



# Concepts of data collection for the CAD-integrated isogeometric analysis

Tobias Teschemacher<sup>1,2</sup> · Anna M. Bauer<sup>1</sup> · Ricky Aristio<sup>1,3</sup> · Manuel Meßmer<sup>1</sup> · Roland Wüchner<sup>3</sup> · Kai-Uwe Bletzinger<sup>1</sup>

Received: 24 April 2022 / Accepted: 8 August 2022 / Published online: 10 October 2022  
© The Author(s) 2022

## Abstract

This publication presents required steps for the realization of the pre- and post-processing for the isogeometric analysis and the isogeometric B-Rep analysis, with a focus on the collection of required data. It reveals the essential prerequisites for the preparation and the collection of geometrical information, which are merged with physical information for the creation of numerical models. It addresses both the direct computation on existing CAD drawings and the geometrical design during the preparation of the numerical models. The developments are presented through the example of the open source Rhino plugin Cocodrilo, which shall bring IGA to a larger community, including research and industrial facilities.

**Keywords** IGA · IBRA · CAD-integrated analysis · Pre- and post-processing · NURBS-based analysis

## 1 Introduction

Ever since the introduction and the subsequent further developments of finite element methods (FEM), computer aided design (CAD)-integrated simulation became a major avenue in the field, as it does provide eased modelling and visualization facilities for complex problems [3, 9, 22]. Initially, CAD has been seen as the tool to digitize the drawing

board. However, it is nowadays employed to fulfill much more tasks, as for example providing a pre- and post-processor for FEM. For classical computer aided engineering (CAE) workflows, the drafted geometries would be meshed or extracted into geometrical primitives, such as triangles or quadrilaterals, which are common base forms within FEM. Generating a FE-suitable, meshed geometry can be very complex and error prone, whereby the developments have been enormous to automatize this process. However, meshing is still time consuming and it does recreate shapes, which corrupts the original bodies and complicates iterative operations. In opposition to generating FE models by converting the geometry description from CAD, Hughes et al. [20, 36] have introduced isogeometric analysis (IGA). It is a FEM approach directly relying on Non-Uniform Rational B-Splines (NURBS), which is a standard representation within CAD models.

The initial concept of IGA was the use of full, untrimmed NURBS patches, like curves, surfaces and solids. NURBS maintain a high continuity, which allows a very efficient simulation with typically a moderate number of degrees of freedom. However, shaping complex geometries with single entities appears to be demanding, resulting repeatedly in badly conditioned FE models. One avenue to circumvent this restriction has been the introduction of T-Splines [70, 71, 82], hierarchic splines [83], and other spline technologies within the scope of

✉ Tobias Teschemacher  
tobias.teschemacher@tum.de

Anna M. Bauer  
am.bauer@tum.de

Ricky Aristio  
ricky.aristio@tum.de

Manuel Meßmer  
manuel.messmer@tum.de

Roland Wüchner  
r.wuechner@tu-braunschweig.de

Kai-Uwe Bletzinger  
kub@tum.de

- <sup>1</sup> Chair of Structural Analysis, Technical University of Munich, Munich, Germany
- <sup>2</sup> School of Engineering, The University of Queensland, Brisbane, Australia
- <sup>3</sup> Institute of Structural Analysis, TU Braunschweig, Brunswick, Germany

IGA, which allowed modeling more sophisticated shapes. However, CAD systems do typically rely on an alternative approach in the shape description, the so-called boundary representation (B-Rep) [73].

B-Rep geometrically defines objects by their circumscription, meaning a solid is being described by a set of enclosing surfaces, which themselves have trimming curves, and those have start and end points. With this methodology, many possibilities arise for the description of objects. This approach eliminates the restriction of NURBS being topological rectangles (or cuboids). New shapes can be generated by adding different boundary curves to the surfaces, i.e. trimming, and joining them to more complex objects. To allow an immediate simulation upon those descriptions, IGA needs to be enhanced by facilitating trimming and the consideration of multiple patches and their interaction, which results in the isogeometric B-Rep analysis (IBRA) [11, 12]. In contrast to the application of certain spline technologies, this approach permits the direct usage of common CAD models without any re-discretization and allows seamless back and forth integration into CAD.

Cohen et al. [16] proposed a guideline for an analysis aware design with IGA. It provides comprehensive suggestions about the quality of NURBS-based models, whereby it relies on untrimmed forms. Therefore, an extension study towards the applicability of IGA with trimming and multi-patches of more generic CAD models is required.

Since the introduction of IGA and IBRA, many simulation tools have been developed, which contain some of the apparent methodologies, as well as complete workflows as e.g. gismo [29], LS-DYNA [53], CoreForm Software/IGA [19], Carat++ [14] and Kratos Multiphysics [21, 47]. Those tools mainly address the solver-side of the technique. However, a vast part of the model preparation, as enhancement of data and geometrical cleaning, happens on the CAD side to enable a numerical simulation. This is a vital part in industrial applications, however, academically not covered comprehensively. Perduta and Putanowicz [62] and Hsu et al. [35] present some innovative approaches and necessities for the pre- and post-processing and integration of IGA in CAD, specifically with a focus on parametric design and [72] presents some avenues for a seamless post-processing. CoreForm IGA [19], Ansa [13], LS-PrePost [54], and Kiwi!3d [44] are currently popular pre- and post-processing tools for IGA. An open-source data consistent CAD-based pre- and post-processing is to the best of our knowledge not available.

A concept for analysis-enhanced geometrical information is presented within this publication. Additionally, a classification of CAD-models is being discussed, which shows the possibilities and limitations of the simulation within current CAD systems. Alongside this methodological discussion, the Rhino plugin Cocodrilo [75] is presented, which has multiple user interaction possibilities, including a GUI and

graphical programming interface for parametric design with IGA-based simulations.

From a formal perspective, this paper is structured as following:

*Data enhanced CAD-systems* provides a possible avenue for the merge between drafts and physical entities within CAD.

*CAD based geometry description* reveals solvable issues and current limitations within CAD-integrated analysis.

*Interaction with the CAD-data* presents user interfaces for CAD models to enhance them with the required physics and user information.

*Software architecture* features a suitable software architecture to achieve a full CAD-integrated design considering all peculiarities discussed within the previous sections.

*Application Examples* shows some possible applications of the presented engineering software.

## 2 Data enhanced CAD-systems

Commonly, two directions are imaginable for merging geometrical shapes and physical objects. One is by structuring physical objects as entities of a selected type and defining specific geometrical expansions and scaling, with which the shape is extrapolated upon a single or multiple geometrical bodies. Second, the vice versa direction is possible, whereby a set of geometries is grouped and defined as part of a certain object.

The first approach has its benefits once the focus is set on highly repetitive objects within a model where the specific forms do not highly vary. This is the case for classical civil engineering purposes, for e.g. buildings with a high number of similar doors or windows. This manner of defining the model would allow multiple ways of deriving alternative geometrical representations, whereby design and engineering needs may be fulfilled simultaneously. Accordingly, many known Building Information Modeling (BIM) models typically rely on this type of data acquisition. However, the restrictions of those types of data collectors are that they may become complex once sophisticated shapes are involved or objects cannot be individually identified.

As this publication is focused on the detailed assessment of IGA, where a strict communication and control of the shape is more important than a fast data treatment towards non-structural results, it is preferred to rely on the latter mentioned approach of enhancing geometrical entities with physical properties.

Accordingly, various possibilities exist for enhancing the geometrical shapes with the essential physical information for accurate numerical simulations considering a

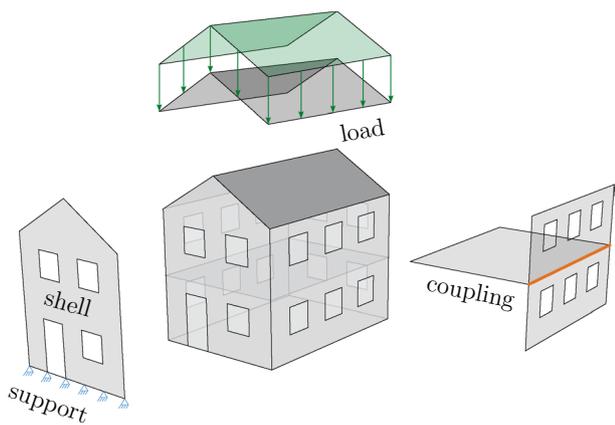


Fig. 1 Analysis enhanced CAD model

model that follows the suggestions from Sect. 3. Initially, user-defined properties may be assigned to various topology items of the draft, which can be virtually connected. The respective data types involve kinematic formulations for all geometries and additionally include a set of load and support conditions. In contrast, some entities are generally not directly accessible for the user interface, which are, for e.g., geometry couplings. This crucial group is mostly controlled indirectly via connected user-provided information, which in practice turns out as a complex operation.

