



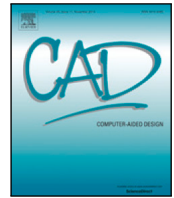
Linking Model-Based Definition and Non-Intrusive Finite Element Analysis for Automated Variation Simulation

Downloaded from: <https://research.chalmers.se>, 2025-11-23 15:31 UTC

Citation for the original published paper (version of record):

Roth, M., Kopatsch, J., Wärmefjord, K. et al (2026). Linking Model-Based Definition and Non-Intrusive Finite Element Analysis for Automated Variation Simulation. CAD Computer Aided Design, 191.
<http://dx.doi.org/10.1016/j.cad.2025.104003>

N.B. When citing this work, cite the original published paper.



Research Paper

Linking Model-Based Definition and Non-Intrusive Finite Element Analysis for Automated Variation Simulation

Martin Roth ^a, Jan Kopatsch ^b, Kristina Wärmefjord ^a, Rikard Söderberg ^a,
Stefan Goetz ^b

^a Chalmers University of Technology, Gothenburg, 41296, Sweden

^b Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, 91058, Germany

ARTICLE INFO

Keywords:

Model-Based Definition (MBD)
Digital thread
Quality Information Framework (QIF)
Tolerancing
Variation simulation
Finite Element Analysis (FEA)
Finite Element meshes

ABSTRACT

Computer-Aided Tolerancing (CAT) software has become the standard for statistically analyzing the effects of geometrical part variations on product quality. Irrespective of CAT's scope and technical depth, Finite Element Analysis (FEA) software, used to simulate the physical product behavior for ideal part geometry in the first place, is also often used for studies with geometrical shapes deviating from their nominal. However, this requires a manual translation of the tolerances specified in the design phase into geometrical variations represented by Finite Element (FE) meshes and their transfer to the FEA software. The method presented in this article exploits the potential of Model-Based Definition by establishing a link between Computer-Aided Design and FEA software to empower the latter for variation simulation based on semantic Geometric Dimensioning and Tolerancing (GD&T) information. To transfer this information exchanged via the Quality Information Framework (QIF) standard, a new mapping algorithm is presented that automatically decomposes FE meshes into geometrical face elements and creates a semantic link with the GD&T information carried in QIF. As a result, geometrical features are simultaneously described through meshes with nodes in the 3D Euclidean space and mathematical geometrical faces in the 2D parameter space. Exploiting this duality, mesh deviations are modeled indirectly by adjusting the mapped feature descriptions. An exemplary implementation in ANSYS® and its usage for non-intrusive structural simulations illustrates that sharing tolerancing information via QIF enables an automated, GD&T standards-compliant variation simulation within FEA software environments and is one step closer to a seamless digital thread for geometry assurance.

1. Motivation

Product development aims to design and realize technical products that can fulfill all internal and external requirements while being unique or at least superior to alternative competitors in decisive points, particularly costs and quality. The product design phase significantly impacts the final product quality [1] and requires numerous decisions about the final part geometry. However, assuring the geometry when converting the idealized designs into physical parts and assemblies is complicated because manufacturing uncertainty and variability are naturally given and unavoidable [2]. Long before this principle was established as the axiom of manufacturing imprecision [3], people recognized that geometrical variation cannot be limited to zero [4] and does not have to be since parts have only to be “sufficiently identical” [5]. To some extent, deviations from the ideal part geometry are acceptable as long as the overall quality of the product can

be assured within predefined limits to conform to the requirements. Variation simulation, which virtually represents the part variations and propagates them on the assembly level, supports analyzing the product behavior in the presence of geometrical uncertainties early on [2]. Since early research starting in the 1960s [6] and the first releases of Computer-Aided Tolerancing (CAT) software in the 1990s [7], a small number of commercial tools dedicated to variation simulation and tolerance analysis¹ have prevailed in industry [8] and are embedded in today's product development workflows.

However, the introduction and especially usage of CAT are largely determined by the software's applicability [9,10]. Referring to the more general term “usability” and its definition given in ISO 9241-11, it can be expressed as “the extent to which a system [...] can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use” [11]. Usability also ap-

* Corresponding author.

E-mail address: martin.roth@chalmers.se (M. Roth).

¹ A differentiation of the terms ‘variation simulation’ and ‘tolerance analysis’ is given in Section 2.1.

plies to CAT and is particularly challenging as diverse processes, users, goals, and contexts of use can differ from case to case and may vary significantly from the initially specified ones [12]. Thus, irrespective of the scope and technical depth of CAT tools, the software already used within the product development process, such as for Computer-Aided Design (CAD) or Finite Element Analysis (FEA), is commonly used beyond its actual purpose and empowered to simulate the product behavior under variations [10,12]. Since this comes with some issues, like low efficiency of FEA-based variation simulation, it has been a recurring research topic over the last decades. Little attention has been paid to increasing the efficiency of the time-consuming manual pre-processing steps, particularly setting up the geometrical model to represent part variations within FEA.

Hence, this article proposes a novel method for empowering FEA software to simulate non-rigid assemblies under variations and analyze geometrical and non-geometrical assembly responses. It is based on the Quality Information Framework (QIF), a Model-Based Definition (MBD) standard, that enables sharing and reusing part geometry and Geometric Dimensioning and Tolerancing (GD&T) information.² The method presented allows automated variation modeling and simulation in FEA while staying compliant with established tolerancing standards and representation models. The paper is structured as follows. After reflecting on the related works in Section 2 and defining this article's scope and research question, the idea of leveraging FEA software for variation simulation by QIF is elaborated in Section 3. Section 4 presents an exemplary implementation of the developed method in ANSYS® and its application for statistical structural analysis of assemblies with part geometry variations. Section 5 concludes this article and gives an outlook on future research work in this field. This article focuses on automatic model setup for representing size, location, and orientation deviations, assuming an ideal form in FEA — representing form deviations is not studied in detail.

2. State of the art and related works

A summary of the relevant background of CAT in Section 2.1 is followed by a reflection of FEA-based variation simulation and tolerance analysis in Section 2.2. The section concludes with a discussion of current research gaps and introduces the research question of this article in Section 2.3.

2.1. Simulating product behavior with geometrical variations

Since geometrical part deviations are unavoidable (see Section 1), they need to be verified, reduced, and monitored to ensure the final product quality when they propagate and accumulate on the assembly level [13]. The term deviation of a part refers to the difference between the ideal and actual geometry obtained for one single part [14]. Variation, in contrast, describes the variability of deviation, which occurs when producing this part multiple times with the same process [14]. The purpose of variation simulation is thus to virtually predict how the assembly behaves under and responds to geometrical part variations and additional deviations induced by the assembly process [2,15]. To limit geometrical variation, product designers specify tolerances, giving information on how and to what extent features, i.e., physical portions of parts or their virtual representation, are acceptable to be off from their nominal size, location, orientation, and form [16]. In contrast to variation simulation, the aim of tolerance analysis is thus to verify if the specified tolerances can fulfill the quality requirements [17], broken down into a set of representative geometrical and non-geometrical

Key Characteristics (KC), sensitive to variations and critical for product performance or aesthetics [18]. Thus, tolerance analysis always includes variation simulation but the variation information is derived from the tolerance specifications. However, both terms are often used synonymously in literature — even though the input differs.

Simulation models By making assumptions and simplifications at the feature, part, and assembly level [19], geometrical and behavior models are used to establish the link between tolerances, variations, deviations, and the KCs [2]. Geometrical models represent the part features at their nominal as well as their deviations and variations before assembly [2]. For this purpose, specific models have been developed, which often fall back on general mathematical representations, such as vectors, tensors, matrices, point clouds, or meshes [8,20]. Behavior models are used to model the interaction between the individual parts and features during and after assembly [15]. Displacement accumulation methods propagate all feature deviations from part to part, amplifying or mitigating each other and, in sum, influencing the KCs [15]. Although methods are presented in literature that directly accumulate the individual tolerance zones [15], displacement accumulation is the basis of most CAT software used in industry [8]. Performing tolerance analysis using CAT software consequently includes variation simulation since the information on the tolerance zones is internally translated into variations to indirectly analyze the tolerances' impact. Sampling techniques, primarily Monte Carlo Sampling [2], are mostly used to draw statistical conclusions on the whole population from a sufficiently large number of samples [21]. Based on assumed or known manufacturing distributions [22], a finite set of parts, deviating within their tolerance zones in size, location, orientation, and form, is represented via the geometrical model and randomly paired to study the probabilistic product behavior and evaluate the KCs [21]. The variations do not necessarily have to be generated based on tolerances but can also originate directly from inspection data [2] or numerical manufacturing process simulations [23,24]. Although sampling lacks efficiency compared to other statistical approaches [25], it can be applied to any distribution and model type [2], highly nonlinear [26], and implicitly represented via numerical simulation represented interrelations [15]. Regarded as a “panacea” [26] for variation simulation, sampling is preferred in research [2] and implemented in common CAT software [8].

Tolerancing and variation information sharing Information on the part geometries, tolerances, and variation is needed to set up and feed the simulation models. MBD has proven its strengths in centralizing most of the information required within one authority model [27], where GD&T specifications are directly attached to and stored within the 3D CAD model [28]. The annotations, both human-readable and computer-interpretable and known under the term PMI (Product and Manufacturing Information) [29], are semantically linked to the geometrical features, facilitating the information-sharing workflows between the subsequent downstream applications with multiple users and software systems involved [30]. Staying in one mutual software landscape helps define a closed information-sharing loop [31] and eases collaborative work [12]. However, it is only realistic for closed, stringent, and small-scale product development processes [12]. Instead, it is common to bridge gaps between CAD and CAT software since some, but not all, CAT systems are directly embedded as applications in CAD [8,32]. Relying on neutral MBD standards fosters information-sharing standardization and assures systems interoperability [27]. For this reason, the STEP AP242 (ISO 10303-242) and JT (ISO 14306:2017) standards are used for automating tolerance analysis by automatically deriving geometrical models from geometry and GD&T information, communicated in a software-neutral format [33]. The QIF standard (ISO 23952:2020) has recently drawn attention since it supports sharing deviation and variation information along with MBD information [34]. Thus, QIF has overlaps with STEP AP242 and JT in the MBD part [35] but can also be used for enriching variation simulation with real manufacturing information [33] to either make

² The term GD&T, preferred by the American Society of Mechanical Engineers (ASME), is used in this article without implying that the standards by the International Organization for Standardization (ISO) are less functional.

more process-oriented statistical evaluations or to feed digital twins to make one-by-one decisions [34]. Literature covers various applications and examples exploiting MBD's potential for tolerancing downstream applications based on STEP AP242 [30,36–39], JT [40], and QIF [12, 41–43].

Information sharing is essential when software, not explicitly developed for CAT but empowered for variation simulation, is used [12]. Reasons for not using CAT can be that users lack access to CAT software, do not have the knowledge or skills to use it, aim to save time since simulation models based on nominal part geometry are already set up in software, or need simulation capabilities beyond the scope of CAT [10,12,44]. Besides using the parametric geometry representation and PMI in CAD, for instance, proposed in [45–47], Computer-Aided Engineering (CAE) software, are extended by variation modeling to, for example, analyze systems in motion [48–50], perform Computational Fluid Dynamics (CFD) simulations [51,52] or carry out structural and thermal analysis studies based on FEA [53–55].

2.2. FEA for variation simulation

Though some CAT tools include an FEA solver to model compliant parts and contacts [2], FEA software is preferred to analyze non-geometrical KCs, e.g., stress or temperature distributions in product design. In reliability and uncertainty modeling theory, FEA with uncertainty is addressed under the term stochastic FEA and distinguished into intrusive and non-intrusive approaches [57].

Intrusive FEA approaches integrate uncertainty directly into the FE equations and the solver [57]. Non-intrusive methods, in contrast, use the deterministic FEA code as a black box [58]. They are usually used in combination with sampling techniques to iteratively call FEA with uncertain inputs to study the stochastic system behavior [57]. Even though intrusive FEA is computationally efficient [58], non-intrusive FEA is preferably used to map geometrical variations as aleatory, i.e., irreducible but modelable, uncertainties [2] due to its ease of

use. Related works systematically define deviated part geometries as varying input and iteratively call the FEA solver to get the probabilistic assembly response as an output. Hence, the FE mesh takes over the role of the geometrical model representing deviations through node displacements, while the numerical solving of the equation system, defined by the constraints, contacts between parts, and loads, functions as an implicit behavior model. Besides classical Design of Experiments (DOE) [59–62], space-filling DOE, primarily Latin-Hypercube [59,63, 64] or Monte Carlo Sampling [53,65–67], is used to either directly call the solver or to evaluate pre-trained surrogate models. Since FEA is, in most cases, too computationally heavy to run for a sufficiently high number of iterations to draw reliable statistical conclusions, surrogate modeling is quite common to replace the computational-intensive FEA within analysis and optimization loops [54,63–65,67,68]. Besides geometrical variations, non-geometrical parameters, such as material or loads, are considered as uncertainty factors [69].

2.3. Scope of the article

Section 2.2 emphasizes that non-intrusive FEA for variation simulation is a common alternative to CAT software used in academia and industry. FEA's strengths in representing non-rigid assemblies, where part deformations and realistic contacts are considered in analyzing non-geometrical assembly responses, come with deficits in geometrical variation modeling. FEA software is not designed to focus on tolerances and variations in the first place. Thus, one major problem is the gap in the information-sharing workflow between CAD and FEA [54]. While MBD is well-established to automate CAT [33,34], a link to FEA software is largely missing. GD&T information from the design phase must usually be transferred manually from annotations made in 2D drawings or attached to 3D models and translated manually into variations. This information is used to vary the mesh either indirectly by adjusting the values of the parametric geometry representation of the parts [59] (see Fig. 1, strategy (a)) or directly by manipulating individual FE nodes of the nominal mesh [70] (see Fig. 1, strategy (b)).

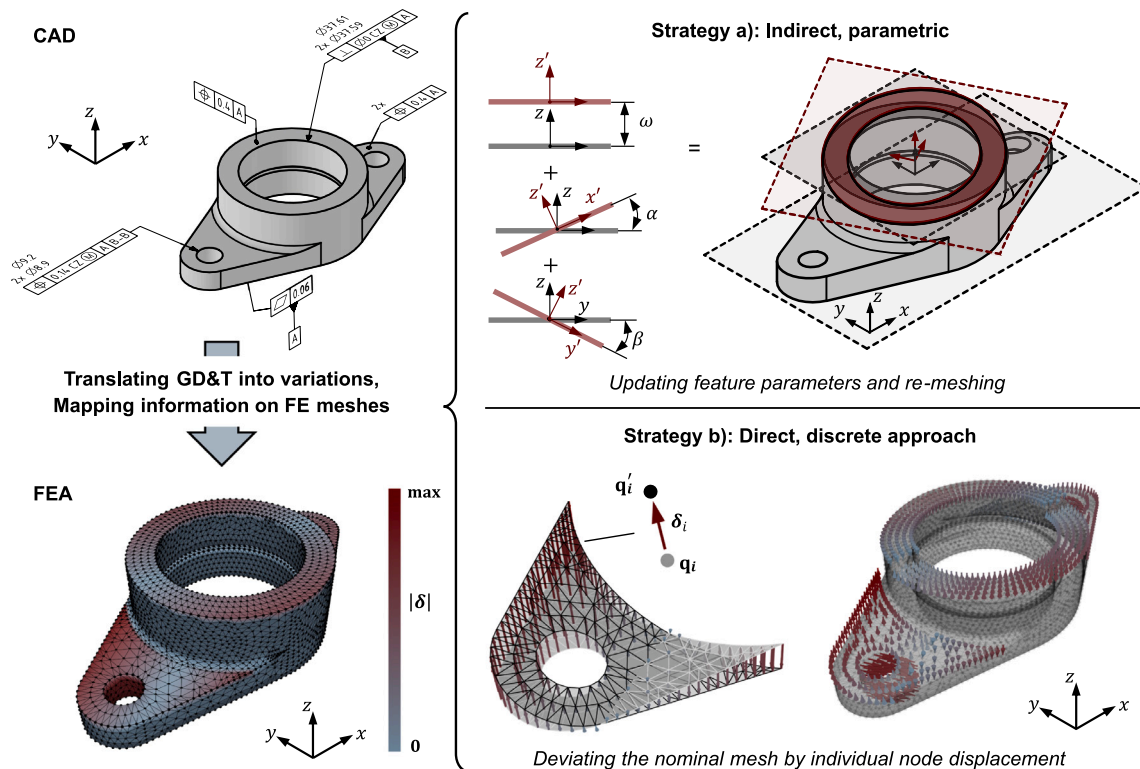


Fig. 1. Representing geometrical part variations in non-intrusive FEA: indirect, parametric strategy (a) vs. direct, discrete strategy (b). The sample part is inspired by an example given in ISO 16792:2021 [56]. Deviations are exaggerated.

Strategy (a) requires a new mesh to be created for each geometry variation, but has the benefit of directly using the parametric description of the geometry for variation modeling. However, the parameterization must often be done manually since the link to CAD or its parametrics might be lost when switching to FEA [54], or it is unsuitable for deviation modeling [71]. Strategy (b) avoids re-meshing the parts for each sample. Still, the tolerance features must initially be associated with the respective subsets of mesh nodes to calculate an individual translation vector for each node to represent the deviated geometry in total. Although multiple methods for node set selection are available, a robust automatic mesh segmentation matching the tolerance features definitions cannot be guaranteed [54]. In addition, geometry is only represented by nodes with three coordinates in the 3D Euclidean space, shared and connected by elements. Parametric information on the features, such as diameter, axis location, and direction of a cylindrical feature, necessary to generate the deviations and calculate the node translation, is lacking and needs to be derived.

In summary, a parametric geometry representation is necessary for deviation modeling on the one hand, and a discrete representation via meshes is necessary for FE on the other. Existing strategies, however, have shortcomings in combining both representations simultaneously. Motivated to overcome the current lack of interoperability between CAD and FEA and the existing barriers in FEA-based variation simulation, this article addresses the research question of how to link CAD and non-intrusive FEA to create a seamless and automated variation simulation workflow.

3. Empowering non-intrusive FEA for variation simulation through QIF 3.0

FEA is well-established in product development and follows a structured workflow comprising several pre-processing, processing, and post-processing steps. In common FEA, considering the geometries as ideal, the meshes for the imported part geometries of an assembly are first generated. The user defines boundary conditions, constraining the degrees of freedom of the parts and describing how the parts are allowed to deform or move. External forces and displacements, affecting the assembly and causing stress and deformation, as well as part materials are added [70]. All information together forms the FE model to be solved to analyze the non-rigid part or assembly behavior through predefined measures, such as maximum stress or deformation (see Fig. 2, left).

