and the CAD could be the reason why “adaptive refinement is still primarily an academic endeavour rather than an industrial technology” [48].
- Mesh smoothing, which is used to improve the mesh quality without changing the mesh topology, requires geometric information to determine the new location of moved nodes [49].
- Mesh updating and re-meshing. Propagating design updates are common in early design stages. In the absence of a link between the original CAD model and the mesh, mesh updating will be challenging. This often means that all the geometry handling steps have to be re-applied which is a difficult task to automate, especially if the topology of the CAD model has changed. With the links preserved, the mesh can be rebuilt fast after a local change.

These examples collectively highlight that without robust mesh-CAD associativity, adaptive refinement, geometry updates, and high-order meshing become prohibitively complex. Therefore, preserving this link is not merely beneficial but essential for an efficient simulation workflow.

4 CHALLENGES OF INTEGRATION OF DIFFERENT MODELS

Building upon the previous discussion of geometry handling challenges, this chapter focuses on the issues that arise when integrating multiple geometric and mesh models within complex simulation workflows. In practice, different representations of geometry and mesh are often required for different analysis domains or fidelity levels, which makes consistent data integration a critical yet unresolved challenge.

In multi-physics and multi-fidelity analysis, various geometry and mesh models are derived from a central master design model. The geometry model might result from different geometry handling strategies for different analyses and correspondingly the mesh models are of different types. The need for closer CAD-CAE integration dates back to 1991 [50], and this has been a topic of interest ever since. Although extensive progress has been made [51], [52], there remains a strong need for a more generic integration into one combined strategy.

For example, in a multi-physics analysis, a tetrahedral mesh may be sufficient for thermo-mechanical analysis of an aero-engine, whereas the structural analysis may require a fully hex meshed model. In a multi-fidelity analysis, a 3D solid mesh may be used for a high-fidelity analysis and a dimensionally reduced 2D mesh may be used for a low-fidelity analysis. One of the main challenges is the lack of mapping capabilities to transfer information between different representations and robustly link these representations. Moreover, local changes to the master CAD model cannot be reflected on other copies of the model automatically. Design modifications often require the analyst to significantly rework the analysis model by repeating pre-processing operations on the new geometry. These issues underline the importance of a unified integration framework that supports traceability, consistency, and automation across all model representations.

Several integration strategies have been proposed to address these challenges, which can generally be categorized into CAD-centric and CAE-centric approaches. CAE-centric methods require the parametric model to be available in the CAE environment, which sets a high requirement for the engineer and is challenging for complex models. With CAD geometry playing a more important role in product design, the CAD model is a good candidate to be the centre of the whole process. Nolan et al. [23] proposed the concept of defining simulation intent, which aims to capture high-level modelling and idealisation decisions to create an efficient and fit-for-purpose analysis. It uses the concept of Cellular Modelling to partition the space into ‘cells’ of simulation significance and uses the concept of Equivalencing to establish the links between entities in different design and analysis models that represent the same region of space. Figure 13 shows an example where different geometries and meshes of the same model are linked together through a database using the concept of simulation intent. The relationships between different models are stored in a central, generic data structure.

While the concept of defining simulation intent provides a mechanism to maintain links between different representations of a master model, generating the cellular model remains a bottleneck in the process. This cellular model is either obtained by capturing interfaces in assemblies or by decomposing the model and keeping partitioning faces as interfaces. In the first case, large assemblies (>1000 components) often contain small gaps or interference that prevent tools such as Parasolid from extracting all interfaces, and in the second case, only a few decomposition tools have been integrated with virtual topology to produce the necessary partitioning. The challenge is compounded because the majority of commercial CAD environments are manifold, whereas a cellular model should be non-manifold, requiring a conversion process that can encounter robustness issues.

Another well-known attempt to integrate CAD with FEA is the isogeometric analysis (IGA) concept. It was initially proposed by Hughes in 2005 [48] and has received a lot of attention and developments since then [54]–[56]. The terminology “isogeometric” comes from the fact that the solution space for a dependent variable is represented using the same functions representing the geometry. Traditional FEA relies on polynomials to approximate the geometry, which brings discretization errors and severes the link to the geometry, making mesh refinement tedious. When using IGA, refinement of the mesh can be easily achieved by using knot insertion while preserving the exact shape of the model (Figure 14). While IGA has shown many advantages over the traditional FEA, handling large deformations and representing complex assembly models with connections remains challenging. Research on IGA is still ongoing, and additional applications are expected to become possible with further development. From the integration point of view, IGA removes the need for a traditional meshing process, but still assumes that feature removal is performed before the mesh generation [48], which remains a time-consuming step.

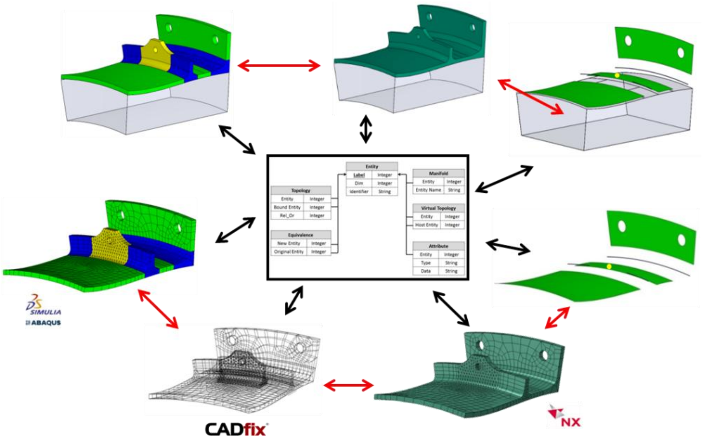

Figure 13: A set of "linked" geometry and meshes for a simple component [53].