Figure 1 shows an example of a generic building that is enhanced by various entities, which shall display the possibilities and the facilities that have to be enabled by the proposed pre-processing approaches.

Enhancing the model with additional information results also in the need for novel interfaces to correctly communicate with the respective numerical solver. A possible avenue, including required data and feasible formatting, is presented within [74]. Apart from the pure geometrical description, a major focus is set on the correct identification of the connections between the individual geometries, physics, and materials. This makes it challenging to directly apply standardized CAD formats.

For some types of simulations, it may not be required to acquire information other than the geometric entities identified by e.g. a type or group. This is e.g. the case for an embedded simulation using the CAD model for boundary conditions (for example IGA in interaction with the discrete element method (DEM) [76], or embedded IGA within computational fluid dynamics), where all provided geometries of a specific type are considered. In this scenario, the solver could be also addressed with common standards as step [38] or iges [77]. Accordingly, the pre-processing would be reduced to generic solver settings which do not require a CAD-based data acquisition.

### 3 CAD based geometry description

Figure 2 presents the CAD model and the displacement results of an IGA simulation of a airplane structure. The model is taken from [78] and has initially been designed for CFD purposes and as such, it is frequently used for benchmarking. The model appears to be suitable for structural analyses with IGA, however, requires special attention on

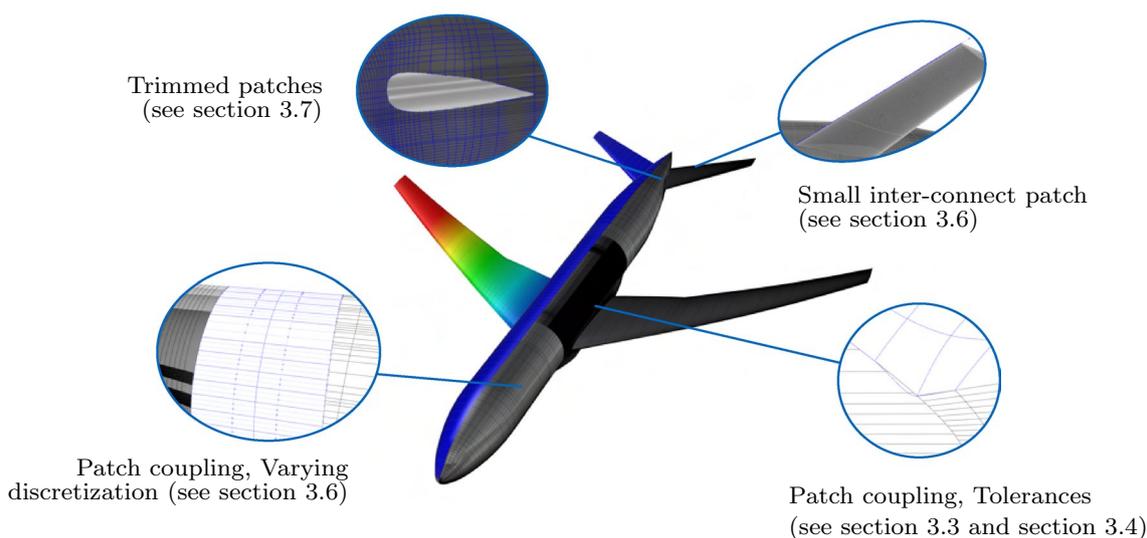


Fig. 2 Airplane common research model geometry [78] with apparent description issues

some details for reliable analyses results. Among others, those cover: trimming, patch coupling of 28 patches with varying discretization, small inter-connection patches, and significant tolerances. If treated inadequately, those issues may result in unfeasible solutions.

Although the CAD-integrated analysis with IGA is based on the exact geometrical shapes provided from the CAD software, the description might not be well-suited for simulations. One of the key challenges in using CAD-provided geometries is that the design and visualization typically need a different, usually lower, level of refinement in the shape description since the model has to represent only the static geometry. Whereas the numerical model and its mathematical description have to represent various sets of information, such as the undeformed and deformed shape and stresses. Accordingly, some issues in the CAD-based context with respect to direct use for structural analysis shall be discussed within this section.

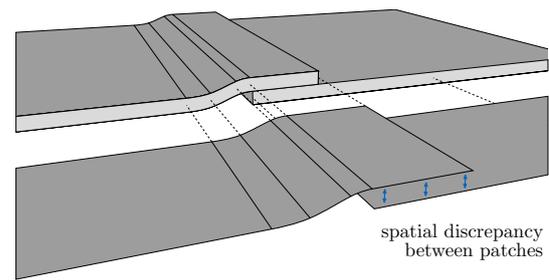
It shall be noted, some of the critical challenges resulting from poor geometry descriptions are intrinsically present in the model and would manifest themselves equivalently in an automated meshing process prior to classical FEM-based simulation [74].

### 3.1 Solid model

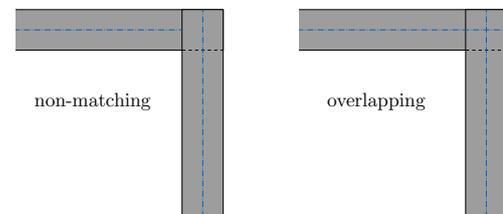
B-Rep-based CAD models are constructed with 1D curves or 2D surfaces, which is especially suitable for fast and stable graphical design purposes. Accordingly, the simulations may rely on 1D or 2D shapes, which can be the description for dimension-reduced formulations such as beams, e.g. [6, 55, 81] and shells, e.g. [8, 24, 43, 57, 58]. Those come along with reduced computational efforts, however, are limited in the representation of very complex structural behaviors.

If 3D solid-based computation is inevitable, increased efforts are required as CAD does not intrinsically provide a mathematical volumetric solid description. One possible avenue is the embedding of the enclosing surfaces into a linearized mesh, or a 3-dimensional NURBS-cube background. This technique is frequently applied within the Finite Cell Method (FCM) [23, 60] and allows for handling various types of geometries such as trimmed solid models [65], CSG models [79], B-Rep models which may be flawed [80], or point cloud based models [48].

Another possible method for the computation of solids within boundary representation has been introduced and described by [15, 45, 46]. They propose to construct a NURBS-based solid from the radial scaling center of the physical domain towards the NURBS description of the faces. Both approaches seem to be promising, however, it shall be noted that they are in controversy of the initial IGA concept of using the CAD-initiated shape description and thus limits multi-stage analyses.



(a) Welded steel plates with mid surfaces.



(b) Wall intersection.

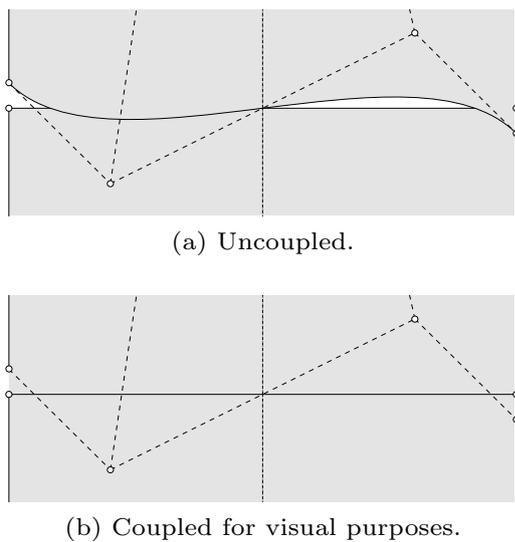
Fig. 3 Model reduction using mid surfaces

Further references present different possibilities for an automated solid simulation from a CAD-model, such as the construction of solid T-Splines [52, 85].

### 3.2 Surface model vs. mid surface model

Thin-walled structures, which are suitable for the computation of dimensionally reduced FE-elements, are often expressed as B-Rep solids in industrial applications since the enclosing surface is decisive in the design. This obstructs a direct evaluation with common surface-based formulations since many of them rely on the mid surface of a thin-walled structure. Figure 3a shows two steel plates that are welded together. The model is displayed as boundary surface model and its reduction to the mid surface. It shows that the solid surfaces appear to have matching faces of adjacent components, whereas the mid surface model has severe gaps. An automated detection of coupling lines is highly dependent on the user-defined threshold. This tolerance can be chosen for certain types of structures. However, it is not possible to automate this process if there are gaps within the same chosen tolerance that should remain open or if other objects are close but should remain uncoupled. Another example, which appears frequently in civil engineering, is the corner case of two intersecting walls as Fig. 3b shows.<sup>1</sup> If the volumes do

<sup>1</sup> Horger et al. [34] present a Mortar-based approach to model such corners and [59] is presenting the numerical simulation of a timber connection, where the interface itself is enhanced by a material law to represent an elastic material layer.