By following these steps, the model for the nominal status is set up and can be enhanced for variation simulation. As highlighted in Fig. 1, non-intrusive FEA methods do not directly incorporate geometrical uncertainties into the code; they are considered by adjusting the nominal FE model. Thus, part deviations can either be added before meshing the parts (see strategy (a), Fig. 1) or afterward directly to the mesh (see strategy (b), Fig. 1). Strategy (b) is employed in this article as it overcomes several drawbacks of strategy (a). An iterative re-meshing of each part for each sample would be time-consuming, especially for larger parts, and usually unnecessary since only a small number of all geometry elements contributing to the overall product quality are to be considered as deviating from their nominal. Applying the deviations directly to the mesh is further more flexible in considering form deviations, as parametric primitives, such as planes and cylinders, are analytically described by a small set of intrinsic characteristics. Modeling shape deviations with parametric surfaces would require a translation of the features into higher-order freeform geometry elements. Following strategy (b), GD&T information needs, first, to be transferred from CAD to FEA software, second, brought into the FE model, and, third, translated into variation information to be used to represent the variations through a representative number of deviated FE meshes (see Fig. 2, right). This strategy is well-aligned with the Skin Model Shapes (SMS) concept, a well-established tolerancing model for representing geometrical deviations. They typically represent geometry

deviating from its ideal through a discrete representation, such as point clouds or surface meshes, and show their strength in realistically representing shape deviations [19]. One possible solution would be to start on geometry element level, meshing them independently and assembling them into a volume mesh. However, this introduces some challenges. Independent surface patches do not share nodes and the generated meshes are not optimized for overall volumetric mesh quality. Creating a high-quality volume mesh from surface patches is more complex than starting from a solid mesh and splitting it into patches while maintaining connectivity to volume elements. For this reason, a nominal mesh, often generated in FEA software [72], is first segmented into features before the deviations are simulated feature by feature and finally combined to represent a complete part instance [73]. When staying within the FEA software, the updated meshes are the varying inputs of a set of FE models to be iteratively solved. An additional modeling step on the assembly level is needed to mimic the assembly process and sequence by bringing the deviated meshes into contact and in an assembled state before the product's behavior can be simulated through FEA (see Fig. 2, right). The final result is a discrete probability distribution for the characteristics of interest, evaluated using statistical measures.

With this general idea of the method, depicted in Fig. 2, in mind, Sections 3.1–3.3 analyze the three main steps and introduce solutions that foster an automated variation simulation based on MBD. Section 3.4 summarizes the main part by focusing on assembly process modeling in FEA when dealing with deviated parts. The flange shown in Fig. 1 accompanies the sections as a consistent example.

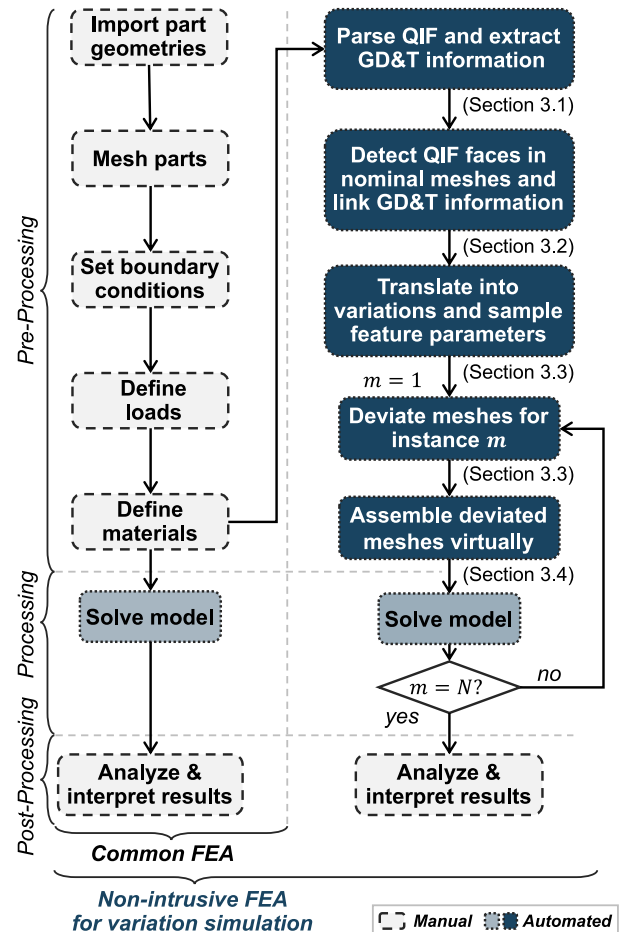


Fig. 2. Proposed workflow for automating FEA-based variation simulation through QIF.

3.1. Bringing information from CAD to FEA

Before information can be used in FEA for variation simulation at all, it must be brought into the software. The following section will elaborate on bridging the gap between CAD and FEA and explain the decision for QIF as MBD standard, building the foundation for the subsequent variation simulation steps.

Traditionally, GD&T information is communicated via technical drawings. Tolerances are added manually as annotations to the different 2D views, so there is no semantic link, i.e., “an associative relationship between digital elements” [56], between the tolerance specifications and the corresponding features in the 3D CAD model. MBD overcomes this drawback by assigning the tolerance specifications directly to the 3D model through PMIs. Along with assembly and part geometry information, GD&T information is captured within a single authority model and shared with subsequent downstream activities. Literature and practice have shown that CAD models enriched with PMI significantly increase the automation level of CAT (see Section 2.1). Direct and indirect tool interfaces are already available in FEA software to link the CAD and FEA environments, where neutral formats are preferred when ensuring software-independent interoperability. However, in contrast to CAT, only geometry and no GD&T information is transferred since the latter is irrelevant for FEA with ideal geometries. Although structural differences exist in data models, schema language, and details in scope [35,74], all three main MBD standards, JT, STEP AP242, and QIF 3.0, are suitable for information sharing between CAD and FEA and automating non-intrusive FEA.³

QIF is based on the XML Schema Definition Language (XSDL), providing data formatting and naming templates [75]. Thus, all files following this language are formatted as XML documents, both human-readable and machine-interpretable [75]. Unique identification designators (id) are used to create the semantic links between the individual QIF objects. Thus, data elements are not duplicated within a referencing object (which is the case in hierarchical models); rather, relationships are created, allowing objects to be used multiple times since they are decoupled from each other [75]. This concept leads to the name “decoupled normalized relationship model”, which is the fundamental of QIF to describe all components topologically and geometrically, their arrangement within assemblies, to be enriched with additional quality-related information [75]. Fig. 3 illustrates the basic structure of QIF. QIF is based on the Boundary Representation (BREP) method [75]. Each part is characterized by at least one <Body>, which is itself a Set of oriented <Shell> elements as a collection of connected <Face> elements.⁴ A <Face> is, in turn, defined through a reference to a geometrical <*Surface23> and at least one outer <Loop> plus optional inner loops. A <Loop> consists of at least one <Edge> that references a geometrical <Curve> and whose start and end are each given as a <Vertex> that references a geometrical <Point> element (see Fig. 3, left).

These elements, therefore, specify the parts' topology, defining how the geometry objects are connected and delimited [75]. The <GeometrySet> collects all objects that describe the actual geometry shape, classified according to their dimension into points, curves, and surfaces [75] (see Fig. 3, left).

QIF further uses three groups to represent GD&T specification: <Characteristics>, <Features>, and <DatumReferenceFrames> (see Fig. 3, right). A characteristic is defined as “a control placed on an element of a feature” [75]. Hence, <CharacteristicItems> apply tolerances to an individual feature by referencing a <*CharacteristicNominal>, which itself links to a <*FeatureNominal> and a <*CharacteristicDefinition> entity. The latter contains information on tolerance type, value, zones, etc., and references a <DatumReferenceFrame>

element, composed of multiple single <*Datum> elements in case of a location or orientation tolerance and can be reused by multiple <*CharacteristicNominal> elements, which define “information unique to a particular instance of a feature” [75].

Following the same logic, <*FeatureNominal> elements reference <*FeatureDefinition> elements with additional detailed feature information but also may trace back to one or multiple topology elements, such as <Face> elements via <EntityInternalIds> (see Fig. 3, right).

Splitting characteristic and feature information into items, nominals, and definitions ensures reusability through shareable elements (definitions) and ambiguity through non-shareable elements (nominals) (see [75] for more details).

Since QIF is designed from a quality inspection perspective, it covers in total seven application areas – the MBD information presented so far is only one of them – to capture information from design and inspection within one unified file and format. Hence, files can be augmented with measurement results and fed back to the simulation, such as FEA, as presented in [41]. Since the feature-based logic of pre-measurement and post-measurement PMI is harmonized [75], the same methods and algorithms can be used to bring virtually generated or measured deviations into simulation applied in product design as well as realization stages [12,34].

Though the availability of direct export functionalities in CAD is currently limited,⁵ open access source code bindings in C++, C#, and Python™ support the implementation of reading and writing functionalities of QIF files [80]. Since existing STEP AP242 and JT interfaces in FEA software typically only offer the possibility to exchange geometry, it is, in any case, necessary to establish a way to bring GD&T information into FEA. For this reason and due to the aforementioned strengths of QIF, QIF is selected over STEP AP242 and JT for variation simulation and is therefore used as the MBD standard to link CAD and FEA in this article.

After the 3D CAD models with semantically annotated GD&T information are exported as QIF, the first step to empower the FE model for variation simulation is to read and parse the files (see Fig. 2). QIF's normalized decoupled relationship model is beneficial for systematically navigating through the ASCII-based file, hopping from one element to another using the identifiers as linkages, and extracting all relevant information needed for the subsequent variation simulation steps.

3.2. Mapping information from QIF to FE meshes based on parameter spaces

After the GD&T information shared via QIF is extracted, it must be translated into variation information to set up the geometrical model for variation simulation (see Fig. 2). As discussed in Section 3.1, GD&T information is expressed via characteristics, features, and datum reference frames, interrelated and commonly linking to parametrically described curve or surface elements indirectly via the <EntityInternalIds> in the <*FeatureNominal> elements (see Fig. 3). In FEA, BREP geometry is not directly used. It must be translated into a discrete representation, a surface or volume mesh with a finite number of connected elements and nodes. As explained in Section 3, this article follows the strategy of manipulating the nominal mesh to model geometrical deviations (see strategy (b) in Fig. 1). Following this strategy requires that the GD&T information is already linked to the nominal FE mesh to bring variation into the simulation model (see Fig. 2). Otherwise, the user has to manually transfer this information

³ In this article, QIF is considered in its third version, QIF 3.0. For better readability, it is just named QIF.

⁴ The suffix Set indicates groups of “n = #” elements of the same type.

⁵ A few CAD software, for instance, ANSYS® Discovery™ SpaceClaim® [76] or Autodesk® Inventor® [77], recently started to offer direct QIF import and export functionalities. In addition, third-party translators, for instance, Capvidia MBDVidia [78] or Elysium [79], can be used to convert native CAD formats into QIF.

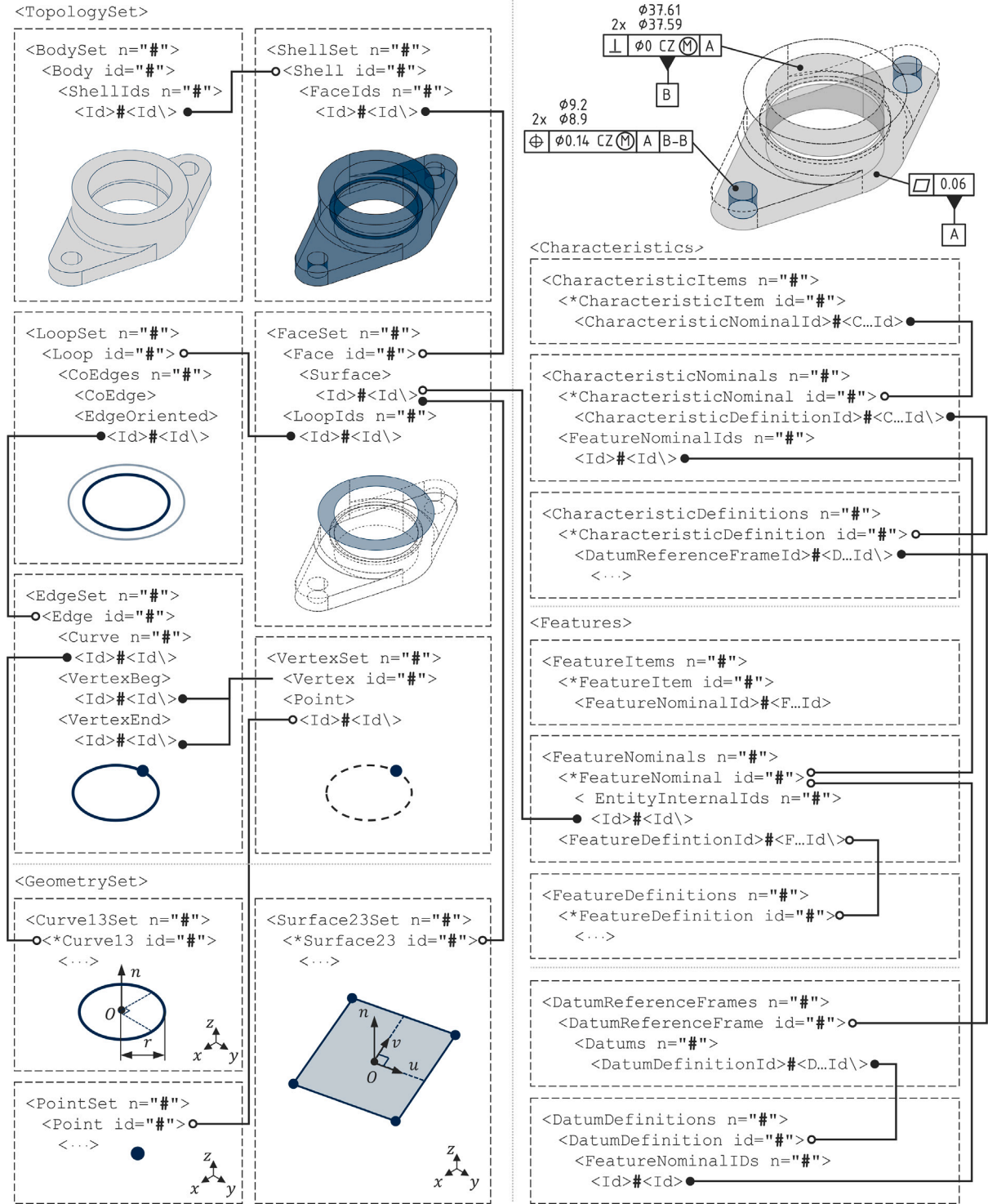


Fig. 3. Schematic illustration of the MBD information model in QIF carrying information on part topology, geometry, and GD&T annotations freely adopted from [34]. The prefix “*” marks substitution groups with different substitutes specified in XSDL.

to the respective portions of the mesh. A valid mapping strategy that automates this process has been presented in [34]. It uses point registration algorithms to automatically segment surface meshes into QIF faces as underlying geometry elements for the topological faces to inherit the link to the GD&T information [34]. Though it has proven its suitability, a new mapping strategy is introduced in this section. The method presented will bring additional value for the subsequent deviation modeling, discussed in Section 3.3, and is the core for an automated FE-based variation simulation.

In QIF, all parametric surfaces are described in a 2D parameter space $\Omega^0 \subseteq \mathbb{R}^2$, where “each point is defined as a pair (u, v) ” [75] (see Fig. 4). All pairs (u, v) , lying inside lower and upper domain limits $u^{lb}/u^{ub}, v^{lb}/v^{ub}$, are mapped to the 3D Euclidean space $\mathbb{R}^3$ via a mapping function $S_j(u, v)$, where “each point is defined as a coordinate triplet (x, y, z) specified in a right-handed Cartesian coordinate system” [75]:

$$\mathbf{q} = S_j(u, v), \quad \mathbf{q} = (x, y, z). \quad (1)$$

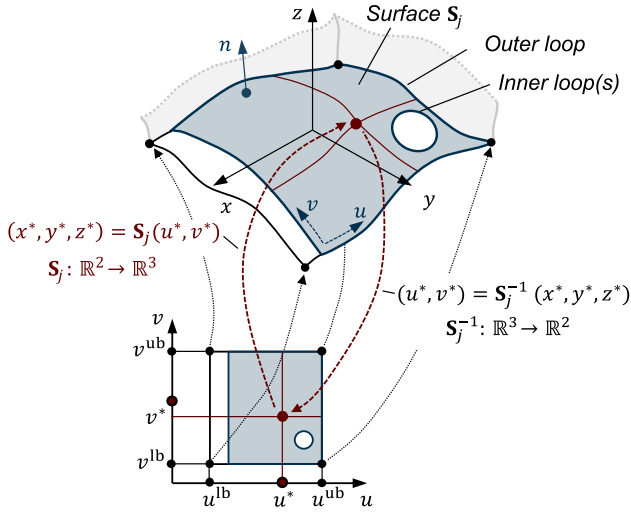


Fig. 4. Correspondence between 2D parametric and 3D Euclidean space in QIF to describe geometrical surfaces inspired by [75].

For this reason, QIF surfaces have the suffix ‘23’ in their name, such as <Plane23> or <Cylinder23>.

Inverting the mapping function Eq. (1) maps points from the Euclidean space back to the parameter domain (see Fig. 4):

$$(u, v) = S_j^{-1}(\mathbf{q}). \quad (2)$$

The inverse function in Eq. (2) can thus be used to verify whether a given point $\mathbf{q}^* = (x^*, y^*, z^*)$ in the physical space, for instance, a mesh node, belongs to a surface S_j . This relation is fundamental to the developed method of identifying QIF faces in FE meshes. The workflow consists of five main steps, explained in general in the following and illustrated in Fig. 5–9 for a planar face of the flange.

Step 1: Surface mesh extraction

The aim is to map QIF faces to the FE mesh surface. Hence, first, all surface elements F_Γ of a mesh $\mathcal{T}$, composed of a set of elements $\mathcal{K}$, need to be identified. This step is obsolete for shell meshes and optional for volume meshes, but reasonable for computing times since the nodes to be mapped are restricted to only mesh nodes on the surface.

A polyhedral FE volume element $K \in \mathcal{K}$ is characterized through a set of element nodes $\mathcal{N}_K$ and faces F_K [81] (see Fig. 5) and can be written as:

$$K = (\mathcal{N}_K, F_K), \quad F_K = \{F_i \subset \mathcal{N}_K \mid i = 1, \dots, I\}. \quad (3)$$

The node set $\mathcal{N}_K$ of an element K represents a subset of all mesh nodes $\mathcal{N}$:

$$\mathcal{N}_K = \{\mathbf{q}_j \in \mathcal{N} \mid j = 1, \dots, J\}. \quad (4)$$

The number of FE faces defines the element type and the number of nodes. Common elements are tetrahedrons, $|F_K| = 4$, $|\mathcal{N}_K| = 4$, and hexahedrons, $|F_K| = 6$, $|\mathcal{N}_K| = 8$ [82].