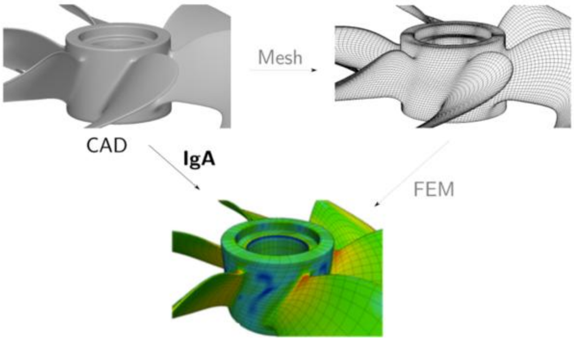

Figure 14: An example illustrating IGA [57].

5 CHALLENGES OF MESH-TO-CAD

While much of the geometry handling workflow has focused on converting CAD models into mesh representations for numerical analysis, the reverse operation—from mesh back to CAD—is gaining increasing importance in modern design and engineering. In several situations, such as design optimization, reverse engineering, and digital twin creation, the ability to recover an editable CAD model from discretized data becomes essential. However, existing geometry kernels and CAD systems provide limited support for this reverse pipeline, which presents its own set of challenges distinct from those in traditional CAD-to-mesh workflows. This section investigates different contexts in which CAD models must be re-created from discrete representations and the technical difficulties encountered in the process.

While geometry kernels and CAD software enable the building of complex model geometries and can robustly convert them to a discretized representation, their ability to create a CAD model from a discretized representation is either missing or very limited. However, there are various situations where this operation is necessary, including post-processing optimization results (topology optimization in Figure 15 (a) or shape optimization in Figure 15 (b)) to recover a usable CAD shape and geometry healing and simplification using a discrete method [58] as shown in Figure 15 (d). The development of digital engineering also creates an increasing need for recovering and integrating the scanned discretized model into the design and analysis workflows. Figure 15 (c) represents scanned data for reverse engineering or redesign of un-digitalized models [59]. This section presents different situations to re-create CAD models from discrete point clouds or meshes.

Converting the initial geometric information captured via a discretised mesh or point cloud back to an accurate CAD geometry has many applications in design, analysis, assembly and manufacture. Although there are various tools available that help fit surfaces, the pipeline from acquiring discretised mesh/point cloud data to generating a usable CAD model remains a challenging and crucial process. It is costly, error-prone and requires human input. The difficulty lies in re-creating a CAD model that interprets the original design intent and enforces the expected constraint. Exploiting prior information of geometric features/constraints, geometry-to-mesh associativity and integrating knowledge in the process can help increase the quality of the result and reduce the level of difficulty.

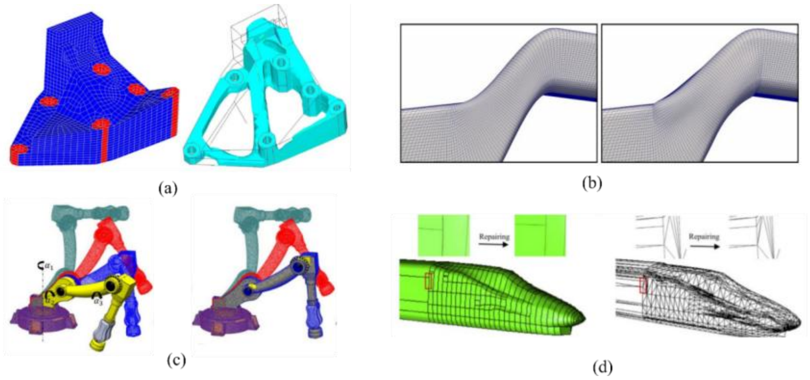

Figure 15: An example of: (a) topology optimization result [60]; (b) shape optimization result [61]; (c) scanned point cloud [59]; (d) geometry repairing based on the mesh [58].

The first situation is to create a CAD geometry from a mesh or a point cloud. The common method is to fit the bounding surfaces of the model. If the input is a point cloud, there is the additional challenge of filtering noise and identifying edges and corners, and no mesh connectivity information is available to facilitate this step. The general procedure is explained in Figure 16 [62]. The discrete data input is first pre-processed and then segmented into separate regions based on some geometry criteria to identify a set of closely matching geometric features or surfaces. Each region is classified depending on the type of surface, before the appropriate surface fitting operation is used to create the surface matching the region. The geometry model created can be either a feature-based parametric model or a free-form NURBS model. The former creates traditional CAD features but is limited to simple surfaces. The latter can reproduce complex shapes with little user intervention, but can result in overfitting. Also, it cannot provide additional information beyond the geometry model.

Manufacturing discrepancies and in-service wear and damage often imply that the real geometry of the product is slightly different from the design CAD model, as shown in Figure 17. Analysing the real-world model requires digitizing the actual shape and reconstructing a CAD model from a point cloud. While the surface fitting method can be employed, it is often more convenient to modify the reference CAD model or design a new one to match the actual shape accurately. This represents the second case of geometry creation-geometry morphing to fit a mesh/point cloud. One approach is to convert the designed CAD geometry to NURBS surfaces and move the control points to morph the geometry to match the scanned data. While the deformation vector can be accurately applied this way, the level of difficulty and effectiveness of the method depends on the extent of similarity between the designed CAD model and the scanned model. If the topology of the scanned model has changed, additional processing is required to identify topological discrepancies and their effects on the matching.

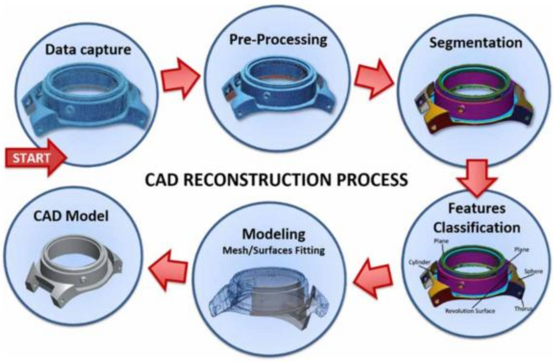

Figure 16: The common process of creating a CAD model from discrete data [62].