**Fig. 4** Coupling of two patches with large model tolerance

not overlap, which would be the correct volumetric modelling, the mid surfaces do not intersect and coupling cannot be detected in an automated way beside a certain tolerance. If they do overlap, the mid surfaces intersect and the coupling is detected but the correct modelling of joined curves in the corner point is still not gained.

To summarize, the extraction towards mid surfaces needs additional understanding of the specific geometrical issue and accordingly manual work load or very sophisticated, problem dependent algorithms, especially at intersections. Note that this challenge is similar for classic mesh-based FEM approaches.

### 3.3 Model tolerance

Dirty CAD descriptions with high model tolerances can lead to gaps between the provided patches, which might be filled visually by the software using an internal triangulated visualization mesh. Consequently, it becomes difficult to visually detect unsuitable geometry (see Fig. 4). The model quality may be such poor that the CAD utility cannot set up the coupling conditions within appropriate tolerances. Additionally, with unbridgeable gaps between patches some coupling methods fail or converge towards wrong solutions and a proper physical description is not ensured (see also [5, 11]).

### 3.4 Inter-patch connections for geometrical continuities

Some CAD programs intrinsically provide operations to join geometry items to fewer objects if there is a geometrical connection. Intersections are detected within a given

accuracy, and model tolerance and additional interface shapes are appended. Those operations are typically carried out in a way helping to seamlessly and efficiently visualize the drafts, however, for engineering purposes, it is often not suitable or not sufficient.

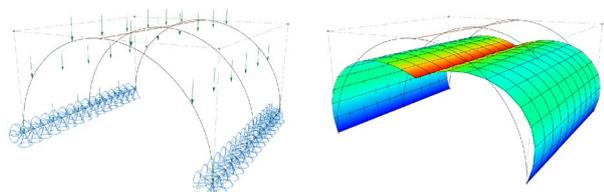
Considering a broad model tolerance, the join operation would find many coupling edges or points, which indeed are decently distanced and it would visualize them as a gap-free object. An example of the coupling edge between two visually non-matching patches is shown in Fig. 4. Figure 4a is displaying the original shape of the two patches. Within Fig. 4b, the same configuration after a join command is depicted. It shows that the visible discrepancy between the two patches is disappearing, even though the discretization does not match. The analysis, however, would not be suitable to perform a stable simulation or maybe does not even detect the ordinary integration. The gap might also be eccentric for unproper modelling, which introduces not existing stresses into the structure. Furthermore, the gap is assigned with a certain behavior when a coupling condition is added. This typically involves different behavior in the area of the gap than inside the material domain, which might affect the results depending on the gap size. Choosing a suitable model tolerance (see Sect. 3.3) avoids this or decreases the error to a tolerable level. Considering this while modeling or repairing the geometries in a way that the wanted connections do not consider overly large gaps would lead to better solutions.

Furthermore, CAD-integrated join operations do typically just consider coupling among the boundaries of the shapes and each edge usually has only one other counterpart since the software structure is adjusted to handle boundary models. It may happen that e.g. T-joints (as shown in Fig. 1) remain undetected and get discarded. This concerns joints at the boundary involving three surfaces or one surface intersecting with another one inside its domain. Undetected couplings, especially those manifold joints, are usually not a big problem in CAD software if the geometry is constructed with sufficient accuracy since the geometries are still close and visually connected. However, the necessary information for the engineering part is not retrieved. Consequently, it is suggested, even though often computationally very expensive, to run additional intersection checks.

Bauer et al. [7] present a range of possible applications for 1D structures, whereby intersection checks between the presented beams are processed to obtain all required couplings. Some shell-based problems with T-joints are presented by Hirschler et al. [32, 33] or Herrema et al. [31].

### 3.5 Geometrical discontinuities

Shapes can be modeled with a defined geometric continuity. However, this does not imply a respective continuity in



(a) Problem description with discontinuity at top. (b) Deformed problem.

**Fig. 5**  $C^0$  and  $G^1$  continuous geometry with possibly undetected  $G^1$  discontinuity in middle of the patch, due to duplicated knots. The knot vector of this example for  $p = 2$  may read:  $\Xi = [0, 0, 0, \frac{\pi}{2}, \frac{\pi}{2}, \pi, \pi, \pi]$ , whereby the inner knot (orange) is duplicated (for further notes about NURBS see [64])

the mathematical description.<sup>2</sup> One may obtain unexpected results, once the geometrical  $G^1$  continuity (rotation) is used to express the bending stiffness, as e.g. within the Kirchhoff–Love-shell [43] or the Bernoulli-beam [6].

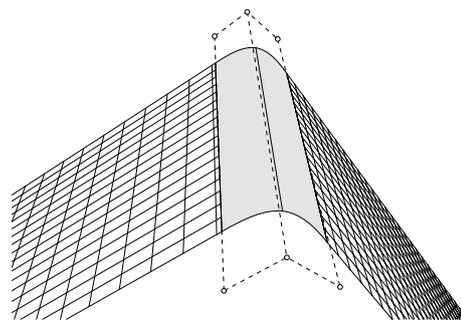
Among various scenarios, this happens frequently in the description of geometrically exact circular shapes, as shown in Fig. 5a. The middle knot is duplicated, which results in a  $C^0$  continuity in geometry space ensuring only that no gap evolves at the specific position. The same rotation on the left and right side of this point, i.e.  $G^1$  continuity, cannot be automatically guaranteed. This discontinuity will still exist after any type of refinement. The outcome of a simulation with this specific problem is given within Fig. 5b. Note that the mid line along the duplicated knot shows a significant discontinuity in rotation, which should not exist if there is no specifically modelled hinge in the shell.

Various different geometry descriptions may be found, which would describe the same shape, without the discontinuity. However, this would corrupt the initial description. Accordingly, the discontinuities may be enforced by the bending strip approach [42] or by weakly applying rotational stiffness in order to not change any geometries, similar to patch-couplings [12, 31, 51].

### 3.6 Patch conditioning

Many CAD programs provide operations to visually eliminate gaps between existent patches. Those operations are optimized for visual purposes. However, the additional patches are repeatedly showing difficult mathematical properties for structural analysis purposes. Those patches have often substantial differences in the geometry extensions of the knot spans causing badly conditioned numerical elements. Efficient automated refinement operations and robust integration techniques are recommended to prevent

<sup>2</sup> Discontinuities are frequently introduced in NURBS-based domains by duplicating knots.



**Fig. 6** Badly conditioned, connected patch

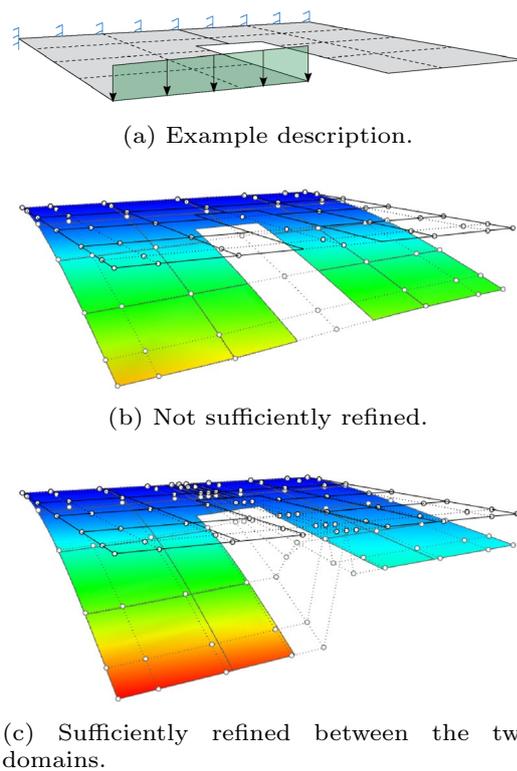
this issue. The refinement derives from the respective span length. However, in some of those patches, boundary effects, introduced by weak coupling approaches might predominate the kinematic properties of the surface formulations, which may lead to an artificial stiffness.<sup>3</sup> A further critical issue is that even if solving appears to converge, the solution might still be unreliable. The assessment of the result quality may become difficult, which may conclude in a loss of trust of the numerical method.

This happens for example if a rounded connection patch between two patches is created by an automated process. Figure 6 is presenting such a problematic patch. The patch has been constructed during a smoothing corner operation. It contains only six control points. Thus, by connecting it to the neighboring patches it would create a numerical stiffness. Sometimes it is hard to detect those patches in larger models, as they are very slim and tiny. However, their effect may have severe impacts on the solution accuracy. Figure 2 is showing a similar problem. Connecting patches have been created at the corner of the wings between the top and bottom to obtain a tight model. The coarse discretization makes the wings almost entirely stiff, while the remaining patches have a very large refinement and respective deformability.