A face F of K is a surface element, i.e., part of the boundary ∂K , if not shared with any other element K' within the mesh $\mathcal{T}$ [83]. The set of all surface element faces is thus defined as:

$$F_\Gamma = \bigcup_{K \in \mathcal{T}} \{F \in \partial K \mid \forall K' \in \mathcal{T}, K' \neq K \Rightarrow F \notin \partial K'\}. \quad (5)$$

The set of surface nodes $\mathcal{N}_\Gamma$ to be considered for the mapping is thus given as (see Fig. 5):

$$\mathcal{N}_\Gamma = \bigcup_{F \in F_\Gamma} F. \quad (6)$$

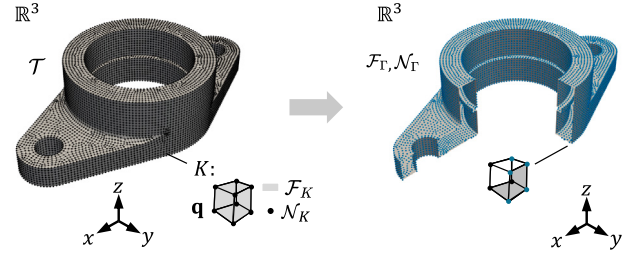


Fig. 5. Mapping surface and GD&T information from QIF on FE meshes. Step 1: Surface mesh extraction.

Step 2: Inverse mapping of FE mesh nodes

The identified set $\mathcal{N}_\Gamma$ is now mapped from $\mathbb{R}^3$ to the QIF surface parameter space $\mathbb{R}^2$ using the inverse of the parametric surface equations as introduced in Eq. (2). A closed-form solution for the inverses exists for all QIF primitives with regular shapes, such as planar, cylindrical, conical, spherical, and toroidal surfaces. Appendix A summarizes the inverse mapping functions of primitives with the corresponding QIF elements as inputs.

For more irregular shapes, represented as non-uniform rational basis spline (Nurbs) surfaces, finding analytic solutions for the inverse S_j^{-1} is complex, and, if possible, depending on the degree of the surface [84]. Therefore, inverse evaluations are usually performed numerically to find a solution within a desired tolerance [84]. Numerical optimization maps the nodes to a Nurbs parameter domain by finding a pair (u^*, v^*) , which mapping $S_j(u^*, v^*)$ to $\mathbb{R}^3$ is sufficiently close to a mesh node $\mathbf{q}$ [85]:

$$(u^*, v^*) = \arg \min_{(u,v) \in [0,1]^2} \|S_j(u, v) - \mathbf{q}\|^2. \quad (7)$$

The non-linear optimization problem in Eq. (7) is solvable with standard algorithms such as Newton–Raphson–Method [85]. As a result of Step 2, all mesh nodes are represented in the parameter space of surface j (see Fig. 6). After step 2, each surface mesh node is positioned within the parameter space of the surface S_j in focus.

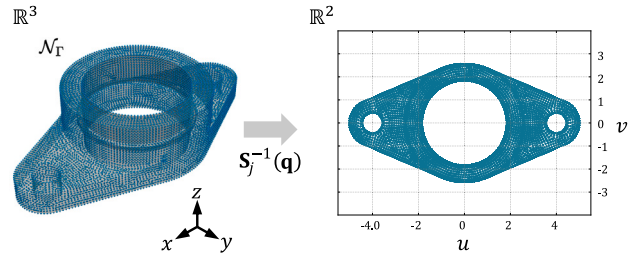


Fig. 6. Mapping surface and GD&T information from QIF on FE meshes. Step 2: Inverse mapping of FE mesh nodes.

Step 3: Geometrical proximity test

Even though all mesh nodes can be represented within a parameter space of one surface, it does not mean they are part of the surface. A projected node (u^*, v^*) lies only on the surface $S_j(u, v)$ when it is close to the original point $\mathbf{q}^*$ represented in the 3D space. Mathematically spoken, this means that the geometrical orthogonal distance d along the surface normal $\mathbf{n}(u^*, v^*)$ between the original point $\mathbf{q}^*$ and the node projected to 2D and back to 3D via $S_j(u^*, v^*)$ needs to be sufficiently small, i.e., smaller than a reasonable numerical tolerance ϵ [86,87]:

$$d(\mathbf{q}) = |(\mathbf{q} - S_j(u, v)) \cdot \mathbf{n}(u, v)| < \epsilon. \quad (8)$$

Only nodes that hold the condition in Eq. (8) are spatially close to the surface and considered in Step 4 (see Fig. 7).

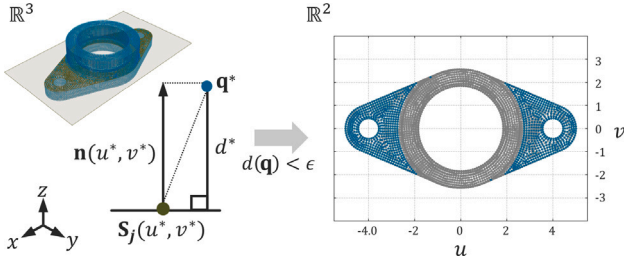


Fig. 7. Mapping surface and GD&T information from QIF on FE meshes. Step 3: Geometrical proximity test.

Step 4: Trimmed domain test

The surface parameter space is defined by its domain Ω_j^0 , rectangular shaped by the boundaries of (u, v) [75,86]:

$$\Omega_j^0 = \{(u, v) \mid u_j^{\text{lb}} \leq u \leq u_j^{\text{ub}}, v_j^{\text{lb}} \leq v \leq v_j^{\text{ub}}\}. \quad (9)$$

Ω_j^0 does not necessarily have to be, but might further be constrained by an outer boundary Γ_0 and inner boundaries Γ_i , denoted as Ω_j . The boundaries are on the topology level represented as loops, composed of multiple edges (see Fig. 3), trimming the surface to its final physical shape [75].

On the geometry level, a closed loop is represented by a union set of connected, non-intersecting curve segments γ , for instance, linear segments, circular or conic arc curves, or splines [86]. Curve segments are in QIF described in a 1D parameter space, linked to the 3D Euclidean space, and referenced by the <Face> element [75]. The 3D curves can be represented in the surface 2D parameter space through a number of discrete points mapped to Ω_j^0 via S_j^{-1} . Hence, a face's outer Γ_0 and inner boundaries Γ_i are defined within the surface parameter domain $\Omega^0 \subset \mathbb{R}^2$ as follows:

$$\Gamma_{0/i} = \bigcup_{o=1}^n \gamma_{0/i,o}, \quad o = 1, \dots, n, \quad (10)$$

where i is the index of the inner boundary and o is the index of the curve segments in total forming one boundary.

Thus, a mapped mesh node $(u^*, v^*) \in \Omega_j^0$ finally belongs to the face j if and only if it is within the trimmed domain Ω_j :

$$(u^*, v^*) \in \Omega_j = \text{region}(\Gamma_0) \setminus \bigcup_{i=1}^n \text{region}(\Gamma_i). \quad (11)$$

The condition given in Eq. (11) can be expressed as a point-in-polygon (PIP) problem, where the task is to check whether a given point lies within a closed polygon [88]. PIP is a common problem in computational geometry that can be solved with the help of the even-odd rule or winding number algorithm [88]. As a result, only mapped points identified within the trimmed domain Ω_j are further considered in the final step, in total forming the subset $\mathcal{N}_{\Omega_j} \subset \mathcal{N}_{\Gamma}$ (see Fig. 8).

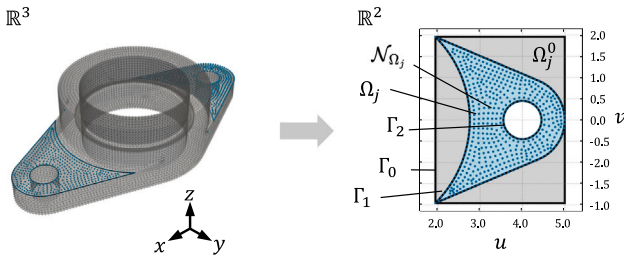


Fig. 8. Mapping surface and GD&T information from QIF on FE meshes. Step 4: Trimmed domain test.

Step 5: Element faces mapping

An element face F_K belongs to the QIF face (see Fig. 9) if all its nodes (linear tetrahedrons: 3, linear hexahedrons: 4 [82]) were identified in Step 4. Hence, a QIF face is represented in $\mathcal{T}$ through the identified set of element faces with their corresponding nodes:

$$\mathcal{F}_{\Omega_j} := \{F \in \mathcal{F}_{\Gamma} \mid F \subset \mathcal{N}_{\Omega_j}\}. \quad (12)$$

Repeating Steps 2–5 for all faces of a QIF body automatically splits the entire surface of a FE mesh $\mathcal{F}_{\Gamma}$ into subsets of element faces, and each element face is uniquely mapped to only one QIF face:

$$\mathcal{F}_i \cap \mathcal{F}_j = \emptyset; \quad \forall i, j = 1, \dots, |\mathcal{F}_{\Gamma}|; \quad i \neq j. \quad (13)$$

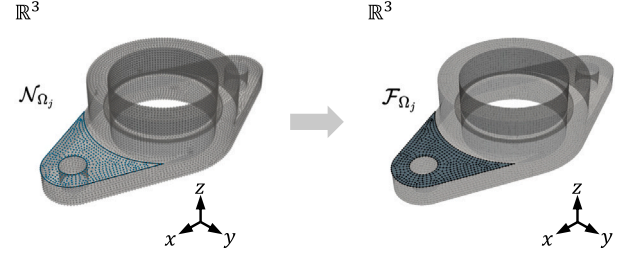


Fig. 9. Mapping surface and GD&T information from QIF on FE meshes. Step 5: Element faces mapping.

Fig. 10 illustrates the mapping results for the entire flange. The proposed method is based on the premise that the coordinate system of the part represented in QIF is aligned with the mesh's coordinate system. However, this is guaranteed if the pre-mesh geometry is derived from the given QIF file or another derivative of the same initial model. The mapping is independent of the mesh size or quality, which will only impact the computing time needed.

Since the meshed faces are associated with the surfaces, and the features linked to these surfaces are coupled to the characteristics (see Fig. 3), the corresponding surface points and elements are indirectly connected to the characteristics and inherit the GD&T information from the QIF file.

Besides systematically transferring the GD&T information from QIF to the FE mesh, part geometry is now represented in the 3D Euclidean and 2D parameter space. A hybrid representation model was created along the way and will prove helpful in the next simulation step.

3.3. Generating deviated FE meshes based on QIF MBD and GD&T information

After mapping the required information on the mesh, it must be translated into a set of deviations to analyze the assembly response (see Fig. 2). Following strategy (b) from Fig. 1, deviations are generated by applying displacements to the nominal node position $\mathbf{q}_i$, creating a shift to its deviated position $\mathbf{q}_i'$. Hence, it is necessary to identify an individual displacement vector δ_i for each node i . First, this implies that the respective portions in the mesh are known. The mapping method presented in Section 3.2 already segmented the nominal mesh into its QIF face elements and linked the required information to the respective mesh portions and their nodes to model the deviations (see Fig. 2). Second, the set of deviated nodes in total needs to represent a feature deviation within the tolerance zone.

Modeling deviations on feature level

Various methods for modeling deviations through points and meshes have been proposed in the literature over the years, primarily within the context of SMS. Deviations are modeled face by face and recombined at the end to represent a part with deviated geometry throughout the entire mesh [73]. SMS aim to realistically model the shapes of the parts through systematic and random deviations applied to the

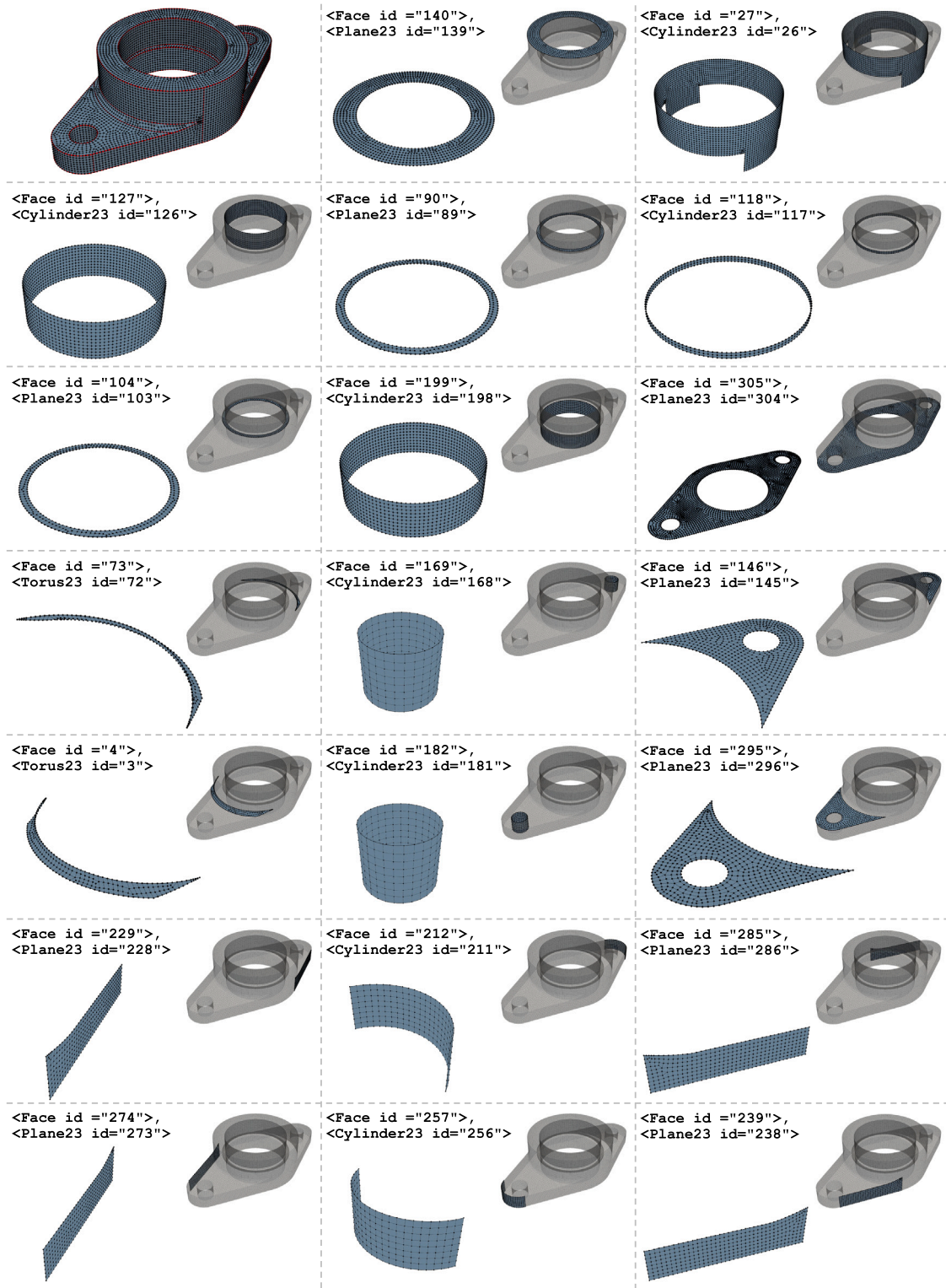


Fig. 10. Mapping results for the flange part from Fig. 1. Each QIF face is linked to a set of FE mesh nodes and elements as subsets of the nominal FE mesh.

individual surfaces while assuring that they, in total, lie within the tolerance zones of the features [72]. The Small Displacement Torsor (SDT) concept is one potential way to control the location and orientation deviations of the meshed surfaces [89]. The SDT concept, introduced in the 1990s [90], is well-established in feature-based

variation simulation, for instance, used in the Unified Jacobian-Torsor Model [91], is implemented in proprietary CAT software [92–94] and has shown its potential as the basis for an automated derivation of feature deviations from PMI annotations communicated through STEP AP242 [36]. Assuming that deviations are comparatively small to

the nominal dimensions, the deviation of a feature can be described by two sets of vectors, comprising three rotational and three translational entries in correspondence to the six potential degrees of freedom of a rigid body [17,89,95]:

$$\tau = (\mu, \nu, \omega, \alpha, \beta, \gamma) \quad (14)$$

The tolerance zone represents the space of all acceptable deviations from the ideal feature, and its shape depends on the feature type and the specified tolerances [96]. Thus, it defines geometrical constraints on the torsor, leading to correlations between their entries and limiting their magnitude [89]. The SDT concept is used in the following to translate the tolerance specifications into nodal displacements, exploiting the established semantic link between GD&T information, QIF faces, and meshed face elements, the discrete FE counterparts of the QIF faces (see Fig. 11).

First, the tolerance zone of the feature of interest is derived based on the information carried within the `<Feature>` and `<Characteristic>` elements, being compliant with the tolerancing standard defined in the `<Standard>` element. Within this tolerance zone, the meshed surface is allowed to deviate, where randomly sampled torsor screw values represent one feature deviation. Geometrical constraints ensure that the torsor complies with the tolerance zone. Fig. 11(a) illustrates the tolerance zone for a `<PlaneFeatureItem>` of the flange with a position tolerance t_p based on the QIF information shown at the top of Fig. 11.

The deviation is a combination of one translational displacement ω in z -direction as well as rotational displacements α, β around x - and y -axis of the local feature coordinate system [89]. The rotations α and β are dependent on t and the length and width of a and b of the plane. Mathematically, the deviation is defined as follows [89]:

$$\tau = (0, 0, \omega, \alpha, \beta, 0) \quad (15)$$

with:

$$-t/2 \leq (\beta \cdot x + \alpha \cdot y + \omega) \leq t/2 \quad (16)$$

$$\omega \in [-t/2; +t/2] \quad (17)$$

$$\alpha \in [-t/a; t/a] \quad (18)$$

$$\beta \in [-t/b; t/b] \quad (19)$$

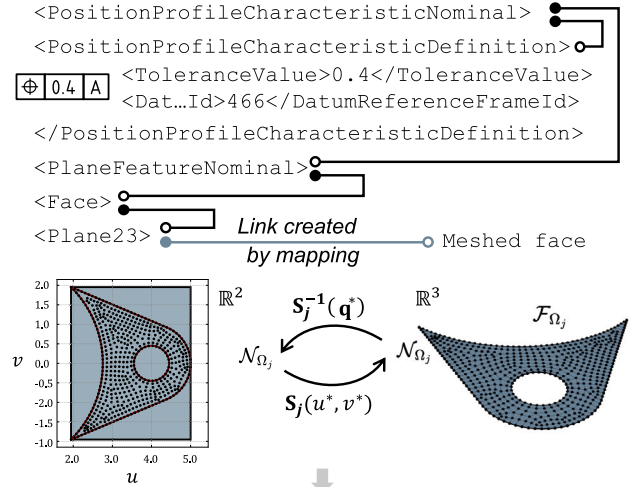
Mathematical descriptions for the zones of all standard geometrical tolerances specified to regular features, including additional information, such as material condition requirements, have been elaborated in the past and can be found in the literature, for instance, in [95,96].

In the next step, the SDT is used to modify the parametric description of the surface $S_j(u, v)$. For the given `<Plane23>` surface, the sampled translational displacement ω corresponds to the displacement of the plane's origin $\mathbf{p}_0$ (`<Origin>`), the rotational displacements α, β affect the two perpendicular in-plane direction vectors $\mathbf{d}_u, \mathbf{d}_v$ (`<DirU>`, `<DirV>`) (see Fig. 11b).