In some applications, the deformation vector is known as a result of a simulation. Therefore, the deformed and undeformed meshes are available along with the original CAD model, but the deformed CAD model is not known. For example, the cold shape of an aero-engine blade is designed in CAD, meshed, and used for a thermo-mechanical analysis. The deformed mesh representing the hot shape of the blade is obtained and needs to be converted to a new CAD model that can be meshed and used for a subsequent CFD analysis [64] as shown in Figure 18. Once again, the new CAD model can be created by stitching adjoining fitting surfaces, following the workflow described in Figure 16. However, if the relationship between the original CAD model and the undeformed mesh has been maintained, clustering the discrete nodes is easier, and the new surfaces can be accurately defined. Otherwise, a common strategy is to first identify which mesh entities (mesh node, mesh edge, and mesh element) lie on which geometrical entities (vertex, curve, and surface) of the CAD model, which requires an automatic match algorithm. This matching algorithm must be able to cope with topological discrepancies between the design CAD model and the analysis model represented by the mesh, which may have been introduced by the mesh generator or by the user during the geometry handling step. A potential solution has been demonstrated in [64], which consists of identifying the boundary of the virtual topology regions and re-parameterizing these regions automatically. Once the associativity with the mesh has been recovered, it is then clear for the deformed mesh which mesh entities should be clustered to create a curve or a surface.

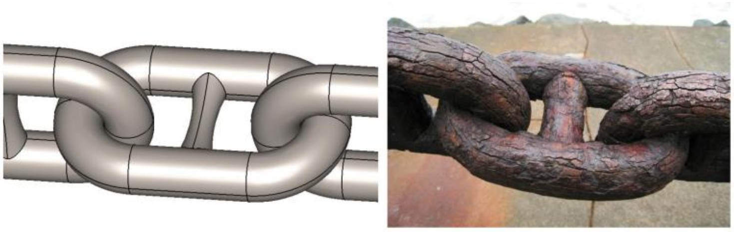

Figure 17: A designed model vs a manufactured model experiencing damage [63].

Another approach is to reparameterize the CAD model to allow the deformation vector captured on the mesh to be applied as a parametric deformation. Using an inverse design routine, the new parameters can be optimized to minimize the distance to a target shape that corresponds to the deformation vector. An example has been demonstrated in [65], where the geometry of a wing surface in its undeformed (jig) shape is deformed to match the in-flight (1G) configuration. The volume mesh can then be deformed using methods such as radial basis function [66], linear elasticity [67], and so on.

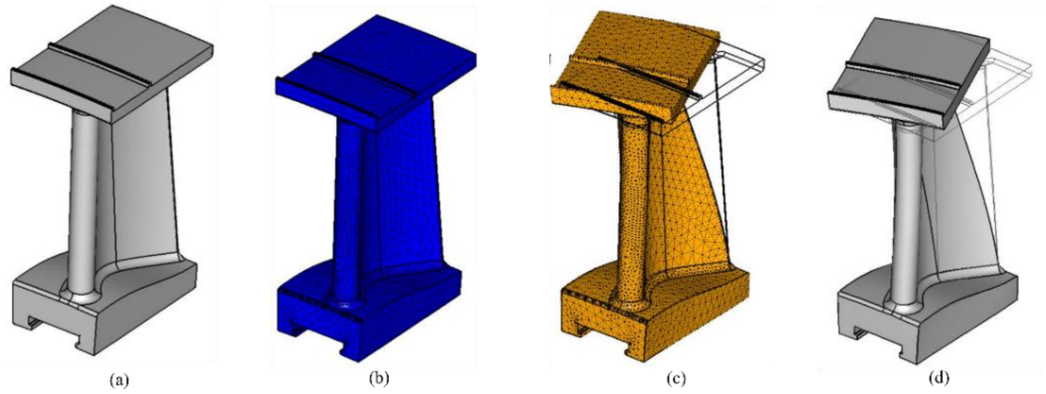

Figure 18: Example of creating a model based on mesh deformation: (a) the original CAD of an aero-engine blade; (b) the undeformed mesh; (c) the deformed mesh after a thermo-mechanical analysis; (d) the created deformed CAD [64].

6 FUTURE TRENDS: AI-AIDED GEOMETRY HANDLING AND MESHING

With the arrival of the artificial intelligence era, AI methods, especially those represented by deep learning, will bring profound changes to the fields of scientific and engineering computation. Deep learning offers significant advantages over traditional techniques, enabling intelligent decision-making, enhanced computational efficiency, and improved optimization capabilities. It is anticipated that this intelligence will make geometry handling and mesh generation processes more adaptable and user-friendly, thereby reducing the reliance on manual interventions and expert knowledge.

Traditional geometry handling often requires extensive manual input to manage diverse and complex geometries. AI-based method, with its ability to learn from data, promises to automate decision-making processes, such as geometry clean-up, de-feature, and decomposition, tailored to the specific needs of industry applications. Some initial research has been carried out. For instance, Qin et al. [68] have leveraged machine learning for shape recognition and the classification of CAD models. However, these methods focus primarily on mesh modifications without reflecting changes directly on the geometric models. Danglade et al. [69] proposed a heuristic rule-based machine learning method for feature removal from CAD models. However, this method still requires human intervention to evaluate the quality of the defeaturing results. The overall process of their defeaturing approach is illustrated in Figure 19. DiPrete et al. [70] introduced an innovative AI-assisted method using reinforcement learning to autonomously decompose planar CAD models into high-quality rectangular blocks. Owen et al. [71] proposed a method for processing CAD model features based on ensembles of decision trees, training samples with mesh quality as a metric, identifying potential problem areas, and presenting a prioritized list of geometric operation suggestions to the user. Each feature type requires a separate trained model for error detection, leading to high training costs and low operational efficiency. Current methods underutilize topological information intrinsic to CAD models and cannot directly update or visualize processed geometry, limiting their practical effectiveness. Further exploration is needed to identify suitable neural network architectures, such as graph neural networks, and develop effective geometry representation and feature extraction methods. Addressing the lack of large datasets for training also presents a significant challenge that requires resolution.