### 3.7 Trimming

Trimming is independent of the knot span description of the NURBS surface and can therefore arbitrarily cut through them. Knot spans with a small physical domain may result from this operation and cause additional challenges for the FE solver regarding the stability of the system. There are control points that have minor influence on the shape/solution resulting in very small entries in the respective stiffness matrix, which is consequently badly conditioned. The

<sup>3</sup> E.g. the penalty approach is achieving continuity by indicating a spring between the domains. The selected spring stiffness, namely penalty factor is apparent in the solution. With patches of small sizes this effect might dominate the results.



**Fig. 7** Trimmed example with possible unwanted interaction within the domain

resulting solution may contain close to infinite values. Furthermore, the results for the degrees of freedom on those control points may differ a lot in all iterations and obstruct the convergence in the solver. Different technologies have been proposed to limit the deflections of those control points, e.g. modifying basis functions [37, 69, 84], stabilization terms [49] or conditioning of solvers [39]. Another issue of trimming descriptions is presented in Fig. 7. A rectangular cantilever plate split in the middle is modeled by one trimmed patch generating two almost independent cantilevers. One of them is loaded at the tip and should consequently deform whereas the other should not. Figure 7b shows the simulation results with an unwanted interaction between the two ends, whereas Fig. 7c is sufficiently refined to properly simulate this problem. This example shall point out the apparent problem, which becomes even more severe if the trimming is not aligned with the knot span description since the necessary refinement to generate independent knot spans that do not share control points and consequently to decouple the boundaries may spread out over the whole domain due to the tensor product structure of NURBS surfaces.

Another problem with trimmed domains is that even though analysis may be resolvable for some models, the given geometry description might be unsuitable, e.g. not

sufficiently refined or distorted, which would directly affect the solution accuracy [11, 12, 17].

In the following, two examples shall be discussed, which illustrate this problem. First, a plate, which is simply supported on two sides, is tested under bending with four different parametric descriptions, whereby all have the same geometric shape of a rectangle (see Fig. 8). The refinement convergence of the maximal displacements is shown in Fig. 8e. The convergence of the trimmed example (see Fig. 8c) is slightly faster compared to the untrimmed configuration (see Fig. 8a). The patch with a highly distorted NURBS representation shows significantly worse results for a similar amount of control points (see Fig. 8b), whereas here the trimmed counterpart is slightly better (see Fig. 8d). It shall be noted that comparing the degree of distortion is not included in this study. This example shall point out that the solution of a trimmed problem is not necessarily weaker than the one from a untrimmed example. However, it needs to be considered that the results may vary, whereby it is important to know the reasons for this difference.

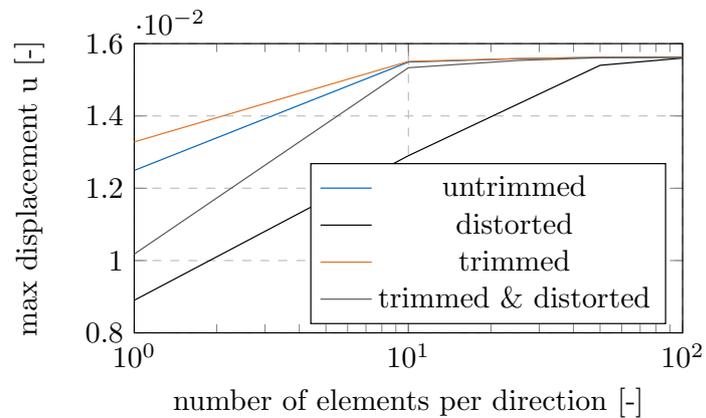
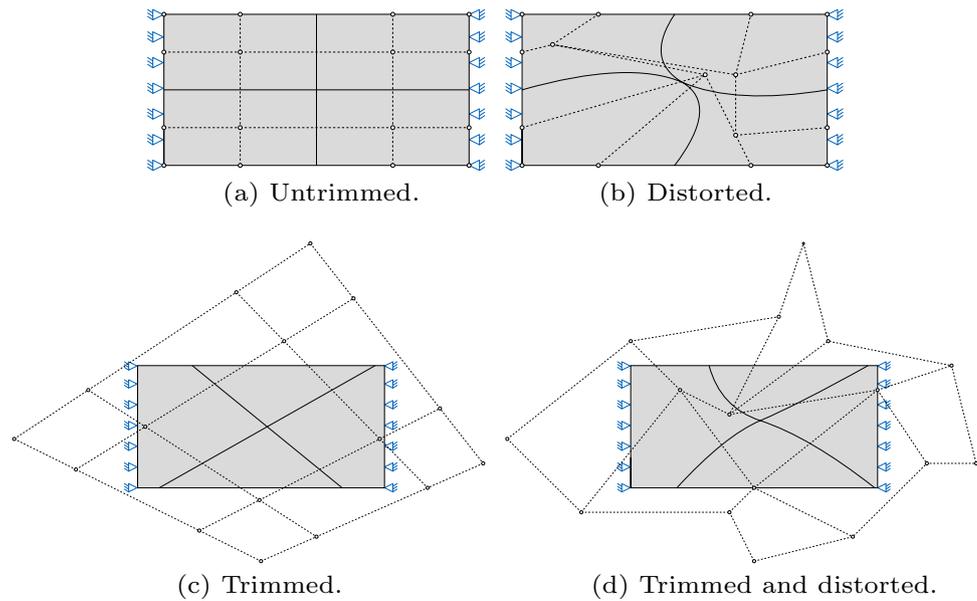
Another example containing a similar issue due to trimming and eventual quantitatively weaker results than the untrimmed model is the formfinding problem [10] of a membrane plate with out-of-plane loads [1, 63]. Given that the radius of the support and the pressure load relate via the boiler formula, the outcome of this problem should be a perfect hemisphere. This solution is approximated almost perfectly as for the untrimmed configuration Fig. 9a. In contrast, the trimmed-multipatch scenario shows severe problems in the description of the wanted shape Fig. 9c. With an increased refinement this problem may be smoothed out, but will always be slightly apparent. Additionally, the multipatch coupling is implemented by the penalty approach (see [12, 18, 50, 51, 63] for IBRA), which results, dependent on the selection of the penalty factor in a corrupted stiffness at the interface. Accordingly, if the solutions are not satisfactory it may be advantageous to use different coupling approaches (see e.g. [2, 4, 30, 74]), or a more advanced penalty computation [61], to improve the numerical conditioning and results.

As a summary, it shall be noted that trimming introduces flexural limitation, which may require a higher refinement levels including more sophisticated solution procedures and stabilization. However, it still presents as a great feature which allows the direct simulation of complex models without any re-discretization.

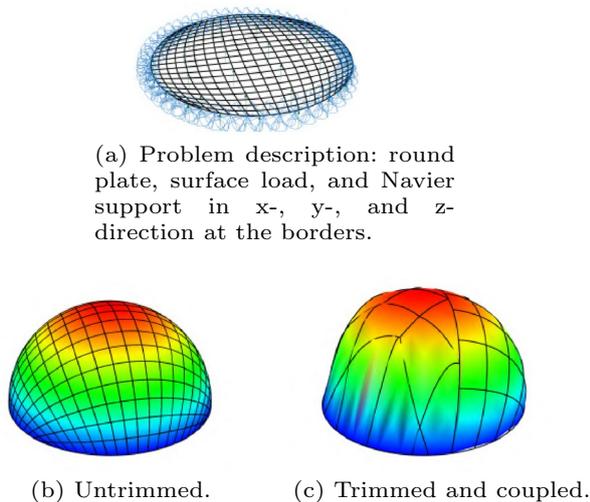
### 3.8 Preparation of the model

Some of the previously mentioned issues may suggest a revision of the B-Rep-based CAD model before eventual simulation. One proposed possibility is the patch

**Fig. 8** Comparison of trimmed, untrimmed, and trimmed and distorted configurations. All patches are modeled with a polynomial degree of the shape functions of  $p = 2$ . A continuous surface load of  $l = 2$  is applied on the physical domain, pointing out of plane, to apply bending to the system



(e) Convergence study of the maximum displacement  $u_z$ .



**Fig. 9** Formfinding [10] under pressure load [1, 63]

degeneration of many patches towards few patches [68]. Typically, this introduces slight chordal errors, however, the positive effects may outweigh the instabilities from the original model. The outcome of this operation is a continuous distribution of control points throughout the patches and a reasonable polynomial order. The model preparation improves the stability of the simulation for many scenarios. However, it creates a modification of the original CAD model. It is generally not trivial to reverse this operation and therefore, updates such as optimizations or form-finding may not be included properly in the original model. Accordingly, it shall be noted that this procedure adds complexity in the CAD-integrated (automated) workflow and opposes the original isogeometric concepts.

The long-term goal should be to incorporate the requirements for an eligible simulation within the design phase and not as part of the pre-processing in the analysis step since

one model would be sufficient and iterations between the two disciplines become simplified.