Finally, the deviations need to be mapped on the respective mesh portions. As a result of the inverse mapping, each element face F with its nodes is described in both the 2D surface parameter and the 3D Euclidean space (see Section 3.2, Step 2). Hence, the new positions of the nodes in the 3D Euclidean space can be derived by evaluating the modified function $S'_j(u, v)$. In doing so, a meshed face is translated and rotated, utilizing the sampled SDT entries, which ensures that all points lie within the specified tolerance zone. The node displacement δ_i corresponds to the vector pointing from the initial node position $\mathbf{q}_i = S_j(u_i, v_i)$ to the displaced one $\mathbf{q}'_i = S'_j(u_i, v_i)$ (see Fig. 11c):

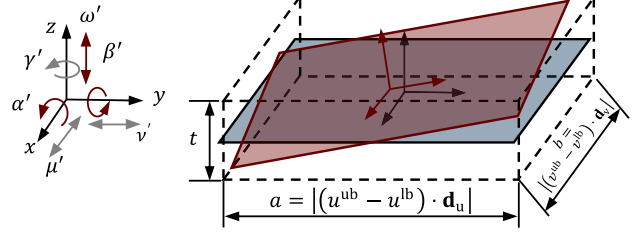
$$\delta_i = S'_j(u_i, v_i) - S_j(u_i, v_i). \quad (20)$$

The same logic can be used to model size deviations. Suppose size tolerances are applied to so-called features of size. In that case, they directly control the dimension of one feature, such as the diameter of a cylinder via a `<DiameterCharacteristicDefinition>`, or

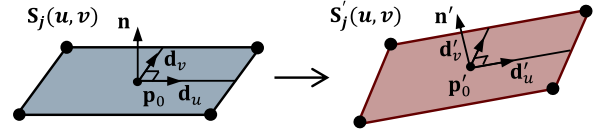


a) Sample SDT entries within the tolerance zone

$$\tau' = (0, 0, \omega', \alpha', \beta', 0)$$



b) Modify parametric description of surface function



c) Calculate nodal displacement vectors

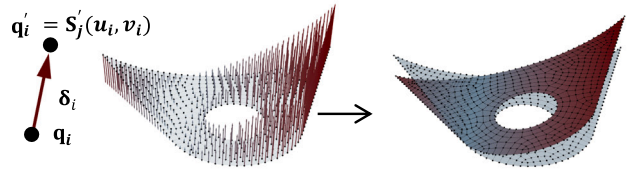


Fig. 11. Applying deviations through individual node displacements in the associated mesh subset exploiting the semantic link between the mesh, the QIF face, and the GD&T information established by the mapping presented in Section 3.2.

indirectly the distance between two elements, such as two opposing planes via a `<WidthCharacteristicDefinition>` [97]. Size deviations of cylindrical and spherical features of sizes can be covered by adjusting the underlying parameters, the intrinsic characteristics, of the surface function $S'_j(u, v)$, leading to an offset of the faces in the surface normal

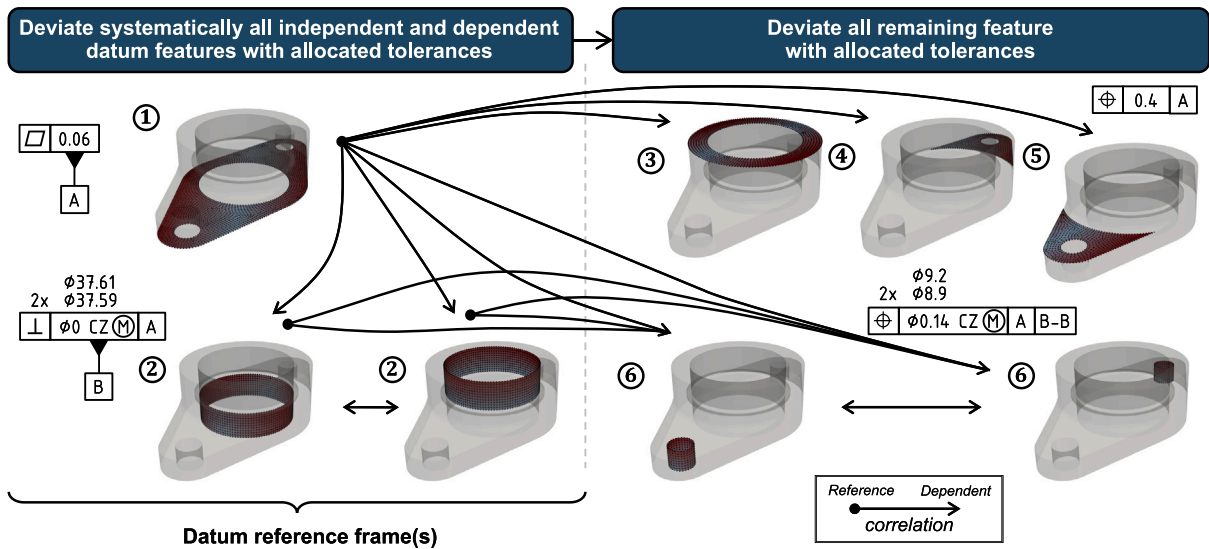


Fig. 12. Workflow to consider correlations and dependencies between the individual features when generating deviated FE meshes. Exemplarily illustrated for the tolerance specification of the flange given in Fig. 1.

direction at each point (u, v) .⁶ Size tolerances specified to two opposing planes control the distance between them and ambiguously define the location of the tolerance features. Thus, size variations can be modeled in different ways, for example, by offsetting one or both planes in their normal direction by modifying their nominal position $\langle \text{Origin} \rangle$. The deviated meshes are generated by feeding the resultant surface functions S' with the face nodes' u, v values.

When modeling deviations, correlations between the tolerance zones of a single feature and part features must be considered [40,98]. Applying multiple tolerances to one feature is common to control its size, location, orientation, and form. Multiple tolerances lead to correlated tolerance zones since a position tolerance, for instance, limits location, orientation, and form [98]. For example, additional orientation and form tolerances impose stricter limits on the feature's orientation and form, which must be considered when generating the deviated meshes to ensure conformity with all specified tolerances. At part level, correlations can occur when multiple features share a mutual tolerance zone (specified as common zone (CZ) within the feature control frame [16] and represented in QIF within the $\langle * \text{CharacteristicDefinition} \rangle$ or orientation and location tolerances reference to datum feature reference frames with a set of datums, deviating from their ideal [63]. For these reasons, the sequence of generating the deviated features matters. It is reasonable to first model the features that form the datum reference frames, starting with the independent datums and gradually generating the deviations for the others, considering their dependencies at both the part and feature levels. In case of the flange (see Fig. 1, top), the planar bottom face is independent and forms datum A, which is referenced by both cylindrical bores, sharing a common tolerance zone and defining datum B (see Fig. 12, left). After that, the deviations for all other tolerances referencing the datum reference frames can be derived considering the correlations on part and feature level (see Fig. 12, right). For the flange example, the position tolerances for both planar features are dependent on the deviated feature representing datum A, while the position tolerances of the two holes are dependent on the features linked to datum A and B, but also on each other since they share a mutual tolerance

zone (indicated by CZ in the control frame). In addition, the position tolerance and the size zone tolerance are coupled by the maximum material condition (M) (see Fig. 12, right). The clear structure in QIF and the links between the elements (see Fig. 3, right) help automatically navigate the part tolerance specification and systematically generate the correlated mesh deviations. The sequence can thus be derived automatically.

Following the proposed strategy, size, location, and orientation deviations can automatically be mapped onto mesh portions, assuming an ideal form of the surfaces. If non-ideal shapes, deviating from their nominal shape within form tolerance zones, are to be represented, more sophisticated shape modeling methods are to be integrated into the proposed method. This article is limited to size, location, and orientation deviations and does not study how to generate FE meshes with form deviations based on QIF. One promising solution can be to calculate an additional set of node displacements and to superimpose them with the ones for location, orientation, and size, as proposed in [89,99].

Combining deviated features on part level

However, the decomposition of deviation modeling into a set of feature-wise translations of the GD&T information into deviated mesh surfaces has the drawback that individual surfaces must be recombined into a single mesh of each part. The emerging challenges are gaps resulting from shared edges, intersections [73], and degenerated meshes since only surface nodes at the feature level are affected by node displacements (see Fig. 13, top). The former is a known issue in SMS modeling and is resolved by local or global methods for mesh regularization [73]. Local methods conduct the regularization after adding the deviation to the nominal model [72]. Global methods combine deviation addition and regularization into one step [73]. This article proposes a two-step local- and global mesh regularization procedure.

The sampled node deviations for more than one face must be accumulated at the connecting edges. In the given example, the planar and the cylindrical mesh portions are sampled individually, leading to two sets of displacement vectors for the edge nodes $\mathcal{N}_{\Omega_i} \cap \mathcal{N}_{\Omega_j}$. Though deviations are small compared to the nominal dimensions, the principle of vector addition is not applicable [73]. Instead, new common edges as an intersection between the deviated features need to be obtained [73]. Neglecting shape errors, the intersection lines can be derived analytically with the help of mathematical standard operations on the updated descriptions for the involved surfaces S' (see the red-colored edges for the cylindrical and planar face of the example

⁶ These types of features of size are shape-invariant under offsetting [97]. Offsetting the surfaces leads to the same result as scaling the features [97]. Hence, modifying the radius or diameter of the QIF surfaces corresponds to a scaling operation, resulting in a shape-invariant offset of the discrete mesh portion.

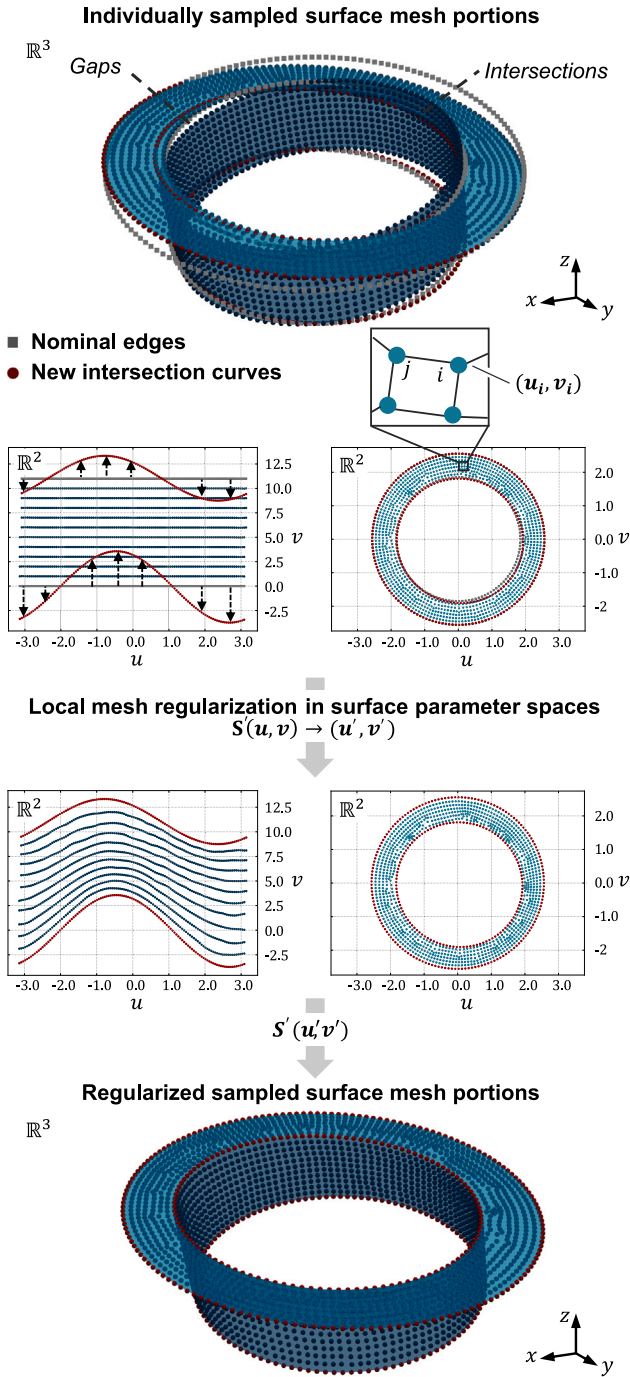


Fig. 13. Regularization of degenerated meshes — local method applied on face level. Exemplarily illustrated for the top plane and cylindrical face of the flange. Deviations are exaggerated.

in Fig. 13). Depending on the deviations' magnitudes and the mesh density, solely updating the vertex positions can lead to intersected and stretched elements within the sampled surface mesh portions, which need to be resolved before further processing (see, for example, the parameter domain of the cylindrical feature in Fig. 13). Instead of considering the problem in the 3D Euclidean space, this step can be carried out in the 2D parameter domain. The nodes are thus only moved on the respective surface by deriving a new set of u', v' parameters, while preserving the 3D mesh connectivity. The boundary nodes of the face element define a set of fixed Dirichlet boundary constraints, and the nodes forming the new intersection curves function as nodal

displacements in the 2D parameter spaces, bringing the deformation of the mesh within the domain. The 2D mesh regularization problem can be solved, for instance, using Laplacian mesh regularization [100], spring analogy models [101], or filtering based on the center of gravity of neighboring nodes [102].

The resulting new u', v' parameters are finally fed back into the deviated surface functions $S'(u, v)$ (see Fig. 13, bottom), leading to the desired sampled deviation (see Eq. (20)) and an additional slight repositioning of the node on the surface of the respective mesh portions. It should be noted that the deviations are exaggerated in Fig. 13 for illustration. The deviations are comparatively small, and mesh deformations are small compared to the nominal mesh dimensions.

The nodal displacements are applied on the part's surface, but they also influence the mesh quality in the interior of the part. Thus, a second global regularization step is needed. For volume mesh regularization, an iterative [103] or direct [73] application of FEA analysis is common, depending on the existence of a one-to-one relationship between input and output mesh.

For variation simulation using QIF, just sampled nodes exhibit initial displacements. In contrast, nodes in non-deviated surface domains should stay consistent with the initial model's outer boundary. By excluding surface nodes, the optimization approach [104] satisfies these requirements by introducing the disadvantage of high-dimensional input fields for optimization and therefore increases computational effort for complex 3D models. As FEA models are used for mesh creation, all necessary information is available to perform part-level FEA analyses.

For this purpose, the set of all surface nodes $\mathcal{N}_\Gamma$ lying on the surface boundary ∂K of the mesh $\mathcal{T}$ is divided into sampled surface nodes $\mathcal{N}_\Gamma^s$ and non-sampled surface nodes $\mathcal{N}_\Gamma^{ns}$. The positions of the non-sampled surface nodes are not changed, avoiding random changes in their positions and thus violations of GD&T definitions. As a consequence, they are used as FEA displacement load by adding the nodal-based deviation vectors δ_i to the respective node coordinates $\mathbf{q}_i \in \mathcal{N}_\Gamma^s$. Each non-sampled surface node $\mathbf{q}_i \in \mathcal{N}_\Gamma^{ns}$ is constrained such that movement in the node's normal direction $\mathbf{n}_i$ is locked and just in-plane corrections are allowed (see Fig. 14). The normal vectors $\mathbf{n}_i$ for each node i are obtained mesh-based by taking the normalized mean of all normal vectors $\mathbf{n}_k$ of the surfaces $F_k^{\Gamma^{ns}}$ adjacent to one node [105,106]:

$$\mathbf{n}_i := \frac{\sum_{k \in F_\Gamma^{ns}} \mathbf{n}_k}{\left\| \sum_{k \in F_\Gamma^{ns}} \mathbf{n}_k \right\|} \quad \forall \mathbf{q}_i \in \mathcal{N}_\Gamma^{ns}. \quad (21)$$

As edge nodes are adjacent to at least two surfaces, they are locked in two directions, allowing only movement in the tangent direction of the edge. After solving the linear system of equations, the displacement vector for each volume mesh node is used to update the node positions in the following computation of the actual behavior model. Fig. 14 illustrates the global FEA method applied to the flange part as a second mesh regularization step.

The method's limitations arise at edges between all sampled and non-sampled features because the displacement vectors of their nodes are strictly taken as load without further degrees of freedom. If not considered during local mesh regularization, this can lead to gaps and, thus, unacceptably high element deformations and incomputable FE models.

3.4. Assembling deviated part meshes

So far, the previous sections showed how deviated meshes can automatically be generated using QIF. To propagate the deviations from the part to the assembly level, the deviated volume meshes for each part obtained by the regularization approach must be brought into contact with each other (see Fig. 2). As the part meshes are defined in their local coordinate systems to relate them to the corresponding face and feature definitions in QIF, they must first be transformed into the global coordinate system and second virtually assembled according

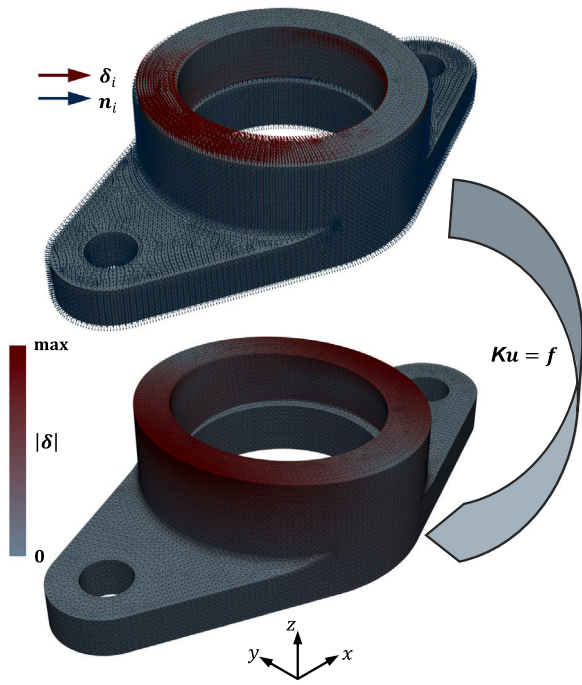


Fig. 14. Regularization of degenerated meshes — global FEA method exemplarily applied on part level by using the displacements and constraints visualized by the vectors in the upper part to obtain the deformed lower part. Deviations are exaggerated.

to given assembly sequences and conditions. Suitable methods to bring deviated meshes in contact have been proposed in the SMS context, such as in [107–111], and can be used for FEA-based variation simulations. Simulating assembly processes under geometry uncertainties is a research field by itself, and a detailed discussion would go beyond the article's scope and shift its focus. Nevertheless, the following brief insights highlight the importance of the assembly process simulation step.

Assembly process simulation

Bringing two parts into contact requires defining a sequence of assembly moves, characterized by joints with mating surfaces and conditions (e.g., fixed, sliding, or floating [109]), that sufficiently reflects the assembly conditions given in reality but can still be handled in reasonable computing times. Suppose a second part, a pin, is mounted to the deviated flange from Fig. 14. First, the pin is guided and aligned cylindrically in the flange's hole before the two planar surfaces come into contact in the second step. It aligns itself with the planar top face (see Fig. 15). Mimicking this assembly situation can be solved in different ways. The first step can, for instance, be handled by the absolute orientation method, determining the transformation between two point sets by a least sum of squares approach of the distance between each pair of points in the two point sets [112]. The derived transformation is applied to the moving point set, i.e., the pin. However, if a one-to-one relationship between the points is unknown, the Iterative Closest Point method [113] (ICP) is needed, determining the nearest neighbors in both mesh portions by a closest point minimization in each iteration.