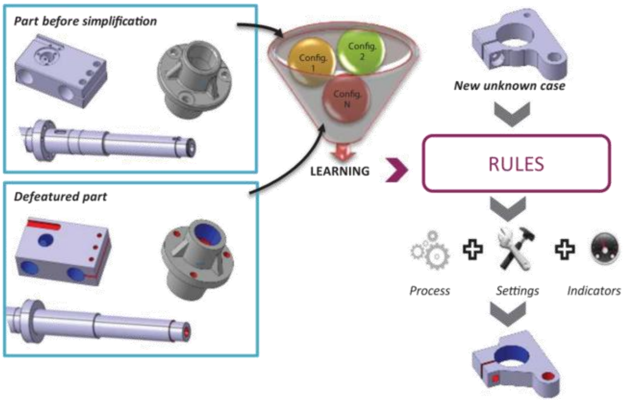

Figure 19: The process of defeaturing CAD models for simulation model preparation by machine learning [69].

In terms of mesh generation, AI-based solutions have been tested on mesh generation, mesh quality evaluation, and mesh smoothing. For example, Yao et al. [72] introduced a method to gradually generate meshes by deciding whether to add new vertices and how. Deng et al. [73] proposed a method using deep neural networks to generate meshes from given 2D sketches. Pan et al. [74] proposed a reinforcement learning method for quadrilateral mesh generation using "Soft Actor-Critic (SAC)". However, this method cannot accommodate user-driven mesh adjustments, limiting its applicability for complex geometries. Zhou et al. [75] proposed using convolutional neural network models to train on a large amount of 2D geometric model decomposition data, allowing for the decomposition of geometric models with the same topological structure into multiple blocks. Chen et al. [76] introduced a new unsupervised neural network-based differential structured mesh generation method (MGNet), but each trained model can only handle one type of shape. Dielen et al. [77] proposed a method based on training direction fields with curvature information to generate the quadrilateral mesh. But it cannot be applied to planes and other surfaces with non-unique curvature. Chen et al. proposed [78] an automatic mesh quality assessment method using deep convolutional networks, achieving a 92.5% accuracy rate on the NACA-Market dataset. Zhang et al. and Wang et al. [79] [80] proposed MQENet and GMeshNet based on dynamic graph attention for 2D mesh quality assessment. Chen et al. [81] introduced an automatic 3D structured mesh validity assessment method MVE-Net, providing datasets for several specific geometric models, but the algorithm is limited to structured topologies. Guo et al. [82] used neural networks to simulate optimization-based smoothing methods, which are highly efficient but have not been seen applied to quadrilateral meshes of complex models. Wang et al. [83] proposed a method combining Laplacian smoothing and deep learning, which was proven by experiments to be superior to traditional Laplacian methods and angle-based smoothing methods. Wang et al. [84] introduced a grid smoothing model called GMSNet, based on graph neural networks (GNNs), which showed higher efficiency compared to optimization-based smoothing methods through experiments. Similarly, as shown in Figure 20, Li et al. [85] proposed a semi-supervised learning-based approach for 2D quadrilateral mesh deformation that allows tangential sliding of boundary nodes while preserving boundary topology. By designing specialized loss functions, their method enables robust and adaptive deformation across various scenarios, addressing common issues like element inversion and distortion, and paving the way for future deep learning applications in mesh processing.

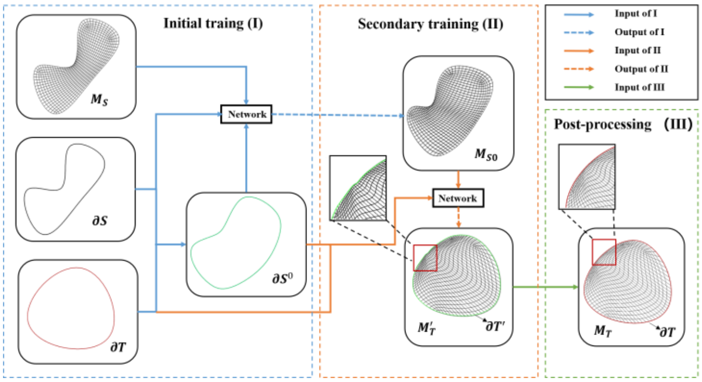

Figure 20: Semi-supervised learning-based 2D quadrilateral mesh deformation method [85].

Many current deep learning approaches for mesh generation remain in early development stages, primarily capable of handling basic geometric shapes and lacking universal applicability. While methods like advancing front can be easily framed as deep learning problems, more complex techniques such as block-structured mesh generation through PDE solving present challenges in devising viable solutions. The potential use of PINN-based methods may offer a promising solution for exploration. Further investigation is essential to incorporate deep learning into foundational mesh generation principles effectively. Existing deep learning-driven methods for mesh quality evaluation mainly concentrate on structured meshes, owing to their straightforward structure and easy definition of globally applicable feature expressions. Research is needed to explore how these techniques can be extended to unstructured meshes. Presently, deep learning-based approaches for geometric smoothing primarily target triangular and basic quadrilateral meshes. Comprehensive exploration of methodologies that can efficiently enhance the quality of complex model is still needed.

AI-based solutions also hold significant promise in integrating different model representations. The key to the integration lies in the ability to intelligently identify and establish equivalence relationships between different models. Equivalences can exist between geometry models, as well as between geometry and mesh models. Xu et al. [86] and Su et al. [87] have leveraged multi-view learning for 3D shape recognition, which involves identifying a 3D shape based on its 2D views captured from various perspectives. Qi et al. [88] have explored point cloud-based methods for 3D shape recognition. Huang et al. [89] proposed a geometric deep learning method to learn the correspondence of a set of collected deformed shapes. Through geometric deep learning, the underlying variations of the shapes are extracted and vertex-to-vertex mapping across different shapes are formed. However, current methods are limited to handling simple models and are not directly applicable in the context of CAD design and analysis. Achieving accurate recognition of correspondences between original CAD models and simplified CAD models needs the development of more sophisticated AI algorithms with enhanced intelligence capabilities.