Accordingly, knowing about the mathematical properties of a chosen design path, as explained within this chapter, shall help to select the appropriate methods for a quantitatively good and seamless design-through analysis workflow within CAD.

## 4 Interaction with the CAD-data

Generally, multiple approaches exist for the user interaction with a CAD model, which are also dependent upon when the interaction with the models happen, as e.g. design and simulation related user interaction at the same time or first design second user interaction by physical data enhancement. Most script-based CAD software would allow to genuinely add the entities within the geometry creation process, as the user data enhancement can happen at the same time. Most systems rely on a graphical interface and do not provide access to their internal data management for the user. Additional procedures have to be incorporated. Within this publication, three possible ways shall be outlined and discussed: the command-based, the GUI-based and the visual-programming-induced process.

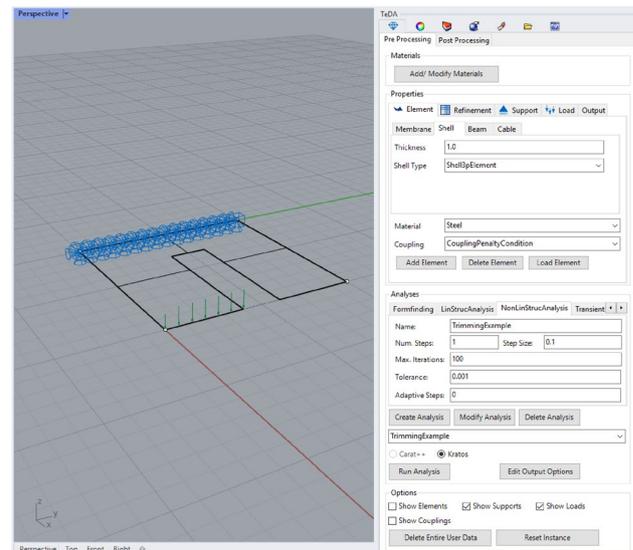
### 4.1 Command-based enhancements

Commands are the least complex interaction possibility. Each command is provided in such a form that it demands certain input, as e.g. the selection of specific geometry objects of defined types, or simpler inputs as strings or numbers. The output is either saved as a user data enrichment of the geometrical entities or as a general data collection at the plugin level.

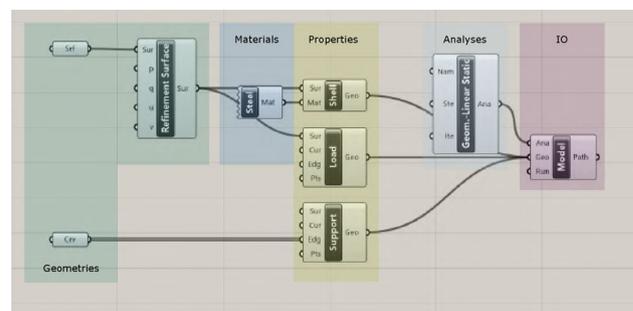
Commands are typically interesting once few operations need to be processed, however, they might appear heavy and blocking if much information needs to be collected. Additionally, an overview of all conducted steps and the supplied information is not available without additional data visualization in the viewport or reporting commands, which is generally suitable, however it lacks a bit of overview.

### 4.2 GUI-based enhancements

A GUI shall help the user to quickly get an overview of the available options and modification parameters. This allows a fairly easy introduction to a novel software, helps the unlearned user and should also reduce possible misunderstanding within the modeling process. An example of a GUI for the structural analysis with thin-walled structures is presented in Fig. 10a.



(a) GUI-based.



(b) Visual programming-based.

**Fig. 10** User interaction avenues within the pre-processing for the trimmed example of Fig. 7

Within Rhino, it turned out that having a hybrid format between the GUI and the commands results in a very efficient workflow. Accordingly, the button of the GUI would call certain commands and provide selected properties, which were supplied in the GUI. The command collects the missing information by letting the user select the respective geometrical items.

### 4.3 Visual programming-based enhancements

Lately, parametric design has evolved significantly allowing to make models dependent upon defined parameters [56, 67]. This allows to easily adapt to changing needs throughout the design process. Consequently, it helps to streamline certain drafts by defining the relevant sizes as parameters and later on only adjust those parameters for an automatic update without having to manually redraw the entire structure.

A popular parametric design tools is the Grasshopper [28], belonging to Rhino [66]. It comes along with a

**Table 1** Summary of usability of user interfaces

	Simplicity	Overview	Consistency	Extensibility	Design-through-analysis
Commands	+	o	–	– –	o
GUI	++	o	o	o	o
Parametric	+	++	++	++	++

– –  $\hat{=}$  high effort; –  $\hat{=}$  extra effort; o  $\hat{=}$  achievable, not optimal; +  $\hat{=}$  good; ++  $\hat{=}$  very good

so-called visual programming platform, which allows to include various components, being e.g. geometrical entities or physical properties. It is upon the user to connect those, by the developers defined interfaces, without, or with barely interacting with the source code, which gives simplified but enormous possibilities.

Within the scope of Grasshopper-based CAE, there already exist many structural analysis tools such as Kangaroo [40], Karamba [41] and Kiwi!3d [44],<sup>4</sup> whereby the latter is also based upon IGA. Within architecture and some fields of engineering, those plugins have been established as essential parts within the design process.

Within the scope of visual programming, the previously discussed commands are exchanged by objects. Those take certain geometrical entities as inputs and pass them along with the physical enrichments. Finally, all data is collected within a model object, where all entities are combined and an exchange with the solver may be processed.

#### 4.4 Discussion about the various interaction processes

It shall be noted that each of the previously mentioned procedures has its advantages and accordingly, use cases. The command and GUI-based approaches are typically essential once the design of the CAD model has been completed and is only imported and processed for the analysis model. In that scenario, selecting the according geometries is pretty simplified and fast, even for untrained users. A summary of some of the features of the interfaces is discussed within Table 1. Simplicity shall cover the questions of how fast a model may be set up and how simple it is to get started with it. The overview gives insight into how easy it is to oversee all selected options. Where Commands and GUI may

only visualize part of the setting options, parametric design allows to always see the chosen parameters. Consistency defines how difficult it is to set up all parameters for a simulation and maintain them during the design process. Once a model is done and needs to be extended, diverse possibilities arise in the different avenues. Finally, IGA comes with the advantage of design-through-analysis features while keeping the original CAD model. Those features can be applied to a full extent with the parametric design.

On the other end, once the analysis shall be included within the design process and the model variations are more than moving some control points, the visual programming turns out as the most efficient way. Specifically, once topology is changed, Rhino might have issues in keeping the correct data in place. In contrast, Grasshopper rebuilds the whole analysis model from the changed parameters. This is possible since the relation between the geometries and physics is not stored w.r.t. to an object id but as a function of the geometric input.

## 5 Software architecture

Consequently, a possible avenue to realize the proposed features, required for a consistent pre- and post-processing of IGA shall be discussed.

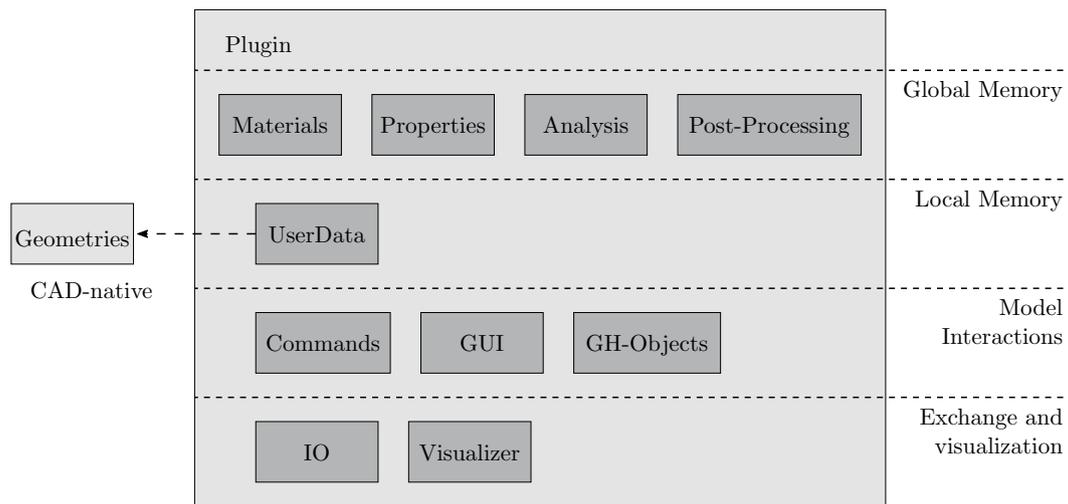
### 5.1 Data structure

The groups of classes (in object-oriented programming manner) are presented within Fig. 11. There are the global memory, which is accessible from any place, the local memory, which is attached to specific entities and the model interactions, which describe the possibilities to enhance, or modify data trees. Global memory means that data is stored once and accessible from a specific location. Local memory is attached to specific geometrical items and can respectively easily be connected. However, it may appear to create a data overhead for multiplicative data. Additionally, if certain information is changed, it needs to be ensured that all entities are modified, which in contrast is easier within the global memory.