The difference surface method is helpful to consider assembly directions along a vector $\mathbf{w}$, [107]. If the assembly direction is not equal to the general z -direction $\mathbf{w}_z$ of the global coordinate system, a rotation is conducted, such that the nodes are in the coordinate system x_w, y_w, z_w and $\mathbf{w} = \mathbf{w}^z$ (see Fig. 15). For each node pair, the distance vector $\mathbf{d}_{ij}$ is computed in the transformed coordinate system and projected onto the z -axis, which leads to the vectors $\mathbf{d}_{ij}^z$. In the given case, the assembly move is along the z -direction of the transformed coordinate system, and

form deviations are not considered. Therefore, the maximum value of $\mathbf{d}_{ij}^z$ defines the translation value in the translation vector $[0, 0, \max(\mathbf{d}_{ij}^z)]$, to be used for the second assembly move, realizing the planar contact between both parts (see Fig. 15).

In doing so, parts are virtually assembled step by step, mimicking the real assembly process. This may lead to over-constrained assemblies in the case of closed-loop contact chains. In combination with part deviations and due to resulting geometrical gaps in the assembly, “a strain and stress state in the parts” [114] may occur, affecting the resulting geometrical and non-geometrical KCs of the assembly. Additional effects occurring during the assembly, such as contact forces from the joining process, should be included if they have a significant impact on the assembly quality.

An initial FEA load step needs to be computed to consider the effect of the assembly on the KCs. It is conducted using the same material definitions as defined for the nominal FEA model. The load is the displacement caused by closing the contact chain to obtain the overall assembly model, viz., closing the last gap in the assembly. Regarding fixation for a properly defined FE model, the same boundary conditions as in the actual load case can, but do not necessarily have to be used, and instead, can be replaced by user-defined settings.

Assembly behavior simulation

Having obtained the assembled mesh and pre-stresses caused by assembly, the actual load case is to be computed. Thus, solving the FEA model finally simulates the behavior of the assembly in use, composed of the meshed parts with geometrical and dimensional deviations. For the flange-pin assembly, the structural behavior under operation with an external load acting on the pin might be of interest. Repeating these steps for a representative number of deviated assemblies scales deviation simulation up to variation simulation and enables the statistical evaluation of quality-critical KCs (see Fig. 2).

It must be noted that QIF, at least in its current version, is not designed to carry any information on assembly constraints and conditions — simply the nominal part positions in the global assembly coordinate system are preserved [34]. Hence, additional information on the assembly conditions and the KCs must be provided manually and separately along the variation analysis simulation chain.

4. Application and discussion

The following sections present an application of the developed FEA-based variation simulation. Besides illustrating the theoretically discussed aspects in Section 3 with an exemplary implementation into the FEA software ANSYS® (see Section 4.1) and its application to a practical use case (see Section 4.2), the results form the basis for a concluding discussion on the advantages and disadvantages of the proposed method for its practical application (see Section 4.3).

4.1. Implementation

The workflow from Fig. 2 is implemented in Python™ (v3.12.9). ANSYS® (v.2025.R1) is used as FEA simulation software, providing a direct application programming interface (API) to Python via PyAnsys™ (v2025.1.3). PyAnsys can be well combined with the Python binding for reading QIF 3.0 files and extracting the necessary information (see Section 3.1), which is provided by the DMSC [115]. For CAD modeling, Autodesk® Inventor® Professional 2026 is used, offering an inbuilt translator into QIF 3.0. Ensuring an automated information flow, the exported QIF files shall function as the pre-mesh geometry to define the model in the ANSYS Workbench. Since ANSYS does not support a direct import of QIF 3.0 files, an additional, but automatic translation step to STEP AP242 via open cascade technology (OCCT) [116] is added to create a seamless workflow. However, this workaround is only needed to overcome the limited import functionalities and to generate the initial mesh. The core workflow is based only on QIF.

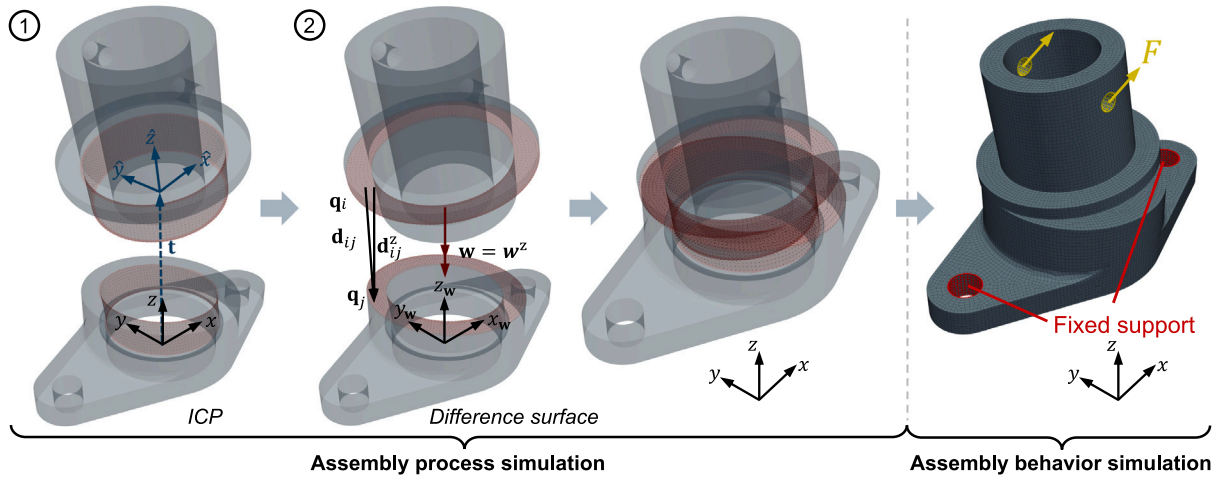


Fig. 15. Modeling the assembly behavior in two steps: Assembly process simulation of regularized, deviated meshes and behavior simulation of the virtually assembled product in operation. Deviations are exaggerated.

The mapping routine with its five steps presented in Section 3.2 is implemented in Python, based on existing code solving the PIP problem [117] and handling Nurbs curves and surfaces [118]. For the deviation modeling routines presented in Section 3.3, Monte Carlo Sampling is used, first sampling a value for the size of the deviation zone, and second filling that zone by breaking the sampled value into a set of displacement vector entries. The correlations are expressed through a set of geometrical constraints to conform to the sampled zone [47]. Furthermore, correlations between multiple tolerance zones applied to a single face element are considered in the sampling [119].

For the FEA setup, the load case created in ANSYS Workbench and defined as APDL (ANSYS Parametric Design Language) commands is exported via an MAPDL input file. As the files are text-based, the file is read only once with PyMAPDL (v.0.70.0) [120], converted to an ANSYS native database format, and stored for the following simulations. Local mesh regularization of the sampled surface mesh portions, presented in Section 3.3, is realized using the ARAP (as-rigid-as-possible) method [121]. For global mesh regularization, the normal-based boundary conditions are set up just once at the beginning of a simulation run and reused for each sample. In contrast, the deviation vectors for each sampled node are set for each sample. As data transfer from Python to MAPDL is slow, iterative commands are executed in a non-interactive fashion, such that commands are accumulated first in Python and then passed as a whole for execution to MAPDL. Gathering the regularized deviation information for each node on the part level, the nodes of the assembly model are translated by the regularized deviation vectors. For assembly process simulation (see Section 3.4), the sequence is set by node sets containing the surfaces to be related to each other, which are defined initially in ANSYS Workbench by named selections and transferred via the input file. Instead of combining the absolute orientation and the difference surface method for assembly sequence modeling, the absolute orientation approach is replaced by a generalized ICP framework [122] for point set registration of cylindrical features due to the availability of existing Python code. It combines the ICP and point-to-plane ICP, which differs from the standard ICP framework by a probabilistic model for determining the parts' transformation vectors. In addition, contrary to the difference surface method described in [107], the relationship between moving and fixed surface nodes is established based on a nearest neighbor search [123]. For point set registration of cylindrical features, the Python library Open3D [124] is used with the implementation of the generalized ICP algorithm [122]. With activated contact elements, the FEA solution is obtained for the assembly model, the KCs are determined, and PyMAPDL is reset to the initial state of the assembly model containing the nominal node positions.

4.2. Case study

The example in focus, a mounting wheel assembly, was initially introduced as a 2D tolerance allocation problem in [125] and extended to a 3D problem with GD&T specifications in [40]. Fig. 16 shows the design of the assembly, which is an adopted version of the one presented in [40]. The pin ①, on which the wheel ② sits with a clearance fit, is mounted on the two supports ③ and ④ with an interference fit. The bracket is composed of two supports ③ + ④ and a spacer ⑤, clamped with a bolted joint.

The assembly strategy, shown in Fig. 16, aims at an ideally constrained assembly. This status is achieved by assembling the spacer ⑤ and the supports ③ and ④ with a screw that is loosely fitted in the spacer's hole. Furthermore, the supports are not pre-stressed by closing the gap between the spacer and supports, for example, by using an elastic material for the spacer, avoiding assembly-induced strains and stresses. The conditions are met for simulation by excluding the spacer's elements from analysis and locking all degrees of freedom of the nodes in the supports' screw holes. The stepwise approach described in Section 4.1 is used for all assembly operations. The wheel is assembled such that point set registration ensures axial alignment with the pin, and the distance between wheel and support ③ is established with the difference surface method.

The used contact conditions are bonded contact between supports and spacer, frictionless contact at all other adjoint planar surfaces,

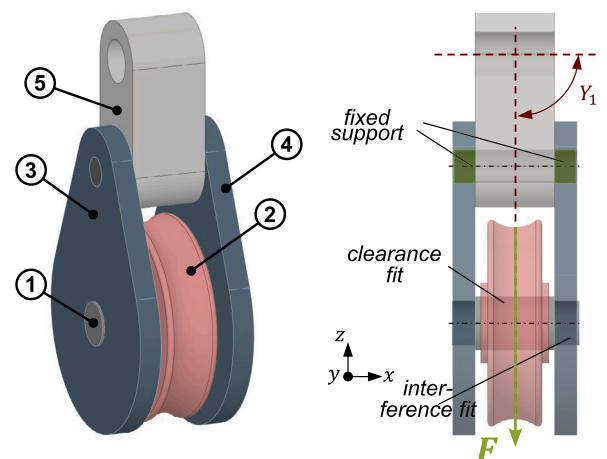


Fig. 16. Overview of the wheel mounting assembly case study.

and frictional contact with a coefficient of friction of 0.1 at cylindrical contact surfaces. The force load of 5000 N in negative z -axis direction is applied by a remote force at the geometric mean of the wheel, acting on the nodes of the wheel surface.

The tilting of the wheel related to the spacer's mounting hole axis around the y -axis is considered as KC and evaluated as a geometrical measurement Y_1 (Fig. 16, right). Therefore, two reference point sets on the wheel aligned in the nominal $y-z$ plane are used to calculate the mean normal vector for the point set, which is used to determine Y_1 . The assembly's mean von Mises stress within the interval $[235 \text{ MPa}; P_{99.99}]$ is considered the second, non-geometrical KC, denoted as Y_2 . The interval limits the stress to the region of interest, which is defined by the yield region as lower and the 99.99% stress percentile as upper bound. This is used to avoid potential singularities. The equivalent stress σ_e is computed by taking the average element von Mises stress obtained by the element node components. In 3D, σ_e is defined as a combination of the normal stresses $\sigma_{x/y/z}$ and shear stresses $\tau_{xy/yz/xz}$ [126]:

$$\sigma_e = \left(\frac{1}{2} [(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{xz}^2)] \right)^{1/2}. \quad (22)$$

The GD&T specifications for all parts conform to the ASME Y14.5–2018 and cover both dimensional and geometrical tolerances. The tolerances for the fits are based on the ISO 286-1:2010 and ISO 286-2:2010. The part specifications are visualized as 3D annotations in Tables B.3–B.4 in Appendix B and supplemented by a summary of the main relevant QIF information. The QIF files for the individual parts are uploaded as supplementary material.

4.3. Results and concluding discussion

The computational effort and the related times to solve the FE model are the bottlenecks in FEA-based variation simulation. Since the study intends to show the workflow rather than provide practical statistical results, a comparatively small sample size of $N = 100$ is chosen. In line with the small sample size, uniform distributions for all tolerances are assumed to cover the zones. The size tolerances are defined by the specified lower and upper limits LL, UL with a range of $t_i = |UL_i - LL_i|$, centered around its mean $LL_i + (UL_i - LL_i)/2$. For the geometrical tolerances, $LL = 0$ and $UL = t_i$ are given. The simulations are run on an average working system (CPU 13th Gen Intel® Core™ i5-13600, RAM: 32 GB, Cores used for simulation: 14).

Fig. 17 picks up the variation simulation workflow introduced in Fig. 2 and illustrates it for the wheel mounting assembly. The mapping algorithm could segment all part meshes into QIF faces, while all surface element faces were unambiguously allocated to a QIF face. All identified mesh portions could be automatically linked with the geometry and MBD information stored in QIF. Each color in Fig. 17, top, indicates a different QIF face. More details on the mapping and the part meshes are given in Tables B.3–B.4. Depending on the mesh density and number and type of QIF faces, computing times for the mapping ranged between 1.0 s and 12.3 s per part (see Table B.2). The mapping step is, however, not crucial since it must only be performed once, while the subsequent steps scale with sample size.

The time needed to automatically derive the feature parameters in the sampling step is negligible compared to the time needed to generate the deviated part meshes in each iteration. The local and global regularization took up most of the computing time, averaging 2.8 s to 17.3 s and 9.3 s to 22.8 s per sample. (see Table B.2). The proposed approach for mesh regularization results in slightly reduced mesh quality in the form of the scaled Jacobian ratio with a part-based mean of 0.586 to 0.606 for the sampled values compared to 0.609 to 0.632 of the nominal model (see Table B.1). Nevertheless, the resulting mesh quality is sufficiently good to compute the actual load case. The implemented

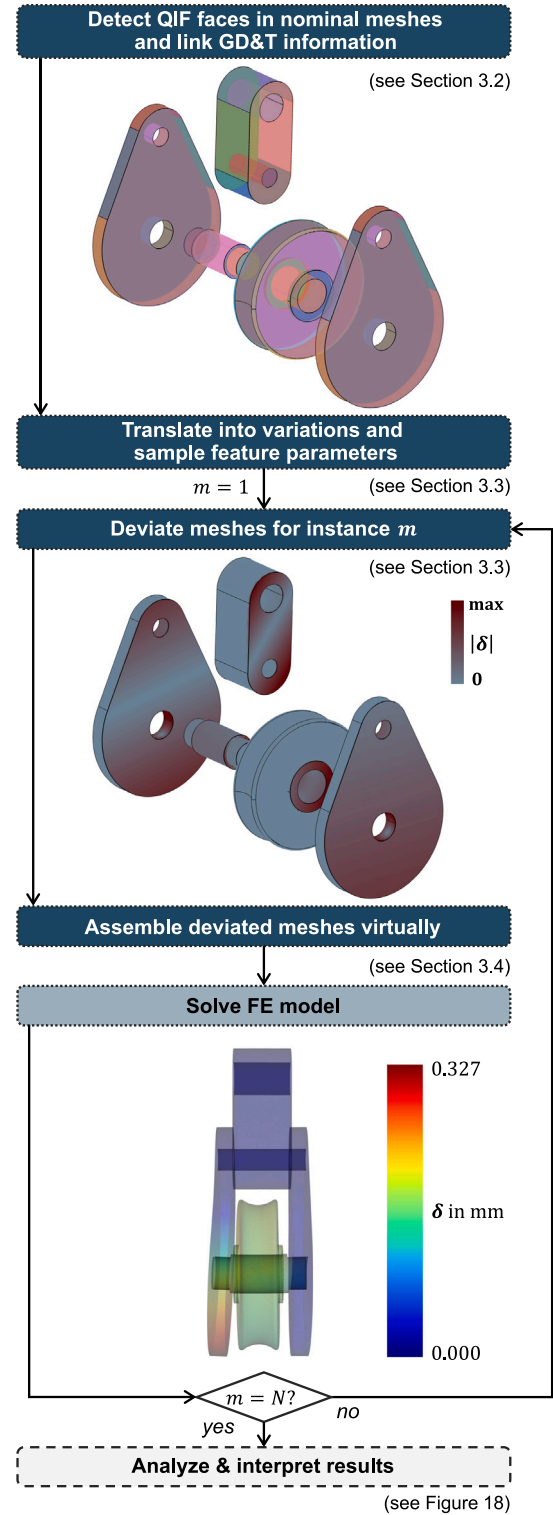


Fig. 17. QIF-based variation simulation branch of the workflow presented in Fig. 2 applied to the wheel mounting assembly. Meshes are suppressed for better visibility but visualized in Tables B.3–B.4.

regularization algorithms proved to be robust to resolve the occurring gaps and intersection issues when consolidating the deviated surface meshes into the volume mesh. Notably, FE meshing strategies aim to minimize issues, such as distorted elements and poor aspect ratios, and

achieve reliable stress results. However, meshes in non-intrusive FEA need also to account for deviation modeling. This is explicitly critical when modeling form deviations, as high density on the surface mesh is needed to represent small local surface deviations.

Iteratively solving the FE model as a black box model with varying part mesh inputs for all samples finally leads to the probabilistic response of the assembly to the variations. The results given in Fig. 18 indicate that the geometrical variations, limited by the allocated tolerances, significantly influence both the geometrical and non-geometrical KC in focus. The obtained frequency distribution of the angle Y_1 is centered around a mean of 90.12 deg with a total range of 2.06 deg and a coefficient of variation of 0.50%. In comparison, the coefficient of variation for the stress measure Y_2 with a mean of 427.74 MPa is 5.95% higher compared to Y_1 .

The study emphasizes that QIF can bridge the gap between CAD and FEA software, carrying the necessary information for an automated setup of the geometrical model for FEA-based variation simulation. The novel mapping technique allows for automatically transferring the information onto FE meshes, translating it into a set of sampled feature parameters to be finally used for generating meshed part instances with deviations conforming to the GD&Ts specified in the design phase. By utilizing the inherited feature characteristics and making use of the representation of points in the 2D parameter space, the SDT concept can be used for modeling of deviated meshes by indirectly modifying the parameters of the associated surface functions. Manual modeling steps are replaced, and a standard-compliant translation of GD&T information into nodal displacements is achieved. The presented MBD approach not only reduces the required effort within pre-processing, but it is also particularly beneficial for users with limited GD&T expertise, allowing them to perform variation simulations directly within their familiar FEA software. Consequently, the proposed QIF-based solution for information mapping extends the capabilities and strengths of FEA simulation to consider geometrical imperfections. While the geometrical model can be set up automatically, the assembly behavior modeling step still lacks automation. The user needs to manually develop a suitable strategy to realistically bring the actual assembly conditions into simulation.

Although the case study underlines the potential of QIF to leverage FEA for non-rigid variation simulation through MBD, future research is needed. Simulation efficiency was not the ultimate goal of the exemplary implementation of the method, and leaves potential for improvement. The pre-processing steps can be sped up by efficient programming and interface design. Still, the case study illustrates that solving the FE model is the primary contributor to the total computing time, hindering its practical application for large sample sizes. Although significant achievements have been made in recent years to speed up the computationally intensive parts of variation simulation, research is still needed to draw statistically reliable conclusions in computing times acceptable for its practical application, while realistically modeling the assembly behavior.