In the scenario where there is no known CAD model for mesh-to-CAD conversion, a critical aspect is the intelligent segmentation of the mesh. Extensive research has been conducted on mesh segmentation using deep learning techniques. Abubakar Sulaiman et al. [90] introduced a three-stage approach utilizing a multi-view recurrent neural network to automatically segment a 3D shape into visually meaningful sub-meshes. In the context of CAD or CAE, the expectation is that the design intent can be recovered post-mesh segmentation. This will require intelligent decisions to be made based on existing experiences. For the second and third cases of mesh-to-CAD conversion, the process will involve the intelligent matching of two CAD models to establish equivalence between them. This task requires sophisticated algorithms capable of identifying and linking the corresponding elements between the two models accurately.

In summary, initial research integrating deep learning into geometry handling and mesh generation shows promise. Continued exploration and refinement of these methods will be crucial to overcoming existing challenges and advancing their practical applications.

7 CONCLUSIONS

Engineering organizations face increasingly complex challenges in simulation due to the growing scale and multidisciplinary nature of models. Efficient and accurate transformation from CAD designs to fit-for-purpose CAE models, along with rapid updates after design iterations, is critical for meeting these demands. This paper systematically reviews key challenges in CAD geometry handling, mesh generation, integration of different model representations, and CAD geometry recovery. Advances in these areas enable more productive use of simulation technologies in engineering practice. Furthermore, AI-aided methods show promising potential to enhance automation and intelligence in geometry processing and mesh generation, although current approaches remain limited and require further research. Looking ahead, the integration of deep learning techniques with traditional computational methods holds great promise for overcoming existing barriers and enabling more adaptive, efficient, and user-friendly simulation workflows.

8 ACKNOWLEDGEMENT

The work in this paper from the Dalian University of Technology is supported by the National Natural Science Foundation of China (Grant No. 52105246) and the Fundamental Research Funds for the Central Universities (Grant No DUT21RC(3)083)

Bowen Liu, http://orcid.org/0009-0005-1650-8356
Yuanxing Lv, http://orcid.org/0009-0000-2707-4024
Kaiting Li, http://orcid.org/0009-0002-9471-7733
Beiyan Jia, http://orcid.org/0009-0001-5905-7533
Liang Sun, http://orcid.org/0000-0002-7900-4447

REFERENCES

[1] Böhm, H.; Hornig, A.; Langkamp, A.; Gude, M.; Keskin, A.: On the Development of Strategies for an Efficient Semi-Automated Hex-Meshing Process of Complex Jet Engine Component Assemblies, ASME Turbo Expo 2015: Turbine Technical Conference and Exposition, 2015, V02BT39A031–V02BT39A031.

[2] Wang, J.; Wanscott, L. B.; Ong, L. J.: LS-DYNA® Performance Enhancement of Fan Blade Off Simulation on the Cray® XC40TM Supercomputer.

[3] Rouet, F.-H.; et al.: Scalability challenges of an industrial implicit Finite Element code, Proceedings of the 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2020, 505–514. https://doi.org/10.1109/IPDPS47924.2020.00059

[4] Slotnick, J.; Khodadoust, A.; Alonso, J.; Darmofal, D.: CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences, NASA/CR-2014-218178, National Aeronautics and Space Administration, Langley Research Center, 2014.

[5] Sobieszczanski-Sobieski, J.; Haftka, R. T.: Multidisciplinary aerospace design optimization: survey of recent developments, Struct. Optim., 14(1), 1997, 1–23. https://doi.org/10.1007/BF01197554

[6] Schrage, D.; Beltracchi, T.; Berke, L.; Dodd, A.; Niedling, L.; Sobieszczanski-Sobieski, J.: AIAA technical committee on multidisciplinary design optimization (MDO) white paper on current state of the art, Multidiscip. Des. Optim. Tech. Comm. Website, 1991.
[7] de Weck, O.; Agte, J.; Sobieszczanski-Sobieski, J.; Arendsen, P.; Morris, A.; Spieck, M.: State-of-the-art and future trends in multidisciplinary design optimization, 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2007, 1905. https://doi.org/10.2514/6.2007-1905
[8] Boussuge, F.; et al.: Capturing simulation intent in an ontology: CAD and CAE integration application, J. Eng. Des., 30(10–12), 2019, 688–725. https://doi.org/10.1080/09544828.2019.1630806
[9] Owen, S. J.; et al.: An immersive topology environment for meshing, Proceedings of the 16th International Meshing Roundtable, 2008, 553–577. doi: 10.1007/978-3-540-75103-8_31. https://doi.org/10.1007/978-3-540-75103-8_31
[10] Pi, J. S.; et al.: CFD vision 2030 study: a path to revolutionary computational aerosciences, 2013.
[11] Chawner, J. R.; Davis, J. F. III; Taylor, N. J.: Geometry, Mesh Generation, and the CFD 2030 Vision, 46th AIAA Fluid Dynamics Conference, 2016, 3485. https://doi.org/10.2514/6.2016-3485
[12] Lei, N.; Feng, Y.; Duan, J.; Zhang, P.; Zheng, X.; Gu, X.: A Survey of Finite Element Mesh Generation Methods, J. Univ. Electron. Sci. Technol. China, 53(6), 2024, 816–843. https://doi.org/10.12178/1001-0548.2024246.
[13] Pietroni, N.; et al.: Hex-mesh generation and processing: a survey, ACM Trans. Graph., 42(2), 2022, 1–44. https://doi.org/10.1145/3554920
[14] Zhao, X.-W.; Tong, X.-M.; Ning, F.-W.; Cai, M.-L.; Han, F.; Li, H.-G.: Review of empowering computer-aided engineering with artificial intelligence, Adv. Manuf., 2025. doi: 10.1007/s40436-025-00545-0. https://doi.org/10.1007/s40436-025-00545-0
[15] Zhu, H.; Shao, Y.; Liu, Y.; Zhao, J. B. T.: Automatic Hierarchical Mid-surface Abstraction of Thin-Walled Models Based on Rib Decomposition, 2016, 60–71. https://doi.org/10.1016/j.advengsoft.2016.01.007
[16] Casero, J. C.; Robinson, T. T.; Armstrong, C. G.; Geron, M.: Idealizing Quasi-Axisymmetric 3D Geometries to 2D-Axisymmetric Finite Element Models, Comput. Aided. Des. Appl., 16(6), 2019, 1020–1033. https://doi.org/10.14733/cadaps.2019.1020-1033
[17] Sun, L.; Yao, W.; Robinson, T. T.; Armstrong, C. G.; Marques, S. P.: Automated Mesh Deformation for Computer-Aided Design Models That Exhibit Boundary Topology Changes, AIAA J., 58(9), 2020, 4128–4141. https://doi.org/10.2514/1.J058760
[18] Robinson, T.; Armstrong, C.: Optimizing parameterized CAD geometries using sensitivities based on adjoint functions, Comput.-Aided Des. Appl., 9, 2012, 1–16. https://doi.org/10.3722/cadaps.2012.253-268
[19] Ito, Y.; Nakahashi, K.: Surface triangulation for polygonal models based on CAD data, Int. J. Numer. Methods Fluids, 39(1), 2002, 75–96. https://doi.org/10.1002/fld.281
[20] Ito, Y.; Nakahashi, K.: Direct surface triangulation using stereolithography data, AIAA J., 40(3), 2002, 490–496. https://doi.org/10.2514/2.1672
[21] Béchet, E.; Cuilliere, J.-C.; Trochu, F.: Generation of a finite element MESH from stereolithography.
[22] Quadros, W. R.; Owen, S. J.: Defeaturing CAD models using a geometry-based size field and facet-based reduction operators, in Proceedings of the 18th International Meshing Roundtable, Springer, 2009, 301–318. https://doi.org/10.1007/978-3-642-04319-2_18
[23] Nolan, D. C.; Tierney, C. M.; Armstrong, C. G.; Robinson, T. T.: Defining simulation intent, Comput. Des., 59, 2015, 50–63. https://doi.org/10.1016/j.cad.2014.08.030
[24] Sheffer, A.; Blacker, T.; Clements, J.; Bercovier, M.: Virtual Topology Operators for Meshing, Int. J. Comput. Geom. Appl., 10(3), 2000, 309–331. https://doi.org/10.1142/S0218195900000188