The groups of classes shall be outlined within the following:

*Plugin* The plugin is the namespace and place of the entire code, which shall be discussed here. Additionally, it is the address for global memory, which is accessible from any instance. Some of its entities are attached to specific geometries, however, some data is not specific for an individual geometry object but appears multiple times. To avoid unnecessary large exchange file sizes due to data duplicates, it is beneficial to store the information once in

<sup>4</sup> Kiwi!3d is based upon a similar Rhino-plugin concept as Cocodrilo and was designed and optimized to work seamlessly with Carat++ [14], whereby Cocodrilo relies on the application of KratosMultiphysics.



**Fig. 11** Software architecture of Cocodrilo

the global memory and access it by pointers. This is e.g. the case for *Materials* and *Properties*.

*Properties* Those contain the kinematic formulations and their respective characteristics. Furthermore, the properties might also indicate global solver attributes, as e.g. required types of degrees of freedom, post-processing values, linear-solver suggestions and compatibilities, and more.

*Materials* Within the finite element software, those are often not copied but passed as reference to all elements, because depending on the chosen constitutive law this might contain many parameters. Accordingly, the plugin tries to reduce the amount of duplicated information by referencing it at the geometry level and storing it globally to avoid redundancies.

*Analysis* Defines the selected analysis type, e.g. structural analyses, transient analyses, and optimizations or form-finding.

*Post-Processing* The post-processing can generally be implemented either within the local memory and attached to the respective geometries or it can be collected at global memory level and then be distributed once specific options are selected. Within the presented implementation, it was chosen to use the global memory, as it turned out as less complicated to structure the available post-processing options for the user. This avoids large lists of possible post-processing options that are only partially filled by the respective analysis. This becomes especially useful when the initial analysis set-up, i.e. the pre-processing, is not available to filter the options.

The post-processing operations are typically quite heavy objects since those may contain the result information of the whole model with a large set of control points, or evaluation points.

*UserData* Directly connects *Materials*, *Properties* and additional geometrical information with the respective geometry items.<sup>5</sup> Furthermore, it may be used to attach additional, mostly embedded entities, as points or curves. It is also used to provide identical identifiers to the objects which is for example beneficial to internally maintain consistent coupling information.

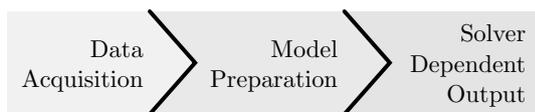
*Commands* Are designed with the respective needs of the corresponding CAD program and should follow the suggestions from Sect. 4.1.

*GUI* The *GUI* should be adapted according to the respective use case as described in Sect. 4.2. Within Cocodrilo in Rhino, this *GUI* is a Windows form providing all features. Essential for a seamless work is a guidance through *Delegates*, which keeps all *GUI*-forms up-to date after data updates.

*GH-Objects* The grasshopper objects are a generalization of the previously defined *Commands*, which take inputs and allow outputs. In that way, a complete simulation with pre- and post-processing can be established. Accordingly, it allows a great flexibility within a still reliable and greatly observable frame.

*IO* The *IO* functions are the main function for the interfaces with the solvers. Those transfer the previously collected data into appropriate input files. Accordingly, once multiple solvers would be required, those can be enhanced by implementing an additional *IO*-class.

<sup>5</sup> It shall be noted that within certain transform operations, as copies up to trimmings in some cases the user data might get lost. Stable copy constructor operations are required. Within Cocodrilo this is processed within the *EventWatcher* class as *onReplaceObject*.



**Fig. 12** Steps towards an output/solver exchange

*Visualizer* The visualizer shows entities as element formulations, supports, loads or other features to have an visual impression of the selected options and conducted assignments. This happens within the display conduit and does not add geometries to the document but only adds them to the view port, to avoid problems with the user interaction.

## 5.2 Algorithms

Within this section and Fig. 12, some of the most crucial steps towards a creation of feasible input files, which require additional attention, shall be briefly discussed.

### 5.2.1 Data acquisition

One of the major challenges within the pre- and post-processing is the data acquisition, which is one of the biggest differences between the Rhino-based commands and GUI and the Grasshopper induced GH-objects. Within the Rhino-based plugin, the geometrical data can be obtained from the active document. The geometries contain the user data with all required information. On the other side, within Grasshopper, it turned out as feasible to collect all simulation-relevant geometries within a final *Model* GH-object. This collection may then be processed with the same procedures as the active document collected selection.

### 5.2.2 Model preparation

Some of the essential model preparation issues are mentioned within Sect. 2, whereby it shall be noted that some of the discussed points may be resolved, whereas others remain limitations within CAD-integrated analysis.

The input within this operation is typically a list of B-Reps, containing mostly single or connected faces, curves and points. At this stage, it is recommended to provide unique identifiers to all entities.<sup>6</sup> This is essential to

<sup>6</sup> It has been investigated to provide identifiers at the geometry construction stage, or once user data is defined and enhanced. Keeping the information consistent during the pure design process is rather complex since even simple operations as e.g. move, scale would recreate such a user data and would consequently disturb the system.

consistently store and transfer all information, including topology, to the solver.

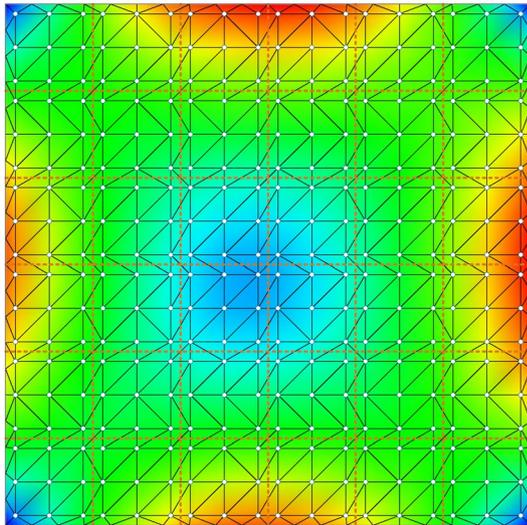
This step is accompanied by the detection of additional coupling cases, as described in Sect. 3.4. It shall be noted that there are significant differences between the Rhino- and the Grasshopper-based approaches. Whereas, within Rhino, most of the connection cases may have been provided already, within Grasshopper, those may often be undefined. In Grasshopper, it is possible to join geometries, similarly to Rhino. However, it is fairly difficult to select embedded or trim geometries, as those are defined in the parameter space and can only be selected by an index. This index may not be determined in the beginning, or may change through joins or when iteratively running the system with slightly changed parameters. This implies difficulties for the user. Since information is stored on separate geometries and not attached to the same objects using e.g. edge properties, the connectivity between geometries have to be found in an additional step. Accordingly, the number of intersection scenarios within the Grasshopper implementation are often more extensive. However, due to differences within the needs of visual connectivity and physical connections, it is still inevitable to do checks if geometries are intersecting, also for the Rhino approach.

### 5.2.3 Solver-dependent output

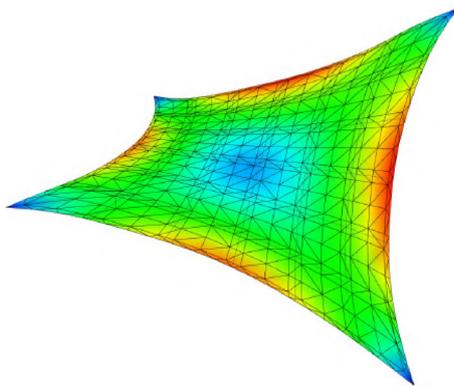
Once the data is collected and the model is prepared, the output may be processed in a solver specific way. A possible implementation for the essential geometric information is shown in [74], whereby, the solver specific settings would inevitably need to be taken into account within the data enhancements.

## 5.3 Post-processing

The aim of the proposed CAD-integrated analysis environment is a visualization upon the initial and the deformed geometries. The general shapes may be expressed by applying the resulting displacements on the control points. Furthermore, to express the magnitudes of the results a color plot shall be introduced. This contour plot is constructed by a mesh to have a fast and continuous visualization, similar to the rendering. This mesh is connected to the integration points of the simulation. Those points are often not defined upon the borders since it is not required for the integration of a surface using Gauss interpolation. Thus to complete the mesh further points need to be introduced at the borders of the shapes (Ideally, those additional points are generated by the solver and stocked with respective results to avoid an extrapolation of information). With this set of points, a meshing operation is processed. The possible outcome



(a) Undeformed post-processing mesh (knot spans are displayed in orange).