This article focused on size, location, and orientation tolerances; form tolerances were not further discussed. Their integration requires extending the SDT-based deviation modeling method presented in Section 3.3 by shape variation strategies developed for SMS, including a realistic representation of manufacturing signatures by numerical process simulation [23,127] as well as the integration of inspection data [128,129]. Although the inverse mapping strategy was developed for the most common geometry elements used in QIF, the final case study focused only on cylindrical and planar features for deviation modeling. Further work is needed to extend the deviation modeling to more complex geometries, such as Nurbs.

In addition to GD&T specification information, QIF is designed to carry inspection information to be appended as a list of feature measurements and optional measurement point sets to an existing QIF file and be used for FEA-based variation simulation, as illustrated in [42]. Hence, the proposed method offers potential to be extended to directly

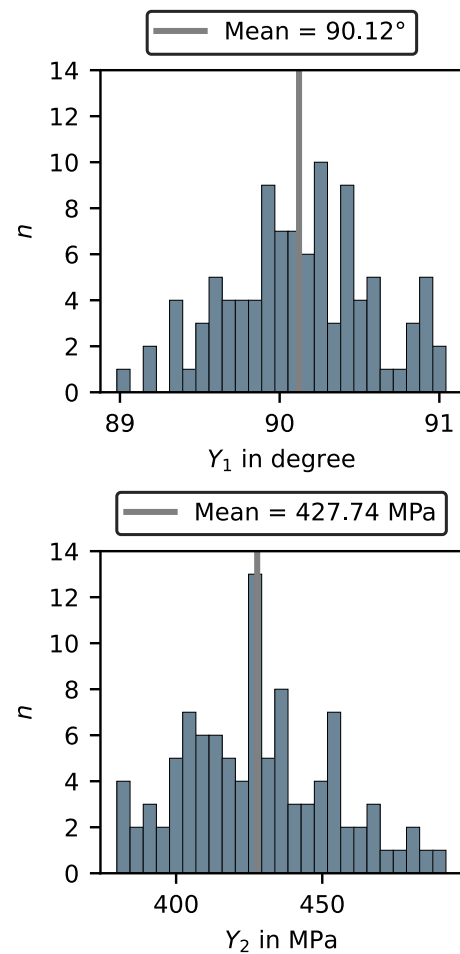


Fig. 18. Resulting frequency probability distributions for the tilting Y_1 and the equivalent stress Y_2 .

map actual measurement information onto FE meshes to perform digital twin applications [34].

Discrete representations, in particular meshes, are often used as an operationalization of the SMS concept [19]. For this reason, methods presented in the SMS context can be adopted for non-intrusive FEA, as it is illustrated in this article. In turn, the QIF-based mapping of GD&T information and deviation modeling in the 2D parameter spaces can be used in existing discrete SMS frameworks. The proposed method helps to automatically split the entire mesh into standardized features, to be either varied based on GD&T or inspection information. So far, MBD is rarely used in this context, mainly due to the difference between the geometry representation in CAD and variation simulation. The article provides new solutions for enhancing information sharing for SMS-based variation simulation frameworks.

A final note is that the method proposed is not restricted to a given FEA software – ANSYS was only exemplarily chosen in the application – and sub-steps, such as the novel mapping method, can also be used for SMS-based variation simulation. However, implementing an MBD solution for variation simulation into existing software requires external accessibility, at least realized through an API.

5. Summary and outlook

It is common to extend FEA to variation simulation to study the effects of part variations on product structural and thermal behavior. A novel MBD method was presented to automate setting up the mesh-based geometrical model, representing the geometrical variations

limited by the tolerances specified in the design phase. Using the QIF standard as an exchange format, a seamless link between CAD and FEA for variation simulation is set up, which allows the automatic mapping of geometry and tolerancing information on the respective FE mesh portions and generates deviated meshes based on the small-displacement torsor concept. The semantic link between the discrete and parametric face representations makes a manual parameterization of features and a manual transfer and translation of the GD&T information into the respective parameter values to represent the tolerance zones properly obsolete. The exemplary implementation in ANSYS illustrates the benefits of the proposed method for automating structural FEA while leaving the door open for its extension to other point-, mesh-, or voxel-based numerical simulation applications, such as CFD, where geometry is an essential contributor to product quality.

This article focused on size, location, and orientation tolerances. Further studies on communicating information about form tolerances and deviations through QIF and mapping this information onto FE-meshed features would help overcome this limitation. Besides, it is worth investigating how to represent non-geometrical uncertainties through MBD. Besides automating time-consuming pre-processing steps, research still needs to improve the computational efficiency of non-intrusive FEA to make statistical analysis and optimization valuable for practical use.

List of abbreviations

APDL	Ansys Parametric Design Language
API	Application Programming Interface
ARAP	As-rigid-as-possible mesh regularization method
ASME	American Society of Mechanical Engineers
BREP	Boundary Representation
CAT	Computer-Aided Tolerancing
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CFD	Computational Fluid Dynamics
DOE	Design of Experiments
DMSC	Digital Metrology Standards Consortium
FE	Finite Element
FEA	Finite Element Analysis
ICP	Iterative Closest Point
id	Identification designator
ISO	International Organization for Standardization
KC	Key Characteristic
MAPDL	Mechanical APDL
MBD	Model-Based Definition
Nurbs	Non-uniform rational basis spline
OCCT	Open CASCADE Technology (OCCT)
PIP	Point-In-Polygon
PMI	Product and Manufacturing Information
QIF	Quality Information Framework
SDT	Small Displacement Torsor
SMS	Skin Model Shapes
XML	Extensible Markup Language
XSDL	XML Schema Definition Language

List of symbols

a	Plane dimension in u -direction
b	Plane dimension in v -direction
d	Orthogonal surface distance of a point $\mathbf{q}$
$\mathbf{d}_0$	Start or prime meridian direction of cylinder, cone, or torus
$\mathbf{d}_{ij}$	Vector between two nodes i and j
$\mathbf{d}_{ij}^z$	Vector between two nodes i and j projected onto the z -axis
$\mathbf{d}_{u/v}$	In-plane direction vectors
D	Diameter

F	Element face
$\mathcal{F}$	Set of element faces
i	Counting index
j	Counting/surface index
k	Edge/face index
K	Polyhedral FE volume element
$\mathcal{K}$	Set of finite elements in a mesh $\mathcal{T}$
l	Height of cylinder
lb	Lower boundary
LL	Lower limit
m	Sample index
$\mathbf{n}$	Normal vector, cylinder's axis vector
N	Sample size
o	Curve segment index
$\mathcal{N}$	Node set
$\mathbf{p}_0$	Origin of a surface element
$\mathbf{q}$	Point or mesh node
$R_{x/y/z}$	Rotation matrix
s	Subscript to indicate that a node is sampled
$\neg s$	Subscript to indicate that a node is non-sampled
S	Parametric surface function
S^{-1}	Inverse parametric surface function
t	Tolerance
$\mathbf{t}$	Translation vector
$\mathcal{T}$	Volume mesh
u	Parameter space variable
$\hat{\mathbf{u}}$	Normalized parameter space vector
ub	Upper boundary
UL	Upper limit
v	Parameter space variable
$\hat{\mathbf{v}}$	Normalized parameter space vector
$\mathbf{w}$	Assembly direction vector
x	x -coordinate of $\mathbf{q}$
y	y -coordinate of $\mathbf{q}$
Y	Assembly response
z	z -coordinate of $\mathbf{q}$
α	Rotation around x -axis
β	Rotation around y -axis
γ	Curve segment
γ	Rotation around z -axis
Γ	Boundary
δ	Nodal displacement vector
δK	Boundary of an element K
ϵ	Threshold value
$\lambda_{u/v}$	Scaling coefficient in u/v parameter direction
μ	Translation in x -axis direction
ν	Translation in y -axis direction
σ_e	Equivalent, von Mises stress
$\sigma_{x/y/z}$	Normal stress components
ξ	Offset in the parameter space
τ	Small Displacement Torsor
$\tau_{xy/yz/xz}$	Shear stress components
χ	Inversion flag
ω	Translation in z -axis direction
Ω	Domain
Ω^0	Domain of surface parameter space

CRediT authorship contribution statement

Martin Roth: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jan Kopatsch:** Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Kristina Wärmefjord:** Writing – review & editing,

Supervision, Project administration, Methodology, Funding acquisition. **Rikard Söderberg**: Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition. **Stefan Goetz**: Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Funding: This work was carried out at the Wingquist Laboratory VINN Excellence Centre within the Area of Advance Production at Chalmers University of Technology and supported by Sweden's Innovation Agency (VINNOVA). The support of the project DigiSync (grant ID 2024-02505) is gratefully acknowledged.

Appendix A. Analytic inverse functions for regular QIF surfaces

This section provides the analytical inverse functions $(u, v) = S^{-1}(\mathbf{q})$ for five selected regular surfaces (see Figs. A.1–A.5). They map points from the 3D Euclidean space back to the 2D parameter space of a given parametric surface. The five regular geometry types are naturally associated with five of the seven invariance classes presented in the ISO GPS standards, giving information on which rigid-body transformation can be applied without losing symmetry [130]. The remaining two invariance classes can be associated with further QIF surfaces, such as Nurbs, which are not discussed in the following. The inverses are mathematically described in alignment with the original functions $\mathbf{q} = S(u, v)$ given in ISO 23952:2020 [75]. The symbols are further associated with the names of the corresponding QIF elements.

A.1. Plane surface

$$S^{-1}(\mathbf{q}) = ((\mathbf{q} - \mathbf{p}_0) \cdot \hat{\mathbf{u}}, (\mathbf{q} - \mathbf{p}_0) \cdot \hat{\mathbf{v}}) \quad (\text{A.1})$$

with:

$$\hat{\mathbf{u}} = \frac{\mathbf{d}_u}{\|\mathbf{d}_u\|} \quad (\text{A.2})$$

$$\hat{\mathbf{v}} = \frac{\mathbf{d}_v - (\hat{\mathbf{u}} \cdot \mathbf{d}_v) \cdot \hat{\mathbf{u}}}{\|\mathbf{d}_v - (\hat{\mathbf{u}} \cdot \mathbf{d}_v) \cdot \hat{\mathbf{u}}\|} \quad (\text{A.3})$$

where:

$\mathbf{p}_0$	Plane's origin	<Origin>
$\mathbf{d}_u$	Direction and scaling vector of u parameter lines	<DirU>
$\mathbf{d}_v$	Direction and scaling vector of v parameter lines	<DirV>

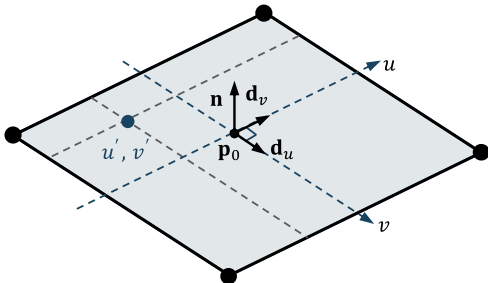


Fig. A.1. Plane surface description in line with [75].

A.2. Cylinder surface

$$S^{-1}(\mathbf{q}) = (u', v') \quad (\text{A.4})$$

with:

$$u' = \frac{\text{atan2}(\rho \cdot (\mathbf{n} \times \mathbf{d}_0), \rho \cdot \mathbf{d}_0)}{\lambda_u} \quad (\text{A.5})$$

$$\rho = \mathbf{q} - \mathbf{p}_0 - ((\mathbf{q} - \mathbf{p}_0) \cdot \mathbf{n}) \cdot \mathbf{n} \quad (\text{A.6})$$

$$v' = \begin{cases} \frac{(\mathbf{q} - \mathbf{p}_0) \cdot \mathbf{n}}{\lambda_v}, & \text{if } \chi_v = 0 \\ l - \frac{(\mathbf{q} - \mathbf{p}_0) \cdot \mathbf{n}}{\lambda_v}, & \text{if } \chi_v = 1 \end{cases} \quad (\text{A.7})$$

where:

$\mathbf{p}_0$	Cylinder's axis origin	<AxisPoint>
$\mathbf{n}$	Cylinder's axis vector	<Direction>
$\mathbf{d}_0$	Start direction of cylinder defining the cylinder seam at $u = 0$	<DirBeg>
l	Cylinder's height	<Length>
λ_u	Scaling coefficient in u parameter direction	scaleU
λ_v	Scaling coefficient in v parameter direction	scaleV
χ_v	Cylinder inversion flag	turnedV

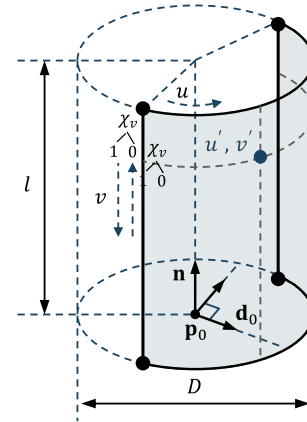


Fig. A.2. Cylinder surface description in line with [75].

A.3. Cone surface

$$S^{-1}(\mathbf{q}) = (u', v') \quad (\text{A.8})$$

with:

$$u' = \frac{\text{atan2}(\rho \cdot (\mathbf{n} \times \mathbf{d}_0), \rho \cdot \mathbf{d}_0)}{\lambda_u} \quad (\text{A.9})$$

$$\rho = \mathbf{q} - \mathbf{p}_0 - ((\mathbf{q} - \mathbf{p}_0) \cdot \mathbf{n}) \cdot \mathbf{n} \quad (\text{A.10})$$

$$v' = \begin{cases} \frac{(\mathbf{q} - \mathbf{p}_0) \cdot \mathbf{n}}{\lambda_v}, & \text{if } \chi_v = 0 \\ l - \frac{(\mathbf{q} - \mathbf{p}_0) \cdot \mathbf{n}}{\lambda_v}, & \text{if } \chi_v = 1 \end{cases} \quad (\text{A.11})$$

where:

$\mathbf{p}_0$	Cone's axis origin	<AxisPoint>
$\mathbf{n}$	Cone's axis vector	<Direction>
$\mathbf{d}_0$	Start direction of cone defining the cone seam at $u = 0$	<DirBeg>
l	Cone's height	<Length>
λ_u	Scaling coefficient in u parameter direction	scaleU
λ_v	Scaling coefficient in v parameter direction	scaleV
χ_v	Cone inversion flag	turnedV

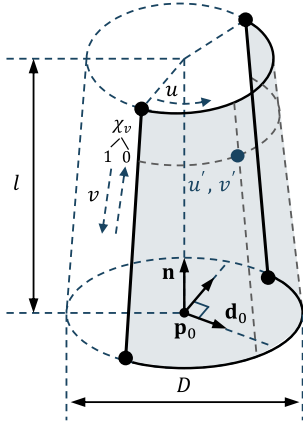


Fig. A.3. Cone surface description in line with [75].

A.4. Sphere surface

$$\mathbf{S}^{-1}(\mathbf{q}) = (u', v') \quad (\text{A.12})$$

with:

$$u' = \frac{u}{\lambda_u} = \frac{\text{atan2}(\rho \cdot (\mathbf{n} \times \mathbf{d}_0), \rho \cdot \mathbf{d}_0)}{\lambda_u} \quad (\text{A.13})$$

$$\rho = \mathbf{q} - \mathbf{p}_0 - ((\mathbf{q} - \mathbf{p}_0) \cdot \mathbf{n}) \cdot \mathbf{n} \quad (\text{A.14})$$

$$v' = \begin{cases} v \cdot \lambda_v, & \text{if } \chi_v = 0 \\ -v \cdot \lambda_v, & \text{if } \chi_v = 1 \end{cases} \quad (\text{A.15})$$

$$v = \arcsin\left(\frac{(\mathbf{q} - \mathbf{p}_0) \cdot \mathbf{n}}{\|\mathbf{q} - \mathbf{p}_0\|}\right) \quad (\text{A.16})$$

where:

$\mathbf{p}_0$	Sphere's center	<Location>
$\mathbf{n}$	North pole direction vector	<DirNorthPole>
$\mathbf{d}_0$	Prime meridian direction vector of sphere at $u = 0$	<DirMeridianPrime>
λ_u	Scaling coefficient in u parameter direction	scaleU
λ_v	Scaling coefficient in v parameter direction	scaleV
χ_v	Sphere inversion flag	turnedV

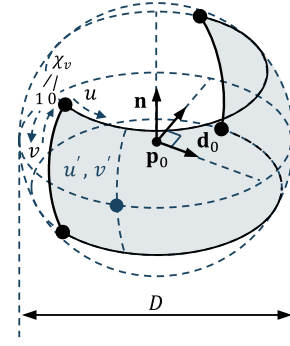


Fig. A.4. Sphere surface description in line with [75].

A.5. Torus surface

$$\mathbf{S}^{-1}(\mathbf{q}) = (u', v') \quad (\text{A.17})$$

with:

$$u' = \frac{u}{\lambda_u} = \frac{\text{atan2}(\rho \cdot (\mathbf{n} \times \mathbf{d}_0), \rho \cdot \mathbf{d}_0)}{\lambda_u} \quad (\text{A.18})$$

$$\rho = \mathbf{q} - \mathbf{p}_0 - ((\mathbf{q} - \mathbf{p}_0) \cdot \mathbf{n}) \cdot \mathbf{n} \quad (\text{A.19})$$

$$v' = \begin{cases} \xi_v + v \cdot \lambda_v, & \text{if } \chi_v = 0 \\ \xi_v - v \cdot \lambda_v, & \text{if } \chi_v = 1 \end{cases} \quad (\text{A.20})$$

$$v = \text{atan2}(\mathbf{r} \cdot \mathbf{n}, \mathbf{r} \cdot \mathbf{c}) \quad (\text{A.21})$$

$$\mathbf{r} = \mathbf{q} - \mathbf{p}_0 + \frac{D}{2} \cdot \mathbf{c} \quad (\text{A.22})$$

$$\mathbf{c} = \cos(u) \cdot \mathbf{n} + \sin(u) \cdot (\mathbf{n} \times \mathbf{d}_0) \quad (\text{A.23})$$

where:

$\mathbf{p}_0$	Torus' axis origin	<AxisPoint>
$\mathbf{n}$	Torus' axis vector	<Direction>
$\mathbf{d}_0$	Prime meridian direction vector of torus at $u = 0$	<DirMeridianPrime>
D	Torus' major diameter	<DiameterMajor>
λ_u	Scaling coefficient in u parameter direction	scaleU
λ_v	Scaling coefficient in v parameter direction	scaleV
ξ_v	Offset in v direction of the parametric space	offsetV
χ_v	Torus inversion flag	turnedV

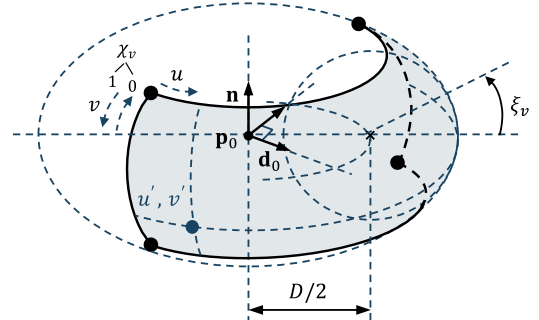


Fig. A.5. Torus surface description in line with [75].