[25] Dannenhoffer, J.; Haimes, R.: Quilts: A technique for improving boundary representations for CFD, in 16th AIAA Computational Fluid Dynamics Conference, 2003, 4131. https://doi.org/10.2514/6.2003-4131
[26] Foucault, G.; Cuillière, J.-C.; François, V.; Léon, J.-C.; Maranzana, R.: Adaptation of CAD model topology for finite element analysis, Comput. Des., 40(2), 2008, 176–196. https://doi.org/10.1016/j.cad.2007.10.009
[27] Shi, Y.; Zhang, Y.; Xia, K.; Harik, R. F.: A Critical Review of Feature Recognition Techniques, Comput.-Aided Des. Appl., 2020. https://doi.org/10.14733/cadaps.2020.861-899
[28] Slyadnev, S.; Turlapov, V.: Simplification of CAD Models by Automatic Recognition and Suppression of Blend Chains, Program. Comput. Softw., 46, 2020, 233–243. https://doi.org/10.1134/S0361768820030081
[29] Lu, J. H.-C.; Song, I.; Quadros, W. R.; Shimada, K.: Geometric reasoning in sketch-based volumetric decomposition framework for hexahedral meshing, Eng. Comput., 30(2), 2014, 237–252. https://doi.org/10.1007/s00366-013-0332-z
[30] Wu, H.; Gao, S.: Automatic Swept Volume Decomposition based on Sweep Directions Extraction for Hexahedral Meshing, Procedia Eng., 82, 2014, 136–148. https://doi.org/10.1016/j.proeng.2014.10.379
[31] Sun, L.; Tierney, C. M.; Armstrong, C. G.; Robinson, T. T.: Decomposing complex thin-walled CAD models for hexahedral-dominant meshing, Comput. Des., 103, 2018, 118–131. https://doi.org/10.1016/j.cad.2017.11.004
[32] Rezayat, M.: Midsurface abstraction from 3D solid models: general theory and applications, Comput. Des., 28(11), 1996, 905–915. https://doi.org/10.1016/0010-4485(96)00018-8
[33] Quadros, W. R.: An approach for extracting non-manifold mid-surfaces of thin-wall solids using chordal axis transform, Eng. Comput., 24(3), 2008, 305–319. https://doi.org/10.1007/s00366-008-0094-1
[34] Robinson, T. T.; Armstrong, C. G.; Fairey, R.: Automated mixed dimensional modelling from 2D and 3D CAD models, Finite Elem. Anal. Des., 47(2), 2011, 151–165. https://doi.org/10.1016/j.finel.2010.08.010
[35] Lee, C. K.; Xu, Q. X.: A new automatic adaptive 3D solid mesh generation scheme for thin-walled structures, Int. J. Numer. Methods Eng., 62(11), 2005, 1519–1558. https://doi.org/10.1002/nme.1239
[36] Pollard, C.; Bucklow, H.; Butlin, G.; Johnson, B.: ITI Limited (ITI), UK; Realistic Engineering Analysis Limited (REAL), 2017.
[37] Chawner, J. R.; Taylor, N. J.: Progress in Geometry Modeling and Mesh Generation Toward the CFD Vision 2030, in AIAA Aviation 2019 Forum, 2019, 2945. https://doi.org/10.2514/6.2019-2945
[38] Bobrowski, K.; Ferrer, E.; Valero, E.; Barnewitz, H.: Aerodynamic shape optimization using geometry surrogates and adjoint method, AIAA J., 2017, 3304–3317. https://doi.org/10.2514/1.J055766
[39] Owen, S.: A survey of unstructured mesh generation technology, 7th Int. Meshing Roundtable, 1998, 25. doi: 10.1.1.34.5079.
[40] Ansys: ICEM CFD, 2020.
[41] Bunin, G.: A continuum theory for unstructured mesh generation in two dimensions, Comput. Aided Geom. Des., 25(1), 2008, 14–40. https://doi.org/10.1016/j.cagd.2007.05.002
[42] Palacios, J.; Zhang, E.: Rotational symmetry field design on surfaces, ACM Trans. Graph., 26(3), 2007, 55. https://doi.org/10.1145/1276377.1276446
[43] Blacker, T.: Automated conformal hexahedral meshing constraints, challenges and opportunities, Eng. Comput., 17, 2001, 201–210. https://doi.org/10.1007/PL00013384
[44] Nieser, M.; Reitebuch, U.; Polthier, K.: Cubecover–parameterization of 3D volumes, Comput. Graph. Forum, 30(5), 2011, 1397–1406. https://doi.org/10.1111/j.1467-8659.2011.02014.x
[45] Huang, J.; Tong, Y.; Wei, H.; Bao, H.: Boundary aligned smooth 3D cross-frame field, ACM Trans. Graph., 30(6), 2011, 143. https://doi.org/10.1145/2070781.2024177