(b) Deformed post-processing mesh.

**Fig. 13** Post-processing mesh

is displayed within Fig. 13a. The corresponding deformed shape is shown in Fig. 13b.

This post-processing mesh may be used for nodal and integration point variables. The nodal variables are interpolated to the respective locations, whereas integration point variables can be applied directly from the post-processing of the elements. Generating a direct mesh of the integration point variables turned out as more beneficial than mapping analysis results to the program inherent visualization mesh since costly search operations can be omitted.

## 6 Application examples

Cocodrilo has initially been developed for research purposes, mainly with a focus on methodological developments within the isogeometric analysis. The plugin Kiwi!3d [44] evolved for the in-house FE-kernel Carat++ that contains the initial IGA developments. The presented open-source software Cocodrilo relies on KratosMultiphysics, which is open-source and covers additional isogeometric analyses developments.

Some research, which has been conducted with Kiwi!3d, is e.g. the parametric design of membranes [26, 27] or light-weight structures generally [25].

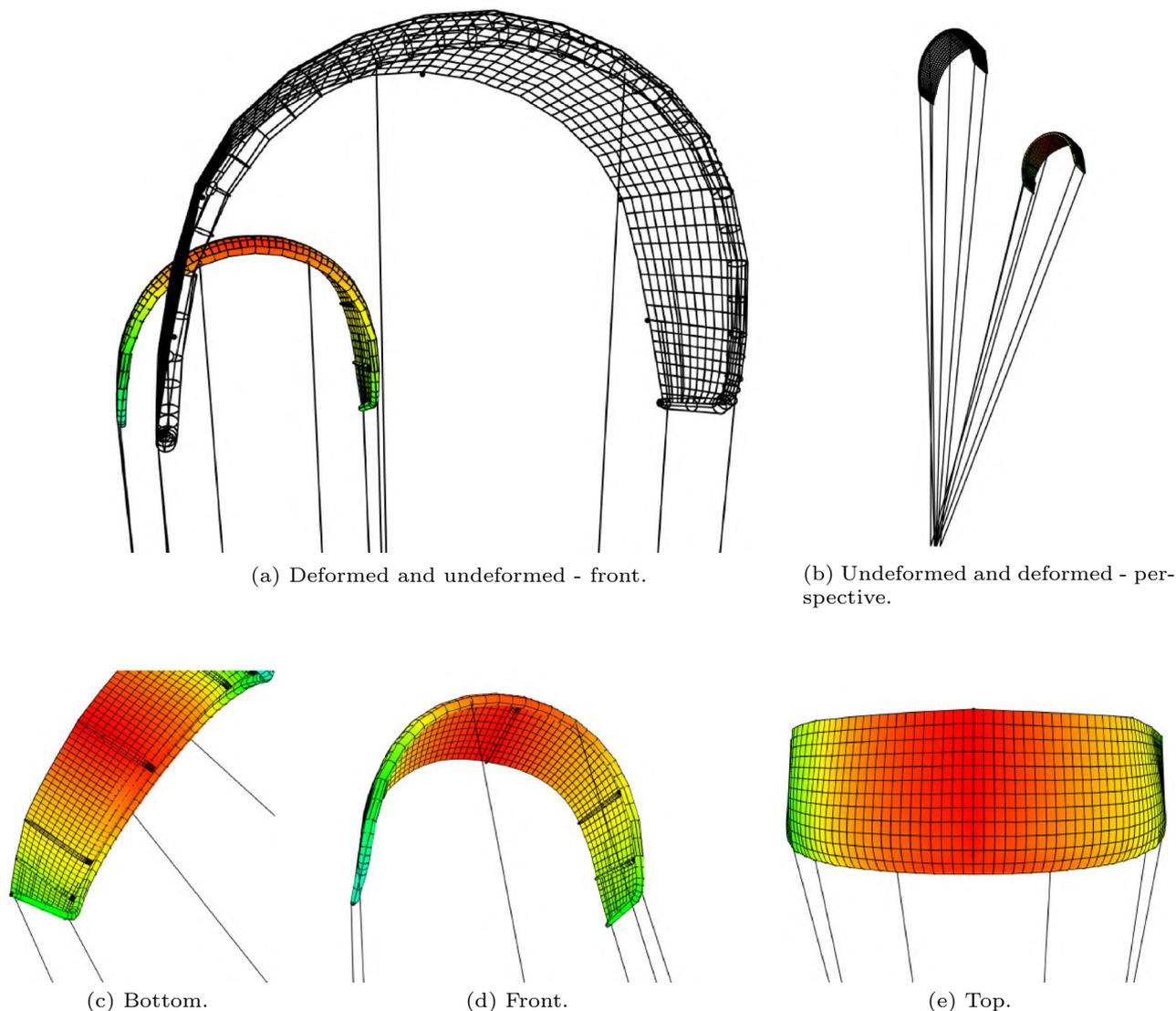
In the following, some application examples making use of the CAD-integrated simulation will be shown.

### 6.1 Kite structure

The first example is a kite structure (see Fig. 14) showing the application of several different structural elements within a geometrically nonlinear analysis. It consists of:

- 1D cable elements for the ropes.
- Pre-stressed 2D membrane elements for the bars of the kite. Those tube-like parts of the kite are prestressed with internal pressure, which provides the expected stiffness.
- 2D shell elements with a very low curvature stiffness for the sails. The sail of the modelled kite is made of a very thick material which does provide a little stiffness in curvature direction.
- Coupling of 1D with 2D elements.
- Coupling 2D elements, between membrane and membrane, membrane and shell, and shell and shell. The connection between the kite bars and the sails are mostly T-joints.
- The accuracy of the connections have been varying, which required a very accurate design of the connections.
- The kite has been modelled in a numerical wind-channel with CFD. The obtained pressures from the simulation have been applied on the sails with a dependency on the global  $x$ -,  $y$ -, and  $z$ -coordinates.
- The structure has been modelled in a geometrically non-linear analysis.

For this example, two interfaces have been employed: the GUI and parametric GH plugin. It turned out that the parametric interfaces had great advantages to study effects of different chosen discretizations (e.g. different patch creations, varying refinements) since the geometry is only one parameter that is added to a structural element or boundary.



**Fig. 14** Kite structure with static pressure. The color plots are indicating the displacements within the sails

In contrast, the GUI-based modeling is set up on a existing geometry as data container. The stored physical information has to be regenerated.

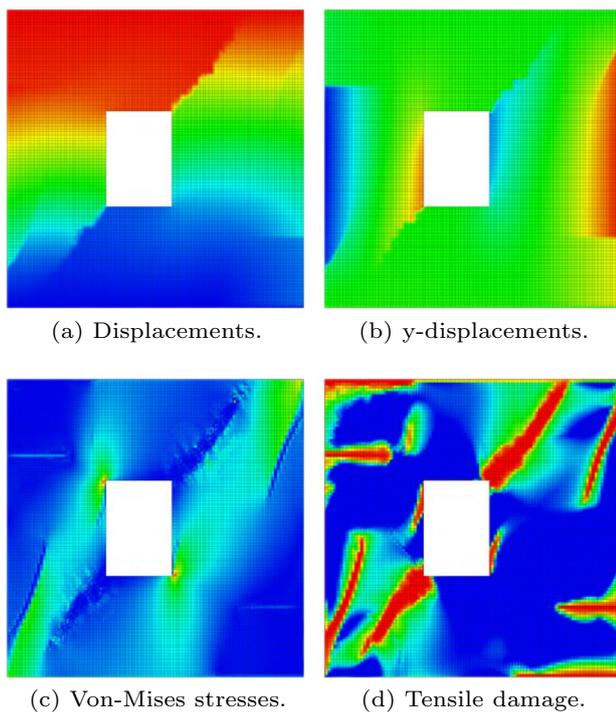
## 6.2 Masonry wall

This example is presented to show the various post-processing possibilities upon the geometrical NURBS-based entities. The physical correctness shall not be examined. Therefore, a masonry wall with hole, which is sheared off until damage, is chosen. The respective results are presented within Fig. 15. The result plot of the total displacement is shown in Fig. 15a and only the horizontal displacement in Fig. 15b. Additionally, two internal stress

properties are presented, the von-Mises stresses, Fig. 15c, and the discrete tensile damage parameter, Fig. 15d. It shows that variables upon integration points may be displayed (damage and stresses) equally as evaluated values of the degrees of freedoms, such as the displacements, but also selected displacement directions. Continuous and discrete result behavior can be depicted.