Table B.1

Part-based mesh and quality metrics.

Metric	Part 1	Part 2	Part 3	Part 4	Part 5
Mesh Size in mm	2	2.5	2.5	2.5	5
Contact Mesh Size in mm	1	1	1	1	3
Form Function Degree	2	2	2	2	2
Element Type	Solid187	Solid187	Solid187	Solid187	Solid187
Nominal Mean Quality	0.631	0.620	0.632	0.632	0.609
Nominal Std Dev	0.177	0.180	0.173	0.173	0.174
Sample Mean Quality	0.595	0.591	0.606	0.606	0.586
Sample Std Dev	0.175	0.182	0.174	0.174	0.173

Table B.2

Computing times for the presented case study.

	Computing times in s				
	Part 1	Part 2	Part 3	Part 4	Part 5
Reading QIF ^a	0.426	0.527	0.474	0.601	0.488
Translating QIF to STEP AP242 ^a	0.014	0.010	0.011	0.026	0.025
Meshing ^a	4.145	4.693	2.808	2.912	1.455
Mapping ^a	12.259	4.881	13.455	5.136	1.004
Sampling ^b	0.003	0.002	0.002	0.003	0.007
Local mesh regularization ^b	17.277	5.525	10.802	8.247	2.781
Global mesh regularization ^b	22.714	20.144	17.173	16.211	9.334
Assembly					
FEA	492.623				

^a Performed only once, total time effort.^b Performed for each sample, average time effort per sample.

Appendix B. Details on the application

Details on the application of the proposed method, the case study, and the results, presented and discussed in Section 4, are summarized in the Tables B.1–B.4.

Appendix C. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.cad.2025.104003>.

Table B.3

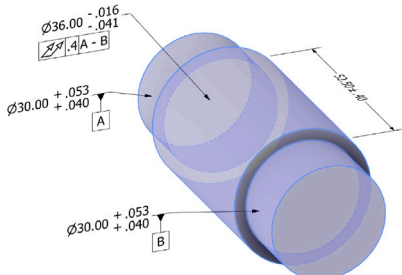
Summary of part tolerance specifications and mapped information on the FE meshes for the case study presented in Section 4. Tolerances are in mm.

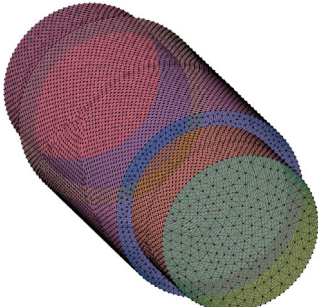
Part 1: Pin

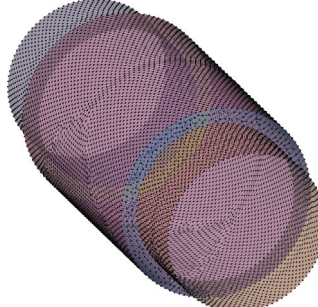
CAD model with PMI

Surface mesh split into QIF faces

Mesh faces with mapped tolerance or datum information







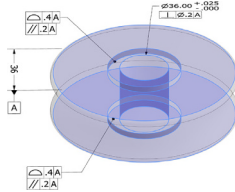
Associated surfaces		Associated characteristics		Associated features		Datums	Volume mesh		
Name	Count	Name	Count	Name	Count	Count	$ \mathcal{K} $	$ \mathcal{N} $	
Plane23	4	DiameterCharacteristic	3	CylinderFeature	3	2	68,344	115,194	
Cylinder23	3	TotalRunOutCharacteristic	1	OppositeParallelPlanesFeature	1				
		WidthCharacteristic	1						
								Surface mesh	
								$ \mathcal{F}_T $	$ \mathcal{N}_T $
								23,130	46,262

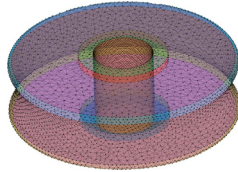
Part 2: Wheel

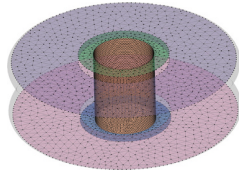
CAD model with PMI

Surface mesh split into QIF faces

Mesh faces with mapped tolerance or datum information

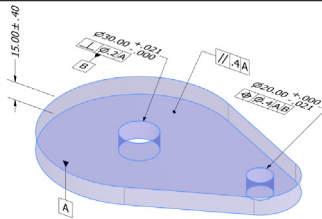
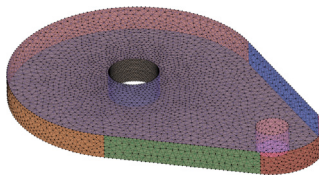
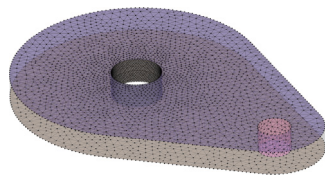


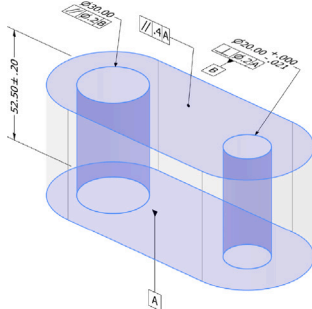
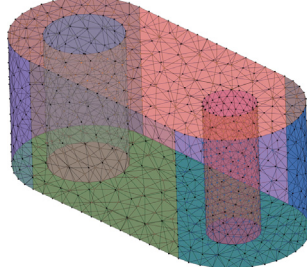
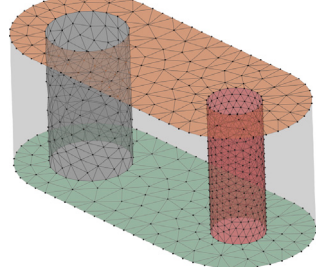




Associated surfaces		Associated characteristics		Associated features		Datums	Volume mesh		
Name	Count	Name	Count	Name	Count	Count	$ \mathcal{K} $	$ \mathcal{N} $	
Plane23	4	SurfaceProfileCharacteristic	2	PlaneFeature	4	1	14,404	25,310	
Cylinder23	3	ParallelismCharacteristic	2	CylinderFeature	1				
Torus23	3	PerpendicularityCharacteristic	1	OppositeParallelPlanesFeature	1				
		DiameterCharacteristic	1						
		WidthCharacteristic	1						
								Surface mesh	
								$ \mathcal{F}_T $	$ \mathcal{N}_T $
								20,648	41,296

Table B.4
Continuation of Table B.3.

Part 3 & 4: Support		Surface mesh split into QIF faces		Mesh faces with mapped tolerance or datum information				
								
Associated surfaces		Associated characteristics		Associated features		Datums	Volume mesh	
Name	Count	Name	Count	Name	Count	Count	$ \mathcal{K} $	$ \mathcal{N} $
Plane23	4	DiameterCharacteristic	2	PlaneFeature	2	2	35,526	61,282
Cylinder23	4	ParallelismCharacteristic	1	CylinderFeature	2			
		PerpendicularityCharacteristic	1	OppositeParallelPlanesFeature	1			
		PositionCharacteristic	1					
		WidthCharacteristic	1					
							Surface mesh	
							$ \mathcal{F}_T $	$ \mathcal{N}'_T $
							13,318	26,634

Part 5: Spacer		Surface mesh split into QIF faces		Mesh faces with mapped tolerance or datum information				
								
Associated surfaces		Associated characteristics		Associated features		Datums	Volume mesh	
Name	Count	Name	Count	Name	Count	Count	$ \mathcal{K} $	$ \mathcal{N} $
Plane23	4	DiameterCharacteristic	2	PlaneFeature	2	2	6,521	11,385
Cylinder23	4	ParallelismCharacteristic	2	CylinderFeature	2			
		PerpendicularityCharacteristic	1	OppositeParallelPlanesFeature	1			
		WidthCharacteristic	1					
							Surface mesh	
							$ \mathcal{F}_T $	$ \mathcal{N}'_T $
							2,566	5,130

- in variation simulation. *Proc Des Soc* 2025;5:2541–50. <http://dx.doi.org/10.1017/pds.2025.10268>.
- [13] Söderberg R, Lindkvist L, Wärmeffjord K, Carlson JS. Virtual geometry assurance process and toolbox. *Procedia CIRP* 2016;43:3–12. <http://dx.doi.org/10.1016/j.procir.2016.02.043>.
- [14] Wärmeffjord K, Hansen J, Söderberg R. Challenges in geometry assurance of megacasting in the automotive industry. *J Comput Inf Sci Eng* 2023;23(6):060801. <http://dx.doi.org/10.1115/1.4062269>.
- [15] Dantan J-Y, Gayton N, Dumas A, Etienne A, Qureshi AJ. Mathematical issues in mechanical tolerance analysis. In: 13e colloque national AIP PRIMECA. Mont-Dore, France; 27.03.–30.03.2012, p. 1–12.
- [16] ASME Y14.5-2009: Dimensioning and Tolerancing Engineering Drawing and Related Documentation Practices; 2009.
- [17] Hong YS, Chang TC. A comprehensive review of tolerancing research. *Int J Prod Res* 2002;40(11):2425–59. <http://dx.doi.org/10.1080/00207540210128242>.
- [18] Thornton AC. A mathematical framework for the key characteristic process. *Res Eng Des* 1999;11(3):145–57. <http://dx.doi.org/10.1007/s001630050011>.
- [19] Anwer N, Schleich B, Mathieu L, Wartzack S. From solid modelling to skin model shapes: Shifting paradigms in computer-aided tolerancing. *CIRP Ann* 2014;63(1):137–40. <http://dx.doi.org/10.1016/j.cirp.2014.03.103>.
- [20] Goetz S, Roth M, Schaechl P, Kramer V, Bode C, Horber D, Franz M, Freitag S, Miehlhng J, Wartzack S. Tolerancing in product design – a holistic description scheme. *Des Sci* 2025;11:e11. <http://dx.doi.org/10.1017/dsj.2025.6>.
- [21] Roth M, Wartzack S. Sampling-based tolerance-cost optimization: Automating least-cost tolerance allocation through joint metaheuristic optimization and sampling-based tolerance analysis. In: Wartzack S, editor. *Research in tolerancing*. Cham: Springer Nature Switzerland; 2024, p. 101–27. http://dx.doi.org/10.1007/978-3-031-64225-8_5.
- [22] Chase KW, Parkinson AR. A survey of research in the application of tolerance analysis to the design of mechanical assemblies. *Res Eng Des* 1991;3(1):23–37. <http://dx.doi.org/10.1007/BF01580066>.
- [23] Lorin S, Lindkvist L, Söderberg R. Simulating part and assembly variation for injection molded parts. In: Volume 5: 6th international conference on micro- and nanosystems; 17th design for manufacturing and the life cycle conference. Chicago, Illinois, USA: American Society of Mechanical Engineers; 2012, p. 487–96. <http://dx.doi.org/10.1115/DETC2012-70659>.
- [24] Stockinger A, Wittmann S, Martinek M, Meerkamm H, Wartzack S. Virtual assembly analysis: Standard tolerance analysis compared to manufacturing simulation and relative positioning. In: DS 60: proceedings of DESIGN 2010, the 11th international design conference. Dubrovnik, Croatia; 17.05–20.05.2010, p. 1421–30.
- [25] Hallmann M, Schleich B, Wartzack S. Sampling-based tolerance analysis: The key to establish tolerance-cost optimization in the product development process. *Procedia CIRP* 2021;100:560–5. <http://dx.doi.org/10.1016/j.procir.2021.05.123>.
- [26] Singh PK, Jain PK, Jain SC. Important issues in tolerance design of mechanical assemblies. Part 1: tolerance analysis. *Proc Inst Mech Eng Part B: J Eng Manuf* 2009;223(10):1225–47. <http://dx.doi.org/10.1243/09544054JEM1304A>.
- [27] Schleich B, Anwer N. Tolerancing informatics: Towards automatic tolerancing information processing in geometrical variations management. *Appl Sci* 2021;11(1):198. <http://dx.doi.org/10.3390/app11010198>.
- [28] Camba J, Contero M, Johnson M, Company P. Extended 3D annotations as a new mechanism to explicitly communicate geometric design intent and increase CAD model reusability. *Computer-Aided Des* 2014;57:61–73. <http://dx.doi.org/10.1016/j.cad.2014.07.001>.
- [29] Lipman R, Lubell J. Conformance checking of PMI representation in CAD model STEP data exchange files. *Computer-Aided Des* 2015;66:14–23. <http://dx.doi.org/10.1016/j.cad.2015.04.002>.
- [30] Hallmann M, Goetz S, Schleich B. Mapping of GD&T information and PMI between 3D product models in the STEP and STL format. *Computer-Aided Des* 2019;115:293–306. <http://dx.doi.org/10.1016/j.cad.2019.06.006>.
- [31] Fang FZ, Li Z, Arokiam A, Gorman T. Closed loop PMI driven dimensional quality lifecycle management approach for smart manufacturing system. *Procedia CIRP* 2016;56:614–9. <http://dx.doi.org/10.1016/j.procir.2016.10.121>.
- [32] Shen Z. Tolerance analysis with EDS/VisVA. *J Comput Inf Sci Eng* 2003;3(1):95–9. <http://dx.doi.org/10.1115/1.1573236>.
- [33] Aderiani AR, Wärmeffjord K, Söderberg R. Model-based definition in computer aided tolerance analyses. *Procedia CIRP* 2022;114:112–6. <http://dx.doi.org/10.1016/j.procir.2022.10.016>.
- [34] Roth M, Rezaei Aderiani A, Morse E, Wärmeffjord K, Söderberg R. Closing gaps in the digital thread with the quality information framework (QIF) standard for a seamless geometry assurance process. *Computer-Aided Des* 2025;182:103860. <http://dx.doi.org/10.1016/j.cad.2025.103860>.
- [35] Kwon S, Monnier LV, Barbau R, Bernstein WZ. Enriching standards-based digital thread by fusing as-designed and as-inspected data using knowledge graphs. *Adv Eng Informatics* 2020;46:101102. <http://dx.doi.org/10.1016/j.aei.2020.101102>.
- [36] S BM, Manu R, Lawrence K D. STEP AP 242 file-based automatic tolerance analysis of mechanical assembly using unified Jacobian Torsor model and direct linearization method. *Int J Comput Integr Manuf* 2023;36(5):756–88. <http://dx.doi.org/10.1080/0951192X.2022.2145017>.
- [37] Praveen OVS, Dileep B, Gayatri S, Deepak Lawrence K, Manu R. Automated tolerance analysis of mechanical assembly using STEP AP 242 managed model-based 3D engineering. In: Chakrabarti A, Arora M, editors. *Industry 4.0 and advanced manufacturing*. Singapore: Springer; 2021, p. 149–57. http://dx.doi.org/10.1007/978-981-15-5689-0_14.
- [38] Venkiteswaran A, Hejazi SM, Biswas D, Shah JJ, Davidson JK. Semantic interoperability of GD&T data through ISO 10303 step AP242. In: Volume 2B: 42nd design automation conference. Charlotte, North Carolina, USA: ASME; 2016. <http://dx.doi.org/10.1115/DETC2016-60133>, V02BT03A018.
- [39] Petruccioli A, Pini F, Leali F. Interoperability framework for a computer-aided approach of tolerance-cost optimization. In: Gerbino S, Lanzotti A, Martorelli M, Mirálses Buil R, Rizzi C, Roucoules L, editors. *Advances on mechanics, design engineering and manufacturing IV*. Cham: Springer International Publishing; 2023, p. 839–50. http://dx.doi.org/10.1007/978-3-031-15928-2_73.
- [40] Roth M. Sampling-based tolerance-cost optimization: the key to optimal tolerance allocation (Ph.D. thesis), Erlangen: Friedrich-Alexander-Universität Erlangen-Nürnberg; 2024. <http://dx.doi.org/10.25593/978-3-96147-720-3>.
- [41] Gopalakrishnan S, Hartman NW, Sangid MD. Model-based feature information network (MFIN): A digital twin framework to integrate location-specific material behavior within component design, manufacturing, and performance analysis. *Integr Mater Manuf Innov* 2020;9(4):394–409. <http://dx.doi.org/10.1007/s40192-020-00190-4>.
- [42] Gopalakrishnan S, Hartman NW, Sangid MD. A digital engineering framework to facilitate automated data exchange between geometric inspection and structural analysis. *Adv Eng Softw* 2023;183:103498. <http://dx.doi.org/10.1016/j.advengsoft.2023.103498>.
- [43] Proctor F, Franaszek M, Michaloski J. Tolerances and uncertainty in robotic systems. In: Volume 2: advanced manufacturing. Tampa, Florida, USA: American Society of Mechanical Engineers; 2017. <http://dx.doi.org/10.1115/IMECE2017-70404>, V002T02A075.
- [44] Walter MSJ, Klein C, Heling B, Wartzack S. Statistical tolerance analysis—A survey on awareness, use and need in german industry. *Appl Sci* 2021;11(6). <http://dx.doi.org/10.3390/app11062622>.
- [45] Goetz S, Schleich B, Wartzack S. CAD-based tolerance analysis in preliminary design stages enabling early tolerance evaluation. In: Volume 2: advanced manufacturing. Pittsburgh, Pennsylvania, USA: American Society of Mechanical Engineers; 2018. <http://dx.doi.org/10.1115/IMECE2018-86396>, V002T02A106.
- [46] Li H, Zhu H, Zhou X, Li P, Yu Z. A new computer-aided tolerance analysis and optimization framework for assembling processes using DP-SDT theory. *Int J Adv Manuf Technol* 2016;86(5–8):1299–310. <http://dx.doi.org/10.1007/s00170-015-8266-9>.
- [47] Wu Y. An automated method for assembly tolerance analysis. *Procedia CIRP* 2020;92:57–62. <http://dx.doi.org/10.1016/j.procir.2020.05.169>.
- [48] Makkonen P, Kousmanen P, Münster R, Mäkeläinen E, Ramseier Y. Quality management by concurrent computer aided tolerance analysis and functional prototyping. In: DS 35: proceedings ICED 05, the 15th international conference on engineering design. Melbourne, Australia; 15.08.–18.08.2005, p. 632–4.
- [49] Mazur M, Leary M, Subic A. A case study of efficient tolerance synthesis in product assemblies under loading. In: DS 81: proceedings of nordDesign 2014. Espoo, Finland; 27.08.–29.08.2014, p. 855–64.
- [50] Großberger A, Reisner M, Prokop G. Innovative kinematic metamodeling: pioneering AI-enhanced suspension tolerance analysis. In: Pfeffer PE, editor. 15th international munich chassis symposium 2024. Berlin, Heidelberg: Springer; 2025, p. 107–19. http://dx.doi.org/10.1007/978-3-662-71070-8_8.
- [51] Elfarrar MA, Önel MN, Yalin G, Malik S. Impact of production tolerances on turbine performance: A CFD based Monte-Carlo simulations study. *Alex Eng J* 2025;117:289–300. <http://dx.doi.org/10.1016/j.aej.2024.12.099>.
- [52] Nguyen DS, Vignat F, Brissaud D. Applying Monte-Carlo methods to geometric deviations simulation within product life cycle. In: Proceedings of the 11th CIRP international conference on computer aided tolerancing. Annecy, France; 26.03.–27.03.2009, p. 1–8.
- [53] Moneta G, Jachimowicz J. Impact of manufacturing tolerances on stress in a turbine blade fir-tree root. *Fatigue Aircr Struct* 2020;2020(12):92–101. <http://dx.doi.org/10.2478/fas-2020-0009>.
- [54] Nerenst TB, Ebro M, Nielsen M, Eifer T, Nielsen KL. Exploring barriers for the use of FEA-based variation simulation in industrial development practice. *Des Sci* 2021;7:e21. <http://dx.doi.org/10.1017/dsj.2021.21>.
- [55] Mahesh G, Saravanan A. A combinatorial optimization approach for geometric tolerance design considering various taxonomies. *Eng Optim* 2024;1–37. <http://dx.doi.org/10.1080/0305215X.2024.2367601>.
- [56] ISO/TC10. ISO 16792:2021 technical product documentation — Digital product definition data practices. 2021, Standard.
- [57] Choi S-K, Grandhi RV, Canfield RA. Reliability-based structural design. London: Springer; 2007.
- [58] Herzog M, Gilg A, Paffrath M, Rentrop P, Wever U. Intrusive versus non-intrusive methods for stochastic finite elements. In: Breiter MH, Denk G, Rentrop P, editors. *From nano to space*. Berlin, Heidelberg: Springer; 2008, p. 161–74. http://dx.doi.org/10.1007/978-3-540-74238-8_13.