[46] Reberol, M.; Chemin, A.; Remacle, J.-F.: Multiple approaches to frame field correction for CAD models, arXiv Prepr. arXiv:1912.01248, 2019.
[47] Ruiz-Gironés, E.; Roca, X.; Sarrate, J.: High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3D CAD representation, Comput. Des., 72, 2016, 52–64. https://doi.org/10.1016/j.cad.2015.06.011
[48] Hughes, T. J. R.; Cottrell, J. A.; Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Eng., 194(39–41), 2005, 4135–4195. https://doi.org/10.1016/j.cma.2004.10.008
[49] Sun, L.; Armstrong, C. G.; Robinson, T. T.; Papadimitrakis, D.: Quadrilateral multiblock decomposition via auxiliary subdivision, J. Comput. Des. Eng., 8(3), 2021, 871–893. https://doi.org/10.1093/jcde/qwab020
[50] Arabshahi, S.; Barton, D. C.; Shaw, N. K.: Towards integrated design and analysis, Finite Elem. Anal. Des., 9(4), 1991, 271–293. https://doi.org/10.1016/0168-874X(91)90002-G
[51] Kim, S.; Lee, K.; Hong, T.; Kim, M.; Jung, M.; Song, Y.: An integrated approach to realize multi-resolution of B-rep model, Proc. ACM Symp. Solid Phys. Model., 2005, 153–162. https://doi.org/10.1145/1060244.1060262
[52] Feng, Q.; Zhou, X.; Li, J.: A hybrid and automated approach to adapt geometry model for CAD/CAE integration, Eng. Comput., 36(2), 2020, 543–563. https://doi.org/10.1007/s00366-019-00713-4
[53] Tierney, C.; Nolan, D.; Robinson, T.; Armstrong, C.: Using mesh-geometry relationships to transfer analysis models between CAE tools, Eng. Comput., 31(3), 2015, 465–481. https://doi.org/10.1007/s00366-014-0377-7
[54] Nguyen, V. P.; Anitescu, C.; Bordas, S. P. A.; Rabczuk, T.: Isogeometric analysis: an overview and computer implementation aspects, Math. Comput. Simul., 117, 2015, 89–116. https://doi.org/10.1016/j.matcom.2015.05.008
[55] Bazilevs, Y.; Calo, V. M.; Cottrell, J. A.; Evans, J. A.; Hughes, T. J. R.; Lipton, S.; Scott, M. A.; Sederberg, T. W.: Isogeometric analysis using T-splines, Comput. Methods Appl. Mech. Eng., 199(5–8), 2010, 229–263. https://doi.org/10.1016/j.cma.2009.02.036
[56] Vuong, A.-V.; Giannelli, C.; Jüttler, B.; Simeon, B.: A hierarchical approach to adaptive local refinement in isogeometric analysis, Comput. Methods Appl. Mech. Eng., 200(49–52), 2011, 3554–3567. https://doi.org/10.1016/j.cma.2011.09.004
[57] Kosinka, J.; Barendrecht, P.; Cashman, T.; Dodgson, N.; Sabin, M.: Modelling and IgA, CAE Geom. Work., 2017.
[58] Chen, J.; Cao, B.; Zheng, Y.; Xie, L.; Li, C.; Xiao, Z.: Automatic surface repairing, defeaturing and meshing algorithms based on an extended B-rep, Adv. Eng. Softw., 86, 2015, 55–69. https://doi.org/10.1016/j.advengsoft.2015.04.004
[59] Shah, G. A.; Polette, A.; Pernot, J.-P.; Giannini, F.; Monti, M.: Simulated annealing-based fitting of CAD models to point clouds of mechanical parts' assemblies, Eng. Comput., 2020, 1–19. https://doi.org/10.1007/s00366-020-00970-8
[60] Zhou, M.; Fleury, R.; Shyy, Y.-K.; Thomas, H.; Brennan, J.: Progress in topology optimization with manufacturing constraints, in 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2002, 5614. https://doi.org/10.2514/6.2002-5614
[61] Liatsikouras, A. G.; Pierrot, G.: Soft handle triggering: a CAD-free parameterization tool for adjoint-based optimization methods, 6th European Conference on Computational Mechanics (ECCM 6) and 7th European Conference on Computational Fluid Dynamics (ECFD 7) Conference Proceedings, 2018.
[62] Buonamici, F.; Carfagni, M.; Furferi, R.; Governi, L.; Lapini, A.; Volpe, Y.: Reverse engineering modeling methods and tools: a survey, Computer-Aided Design & Applications, 15(3), 2018, 443–464. https://doi.org/10.1080/16864360.2017.1397894
[63] Bucklow, H.: Challenges in CAE Geometry.
[64] Chinn, A.: Back to CAD: Robust Morphing of CAD Geometry based on Simulation Results, ITI.

[65] Lecallard, B.; Robinson, T. T.; Marques, S. P.: Mesh and Geometry Manipulations for Optimization and Inverse Design, AIAA Scitech 2021 Forum, 2021, 1901. https://doi.org/10.2514/6.2021-1901
[66] Biancolini, M. E.: Fast Radial Basis Functions for Engineering Applications, Springer, 2017. https://doi.org/10.1007/978-3-319-75011-8
[67] Selim, M. M.; Koomullil, R. P.: Mesh deformation approaches–a survey, 2016.
[68] Qin, F.; Li, L.; Gao, S.; Yang, X.; Chen, X.: A deep learning approach to the classification of 3D CAD models, Journal of Zhejiang University Science C, 15(2), 2014, 91–106. https://doi.org/10.1631/jzus.C1300185
[69] Danglade, F.; Pernot, J.-P.; Véron, P.: On the use of machine learning to defeature CAD models for simulation, Computer-Aided Design & Applications, 11(3), 2014, 358–368. https://doi.org/10.1080/16864360.2013.863510
[70] DiPrete, B. C.; Garimella, R. V.; Cardona, C. G.; Ray, N.: Reinforcement Learning for Block Decomposition of CAD Models, arXiv Preprint arXiv2302.11066, 2023.
[71] Owen, S. J.; Shead, T.; Martin, S.: CAD Defeaturing using Machine Learning, Sandia National Laboratories, Albuquerque, NM, 2019.
[72] Yao, S.; Yan, B.; Chen, B.; Zeng, Y.: An ANN-based element extraction method for automatic mesh generation, Expert Systems with Applications, 29(1), 2005, 193–206. https://doi.org/10.1016/j.eswa.2005.01.019
[73] Deng, Z.; Liu, Y.; Pan, H.; Jabi, W.; Zhang, J.; Deng, B.: Sketch2pq: freeform planar quadrilateral mesh design via a single sketch, IEEE Transactions on Visualization and Computer Graphics, 2022. https://doi.org/10.1109/TVCG.2022.3170853
[74] Pan, J.; Huang, J.; Cheng, G.; Zeng, Y.: Reinforcement learning for automatic quadrilateral mesh generation: A soft actor–critic approach, Neural Networks, 157, 2023, 288–304. https://doi.org/10.1016/j.neunet.2022.10.022
[75] Zhou, Y.; Cai, X.; Zhao, Q.; Xiao, Z.; Xu, G.: Quadrilateral Mesh Generation Method Based on Convolutional Neural Network, Information, 14(5), 2023, 273. https://doi.org/10.3390/info14050273
[76] Chen, X.; Liu, J.; Pang, Y.; Chen, J.; Chi, L.; Gong, C.: MGNet: a novel differential mesh generation method based on unsupervised neural networks, Engineering with Computers, 38(5), 2022, 4409–4421. https://doi.org/10.1007/s00366-022-01632-7
[77] Dielen, A.; Lim, I.; Lyon, M.; Kobbelt, L.: Learning direction fields for quad mesh generation, Computer Graphics Forum, 2021, 181–191. https://doi.org/10.1111/cgf.14366
[78] Chen, X.; Liu, J.; Pang, Y.; Chen, J.; Chi, L.; Gong, C.: Developing a new mesh quality evaluation method based on convolutional neural network, Engineering Applications of Computational Fluid Mechanics, 14(1), 2020, 391–400. https://doi.org/10.1080/19942060.2020.1720820
[79] Zhang, H.; Li, H.; Li, N.; Wang, X.: MQENet: A Mesh Quality Evaluation Neural Network Based on Dynamic Graph Attention, arXiv Preprint arXiv2309.01067, 2023.
[80] Wang, Z.; Chen, X.; Li, T.; Gong, C.; Pang, Y.; Liu, J.: Evaluating mesh quality with graph neural networks, Engineering with Computers, 38(5), 2022, 4663–4673. https://doi.org/10.1007/s00366-022-01720-8
[81] Chen, X.; Liu, J.; Gong, C.; Li, S.; Pang, Y.; Chen, B.: MVE-Net: An automatic 3-D structured mesh validity evaluation framework using deep neural networks, Computer-Aided Design, 141, 2021, 103104. https://doi.org/10.1016/j.cad.2021.103104
[82] Guo, Y.; Wang, C.; Ma, Z.; Huang, X.; Sun, K.; Zhao, R.: A new mesh smoothing method based on a neural network, Computational Mechanics, 2021, 1–14.
[83] Wang, N.; Zhang, L.; Deng, X.: Unstructured surface mesh smoothing method based on deep reinforcement learning, Computational Mechanics, 2023, 1–24. https://doi.org/10.1007/s00466-023-02370-3
[84] Wang, Z.; Chen, X.; Yan, J.; Liu, J.: Proposing an intelligent mesh smoothing method with graph neural networks, Frontiers of Information Technology & Electronic Engineering, 2024. http://doi.org/10.1631/FITEE.2300878.

[85] Li, S.; Li, Q.; Li, K.; Sun, L.: Deep learning-based semi-supervised approach for high-quality 2D quadrilateral mesh deformation, Engineering with Computers, 2025, 1–15. https://doi.org/10.1007/s00366-025-02147-7

[86] Xu, Y.; Zheng, C.; Xu, R.; Quan, Y.; Ling, H.: Multi-view 3D shape recognition via correspondence-aware deep learning, IEEE Transactions on Image Processing, 30, 2021, 5299–5312. https://doi.org/10.1109/TIP.2021.3082310

[87] Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E.: Multi-view convolutional neural networks for 3D shape recognition, Proceedings of the IEEE International Conference on Computer Vision, 2015, 945–953. https://doi.org/10.1109/ICCV.2015.114

[88] Qi, C. R.; Su, H.; Mo, K.; Guibas, L. J.: PointNet: Deep learning on point sets for 3D classification and segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, 652–660.

[89] Huang, J.; Sun, H.; Kwok, T.-H.; Zhou, C.; Xu, W.: Geometric deep learning for shape correspondence in mass customization by three-dimensional printing, Journal of Manufacturing Science and Engineering, 142(6), 2020, 61003. https://doi.org/10.1115/1.4046746

[90] Gezawa, A. S.; Wang, Q.; Chiroma, H.; Lei, Y.: A Deep Learning Approach to Mesh Segmentation, Computer Modeling in Engineering & Sciences, 135(2), 2023. https://doi.org/10.32604/cmes.2022.021351