- [59] Vlahinos A, Kelkar S. Body-in-white weight reduction via probabilistic modeling of manufacturing variations. In: Proceedings of the 2001 SAE international body engineering conference. Detroit, Michigan, USA; 16.10-18.10.2001, p. 2001-01-3044. <http://dx.doi.org/10.4271/2001-01-3044>.
- [60] Schleich B, Wartzack S. How to determine the influence of geometric deviations on elastic deformations and the structural performance? Proc Inst Mech Eng Part B: J Eng Manuf 2013;227(5):754-64. <http://dx.doi.org/10.1177/0954405412468994>.
- [61] Al-Refaie A, Haddadin M, Qabaja A. Optimal parameters and tolerances in concurrent product and process design using simulation and fuzzy goal programming. Int J Qual Reliab Manag 2019;37(2):372-92. <http://dx.doi.org/10.1108/IJQRM-11-2018-0310>.
- [62] Lee K-K, Ro Y-C, Han S-H. Tolerance optimization of a lower arm by using genetic algorithm and process capability index. Int J Precis Eng Manuf 2014;15(6):1001-7. <http://dx.doi.org/10.1007/s12541-014-0428-4>.
- [63] Schleich B, Stockinger A, Wartzack S. On the impact of geometric deviations on structural performance. Proc the 12th CIRP Conf Comput Aided Toler 2012.
- [64] Franz M, Schleich B, Wartzack S. Tolerance management during the design of composite structures considering variations in design parameters. Int J Adv Manuf Technol 2021;113(5-6):1753-70. <http://dx.doi.org/10.1007/s00170-020-06555-5>.
- [65] Camuz S, Söderberg R, Wärmefjord K, Lundblad M. Tolerance analysis of surface-to-surface contacts using finite element analysis. Procedia CIRP 2018;75:250-5. <http://dx.doi.org/10.1016/j.procir.2018.04.029>.
- [66] Falgarone H, Thiébaud F, Coloos J, Mathieu L. Variation simulation during assembly of non-rigid components. realistic assembly simulation with ANA-TOLFLEX Software. Procedia CIRP 2016;43:202-7. <http://dx.doi.org/10.1016/j.procir.2016.02.336>.
- [67] Zheng H, Upadhyay K, Litwa F, Paetzold K. A meta-model based approach to implement variation simulation for sheet metal parts using mesh morphing method. In: 13th European LS-DYNA conference 2021. Ulm, Germany; 2021.
- [68] Jayaprakash G, Thilak M, SivaKumar K. Optimal tolerance design for mechanical assembly considering thermal impact. Int J Adv Manuf Technol 2014;73(5-8):859-73. <http://dx.doi.org/10.1007/s00170-014-5845-0>.
- [69] Mazur M, Leary M, Subic A. Application of polynomial chaos expansion to tolerance analysis and synthesis in compliant assemblies subject to loading. J Mech Des 2015;137(3):031701. <http://dx.doi.org/10.1115/1.4029283>.
- [70] Corrado A, Polini W. FEA integration in the tolerance analysis using skin model shapes. Procedia CIRP 2018;75:285-90. <http://dx.doi.org/10.1016/j.procir.2018.04.055>.
- [71] Shen Z, Ameta G, Shah JJ, Davidson JK. A comparative study of tolerance analysis methods. J Comput Inf Sci Eng 2005;5(3):247-56. <http://dx.doi.org/10.1115/1.1979509>.
- [72] Schleich B, Anwer N, Mathieu L, Wartzack S. Skin model shapes: A new paradigm shift for geometric variations modelling in mechanical engineering. Computer-Aided Des 2014;50:1-15. <http://dx.doi.org/10.1016/j.cad.2014.01.001>.
- [73] Yan X, Ballu A. Generation of consistent skin model shape based on FEA method. Int J Adv Manuf Technol 2017;92(1-4):789-802. <http://dx.doi.org/10.1007/s00170-017-0177-5>.
- [74] Heysiatlab S, Morse EP. From STEP to QIF: Product and manufacturing information. In: 31st ASPE annual meeting. 65, Portland, OR, USA: American Society for Precision Engineering; 2016, p. 312-7. <http://dx.doi.org/10.13140/RG.2.2.34292.55682/1>.
- [75] ISO/TC184/SC4. ISO 23952:2020 automation systems and integration — quality information framework (QIF) — an integrated model for manufacturing quality information. (23952). 2020, Standard.
- [76] Autodesk Inc. ANSYS Discovery SpaceClaim Online Documentation, URL https://help.spaceclaim.com/v19.0/en/Content/Importing_and_exporting.htm, [Accessed: 30.07.2025].
- [77] Autodesk Inc. Autodesk Inventor Professional 2024. Saving and exporting files reference, URL <https://help.autodesk.com/view/INVNTOR/2024/ENU/?guid=GUID-08D6DD13-3066-4210-9F19-5F7E8DB8AD5D>, [Accessed: 30.07.2025].
- [78] Capvidia. Capvidia MBDVidia. Easy & Powerful MBD Workflow CAD Translation Software, URL <https://www.capvidia.com/products/mbdvidya>, [Accessed: 30.07.2025].
- [79] Elysium. Elysium. QIF Translation, URL <https://www.elysium.com/qif-translation>, [Accessed: 30.07.2025].
- [80] Campbell D, Brown C, Brown R, Herron J, Admire R, Horst J, Leland C, Stahl R. Why QIF matters – a roadmap for digital manufacturing. In: Proceedings of the 10th model-based enterprise summit (MBE 2019). Gaithersburg, Maryland, USA; 02.04-04.04.2019, p. 58-63. <http://dx.doi.org/10.6028/NIST.AMS.100-24#page=66>.
- [81] Brenner SC, Scott LR. The mathematical theory of finite element methods. New York: Springer; 2008. <http://dx.doi.org/10.1007/978-0-387-75934-0>.
- [82] Zienkiewicz OC, Taylor RL, Zhu J. The finite element method: its basis and fundamentals, Sixth ed.. Amsterdam: Elsevier; 2010.
- [83] Szczygiel I, Szłęk A, Bialecki R, Nowak AJ. Identification of the boundary surfaces in 3D finite element codes. Adv Eng Softw 1992;14(1):33-9. [http://dx.doi.org/10.1016/0965-9978\(92\)90081-P](http://dx.doi.org/10.1016/0965-9978(92)90081-P).
- [84] Krishnamurthy A, Khardekar R, McMains S, Haller K, Elber G. Performing efficient NURBS modeling operations on the GPU. IEEE Trans Vis Comput Graphics 2009;15(4):530-43. <http://dx.doi.org/10.1109/TVCG.2009.29>.
- [85] Chen X-D, Yong J-H, Wang G, Paul J-C, Xu G. Computing the minimum distance between a point and a NURBS curve. Computer-Aided Des 2008;40(10-11):1051-4. <http://dx.doi.org/10.1016/j.cad.2008.06.008>.
- [86] Mortenson ME. Geometric modeling. New York: Wiley; 1985.
- [87] Piegl L, Tiller W. The NURBS book. Monographs in visual communications, Berlin, Heidelberg: Springer Berlin Heidelberg; 1995. <http://dx.doi.org/10.1007/978-3-642-97385-7>.
- [88] Hormann K, Agathos A. The point in polygon problem for arbitrary polygons. Comput Geom 2001;20(3):131-44. [http://dx.doi.org/10.1016/S0925-7721\(01\)00012-8](http://dx.doi.org/10.1016/S0925-7721(01)00012-8).
- [89] Yi Y, Liu T, Yan Y, Feng J, Liu J, Ni Z, Liu X. A novel assembly tolerance analysis method considering form errors and partial parallel connections. Int J Adv Manuf Technol 2024;131(11):5489-510. <http://dx.doi.org/10.1007/s00170-022-09628-9>.
- [90] Bourdet P, Mathieu L, Lartigue C, Ballu A. The concept of small displacement torsor in metrology. Ser Adv Math Appl Sci Advanced Math Tool Metrol II 1996;110-22.
- [91] Ghie W, Laperrière L, Desrochers A. A unified Jacobian-torsor model for analysis in computer aided tolerancing. In: Gogu G, Couellier D, Chedmail P, Ray P, editors. Recent advances in integrated design and manufacturing in mechanical engineering. Dordrecht: Springer; 2003, p. 63-72. http://dx.doi.org/10.1007/978-94-017-0161-7_7.
- [92] Frère L-M, Royer M, Fourcade J. Tolerance analysis using a computer aided tolerancing software: ANATOLE 3D. Procedia CIRP 2018;75:267-72. <http://dx.doi.org/10.1016/j.procir.2018.04.079>.
- [93] Clozel P, Rance P-A. MECAMaster: A tool for assembly simulation from early design, industrial approach. In: Villeneuve Fc, Mathieu L, editors. Geometric tolerancing of products. Wiley; 2013, p. 241-73. <http://dx.doi.org/10.1002/9781118587027.ch10>.
- [94] Moro T, Askri R, Rance P-A, Clozel P, Martin J, Veen SVD, Canderatz F, Judic J-M. New flexible tolerancing strategy and development for complex structural aeronautical assemblies analysis in MECAMaster® software environment. Procedia CIRP 2022;114:171-6. <http://dx.doi.org/10.1016/j.procir.2022.10.023>.
- [95] Li H, Zhu H, Li P, He F. Tolerance analysis of mechanical assemblies based on small displacement torsor and deviation propagation theories. Int J Adv Manuf Technol 2014;72(1-4):89-99. <http://dx.doi.org/10.1007/s00170-014-5630-0>.
- [96] Peng H, Chang S. Including material conditions effects in statistical geometrical tolerance analysis of mechanical assemblies. Int J Adv Manuf Technol 2022;119(9-10):6665-78. <http://dx.doi.org/10.1007/s00170-021-08247-0>.
- [97] Morse EP, Sriniivasan V. Size tolerancing revisited: A basic notion and its evolution in standards. Proc Inst Mech Eng Part B: J Eng Manuf 2013;227(5):662-71. <http://dx.doi.org/10.1177/0954405412470418>.
- [98] Schleich B, Wartzack S. Evaluation of geometric tolerances and generation of variational part representatives for tolerance analysis. Int J Adv Manuf Technol 2015;79(5-8):959-83. <http://dx.doi.org/10.1007/s00170-015-6886-8>.
- [99] Liu J, Zhang Z, Ding X, Shao N. Integrating form errors and local surface deformations into tolerance analysis based on skin model shapes and a boundary element method. Computer-Aided Des 2018;104:45-59. <http://dx.doi.org/10.1016/j.cad.2018.05.005>.
- [100] Ohtake Y, Belyaev A, Bogaevski I. Mesh regularization and adaptive smoothing. Computer-Aided Des 2001;33(11):789-800. [http://dx.doi.org/10.1016/S0010-4485\(01\)00095-1](http://dx.doi.org/10.1016/S0010-4485(01)00095-1).
- [101] Selim MM, Koomullil RP. Mesh deformation approaches – a survey. J Phys Math 2016;7(2). <http://dx.doi.org/10.4172/2090-0902.1000181>.
- [102] Guan B, Lin S, Wang R, Zhou F, Luo X, Zheng Y. Voxel-based quadrilateral mesh generation from point cloud. Multimedia Tools Appl 2020;79(29-30):20561-78. <http://dx.doi.org/10.1007/s11042-020-08923-5>.
- [103] Claus F, Hamann B, Hagen H. A finite-element based mesh morphing approach for surface meshes. Computer-Aided Des 2022;146:103232. <http://dx.doi.org/10.1016/j.cad.2022.103232>.
- [104] Parthasarathy V, Kodiyalam S. A constrained optimization approach to finite element mesh smoothing. Finite Elem Anal Des 1991;9(4):309-20. [http://dx.doi.org/10.1016/0168-874X\(91\)90004-I](http://dx.doi.org/10.1016/0168-874X(91)90004-I).
- [105] Gouraud H. Continuous shading of curved surfaces. IEEE Trans Comput 1971;C-20(6):623-9. <http://dx.doi.org/10.1109/T-C.1971.223313>.
- [106] Cenancovic M, Hansbo P, Larson MG. Finite element procedures for computing normals and mean curvature on triangulated surfaces and their use for mesh refinement. Comput Methods Appl Mech Engrg 2020;372:113445. <http://dx.doi.org/10.1016/j.cma.2020.113445>.
- [107] Schleich B, Wartzack S. Novel approaches for the assembly simulation of rigid skin model shapes in tolerance analysis. Computer-Aided Des 2018;101:1-11. <http://dx.doi.org/10.1016/j.cad.2018.04.002>.
- [108] Schleich B, Wartzack S. Approaches for the assembly simulation of skin model shapes. Computer-Aided Des 2015;65:18-33. <http://dx.doi.org/10.1016/j.cad.2015.03.004>.

- [109] Goka E, Homri L, Beurepaire P, Dantan J-Y. Statistical tolerance analysis of over-constrained mechanical assemblies with form defects considering contact types. *J Comput Inf Sci Eng* 2019;19(2):021010. <http://dx.doi.org/10.1115/1.4042018>.
- [110] Ma S, Hu T, Xiong Z. Precision assembly simulation of skin model shapes accounting for contact deformation and geometric deviations for statistical tolerance analysis method. *Int J Precis Eng Manuf* 2021;22(6):975–89. <http://dx.doi.org/10.1007/s12541-021-00505-1>.
- [111] Yan X, Ballu A. Assembly simulation of skin model shapes: A comparison of two methods taking into account external forces. *Procedia CIRP* 2018;75:303–8. <http://dx.doi.org/10.1016/j.procir.2018.04.072>.
- [112] Horn BKP. Closed-form solution of absolute orientation using unit quaternions. *J Opt Soc Amer A* 1987;4(4):629. <http://dx.doi.org/10.1364/JOSAA.4.000629>.
- [113] Besl P, McKay ND. A method for registration of 3-D shapes. *IEEE Trans Pattern Anal Mach Intell* 1992;14(2):239–56. <http://dx.doi.org/10.1109/34.121791>.
- [114] Gouyou D, Ledoux Y, Teissandier D, Delos V. Tolerance analysis of overconstrained and flexible assemblies by polytopes and finite element computations: Application to a flange. *Res Eng Des* 2018;29(1):55–66. <http://dx.doi.org/10.1007/s00163-017-0256-5>.
- [115] Digital Metrology Standards Consortium. QIF schema and bindings. 2025.
- [116] Laughlin T. pyOCCT – Python Bindings for OpenCASCADE via Pybind11, URL <https://github.com/trelau/pyOCCT>, [Accessed: 09.05.2025].
- [117] Gillies S, van der Wel C, Van den Bossche J, Taves MW, Arnott J, Ward BC, others. Shapely. 2022, <http://dx.doi.org/10.5281/ZENODO.7428463>.
- [118] Bingol OR, Krishnamurthy A. NURBS-python: An open-source object-oriented NURBS modeling framework in Python. *SoftwareX* 2019;9:85–94. <http://dx.doi.org/10.1016/j.softx.2018.12.005>.
- [119] Ziegler P, Wartzack S. Sensitivity analysis of features in tolerancing based on constraint function level sets. *Reliab Eng Syst Saf* 2015;134:324–33. <http://dx.doi.org/10.1016/j.res.2014.09.017>.
- [120] ANSYS Inc. PyMAPDL. Documentation, URL <https://mapdl.docs.pyansys.com/>, [Accessed: 30.07.2025].
- [121] Sorkine O, Alexa M. As-rigid-as-possible surface modeling. In: *Symposium on geometry processing*. 4, 2007, p. 109–16.
- [122] Segal AV, Haehnel D, Thrun S. Generalized-ICP. In: Trinkle J, Matsuoka Y, Castellanos JA, editors. *Robotics: science and systems v*. The MIT Press; 2010, p. 1–8. <http://dx.doi.org/10.7551/mitpress/8727.003.0022>.
- [123] Maneewongvatana S, Mount DM. Analysis of approximate nearest neighbor searching with clustered point sets. 1999, <http://dx.doi.org/10.48550/arXiv.cs/9901013>.
- [124] Zhou Q-Y, Park J, Koltun V. Open3D: a modern library for 3D data processing. 2018, <http://dx.doi.org/10.48550/arXiv.1801.09847>.
- [125] Jeang A, Chang C-L. Combined robust parameter and tolerance design using orthogonal arrays. *Int J Adv Manuf Technol* 2002;19(6):442–7. <http://dx.doi.org/10.1007/s001700200046>.
- [126] ANSYSInc. Mechanical APDL 2024 R2. Theory Reference. Chapter 2.4. Combined Stresses and Strains, URL https://ansyshelp.ansys.com/public/account/secured?returnurl=/Views/Secured/corp/v242/en/ans_thry/thy_str4.html?q=SEQV, [Accessed: 30.07.2025].
- [127] Das A, Franciosa P, Williams D, Ceglarek D. Physics-driven shape variation modelling at early design stage. *Procedia CIRP* 2016;41:1072–7. <http://dx.doi.org/10.1016/j.procir.2016.01.031>.
- [128] Lindau B, Lindkvist L, Andersson A, Söderberg R. Statistical shape modeling in virtual assembly using PCA-technique. *J Manuf Syst* 2013;32(3):456–63. <http://dx.doi.org/10.1016/j.jmsy.2013.02.002>.
- [129] Franciosa P, Gerbino S, Patalano S. Simulation of variational compliant assemblies with shape errors based on morphing mesh approach. *Int J Adv Manuf Technol* 2011;53(1–4):47–61. <http://dx.doi.org/10.1007/s00170-010-2839-4>.
- [130] Anwer N, Scott PJ, Srinivasan V. Toward a classification of partitioning operations for standardization of geometrical product specifications and verification. *Procedia CIRP* 2018;75:325–30. <http://dx.doi.org/10.1016/j.procir.2018.02.018>.