## 6.3 Sandwich panel

The subsequent example treats a sandwich panel where the upper and bottom surfaces are coupled by a honeycomb pattern. The shape of the bottom NURBS surface shall be form found with a shape optimization procedure (see [10]),



**Fig. 15** Various results for a masonry shear wall test

with a minimization of the strain energy amid constant pressure load (see Fig. 16b). The result of the shape optimization is then used in a second analysis to determine the structural behavior of the sandwich panel under a uniform surface load on the upper surface (see Fig. 16c).

This example shall demonstrate the features of a parametric staged analysis, which involves many of the benefits of a CAD-integrated IGA analysis. The respective setup of the example is displayed in Fig. 16a. It shows the two consecutive analyses, the structural optimization (see Fig. 16b) and the structural analysis (see Fig. 16c). The latter applies the outcome of the first one and automatically constructs the corresponding honeycomb pattern for the connection to the upper plate. The resulting displacement plots (qualitative) of the whole sandwich panel are displayed in Fig. 16d–i for different settings of the initial form-finding and respectively varying shapes of the bottom plate. In this way, material may be reduced, while keeping displacements similar or even lower.

It shall be noted that also different optimization/form-finding techniques (e.g. hanging models) could be embedded easily in the design. The parametric model would enable a simplified exchange between the varying results. Hereby, the chosen algorithm could be selected empirically or even by defined criteria (e.g. intended load case classification). A selection may also involve visual features. Further

parameters, which may be applied in the model, could cover e.g. variation of the internal filler of the panel structure with different size or patterns such as quads, diamonds, free forms. Therefore, the influence of distinctive shapes or a varying amount of reinforcement could be studied easily.

This example has been selected to present the embedding of certain analysis procedures in the CAD-integrated design environment. It shows that by parametric design and analysis modeling it is possible to combine various simulations within a single workflow. Changes are adopted without effort, and the final model considers them including all corresponding changes in the design chain. This makes it easy to erect structures that adapt to varying environments and load cases.

## 7 Conclusion and outlook

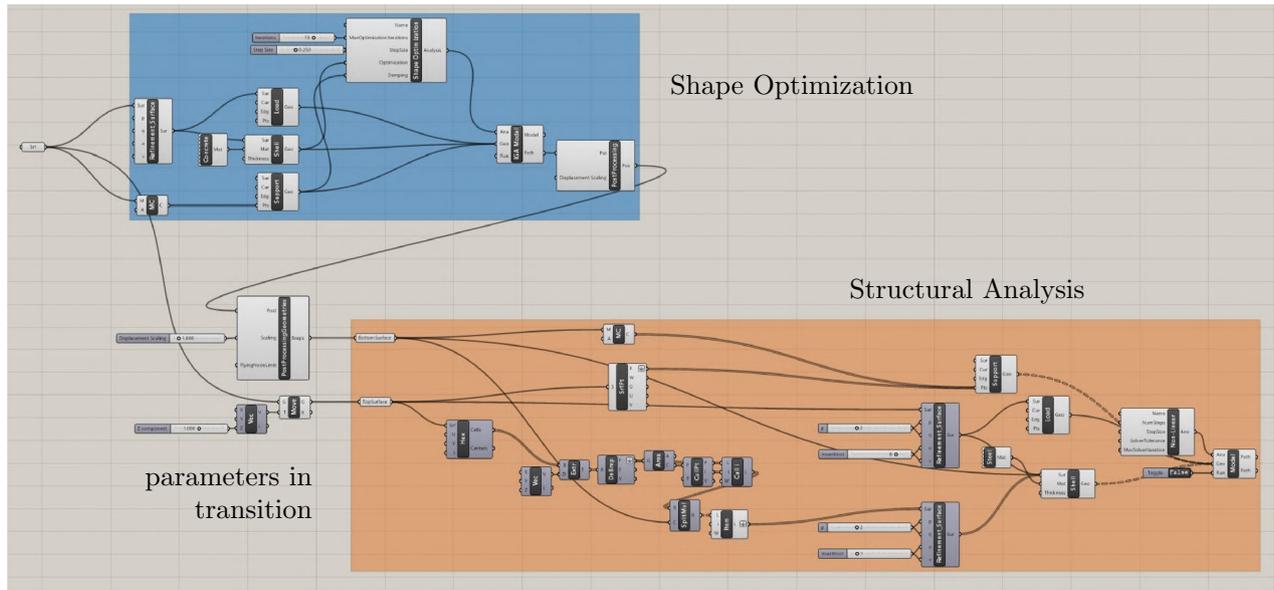
While IGA is gaining significant attention within the previous years, yet few pre- and post-processing tools exist. Even though, those tools are essential for proofing the applicability of the newly developed methods in a larger context. To help other developers to integrate IGA inside their industrial projects, required procedures and a software architecture have been presented within this publication.

Challenges and complexities within the geometrical assessment have been presented along the experience from practice, including eventual avenues for possible fixes. Additionally, this paper provides some geometrical implications which are required for a flawless design-through analysis workflow with IGA. When considering these principles, extensive model cleaning becomes obsolete. Thus, those may be used as recommendations for an IGA suitable geometrical design.

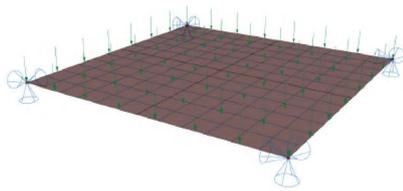
The open-source plugin Cocodrilo [75] is discussed, which implements the introduced approaches and shows the possibilities within architecture and engineering that are arising alongside the developments with IGA and IBRA.

The presented pre- and post-processing tools are currently available for thin-walled 1D and 2D structures. Additional custom element formulations for these geometries can easily be added to both the solver and the pre- and post-processing tool. The solver requires the corresponding stiffness matrix and residual force vector at the integration points. Functions regarding NURBS and standard (Gauss) integration are already provided. The pre-processor requires additions in the GUI and the IO functions to gather complementary information and transfer them to the solver.

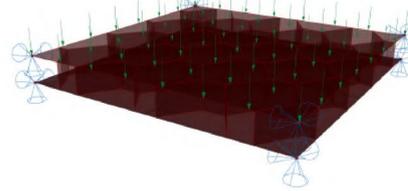
An extension towards solid formulations would enhance the usability significantly and shall be suggested to immediate subsequent research. Furthermore, the implementation



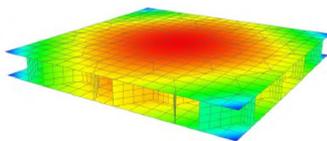
(a) Parametric staged design of sandwich panel.



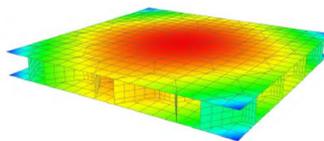
(b) Setup of structural optimization.



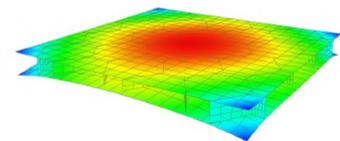
(c) Setup of entire sandwich panel.



(d) Non-optimized - perspective.



(e) Optimized solution 1 - perspective.



(f) Optimized solution 2 - perspective.



(g) Non-optimized - front.



(h) Optimized solution 1 - front.



(i) Optimized solution 2 - front.

**Fig. 16** Parametric staged design of sandwich panel

and accordingly coverage of different physics, towards a CAD-integrated multiphysics plugin shall be proposed.

**Acknowledgements** The authors thank the reviewers whose constructive comments improved the quality of the article. We would like to give thanks to Benjamin Best for his extensive work on designing, modeling and simulating the Kite structure (see Sect. 6.1). The authors gratefully acknowledge support from the *Deutsche Forschungsgemeinschaft* (DFG) as part of the SPP 2187 with the project “Der digitale Baukasten – Simulationsbasierte Modelle und Methoden für den Entwurf modularer Tragsysteme aus Beton” and the project “Eine Methode zur effizienten Simulation grosser Mauerwerksscheiben unter exzentrischer und/oder zyklischer biaxialer Beanspruchung auf Grundlagewirklichkeitsnaher Kleinkörperversuche” (BL306/34-1).

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Code availability** The code of Cocodrilo is available at [75]. The github repository shall be used to attract more developers with the aim of a growing community.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interests in this paper.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Apostolatos A (2019) Isogeometric analysis of thin-walled structures on multipatch surfaces in fluid-structure interaction. Dissertation, Technische Universität München, München
2. Apostolatos A, Schmidt R, Wüchner R, Bletzinger K-U (2014) A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis. *Int J Numer Methods Eng* 97(7):473–504
3. Arabshahi S, Barton DC, Shaw NK (1993) Steps towards CAD-FEA integration. *Eng Comput* 9(1):17–26
4. Babuška I (1973) The finite element method with Lagrange multipliers. *Numer Math* 20:179–192
5. Bauer AM (2020) CAD-integrated isogeometric analysis and design of lightweight structures. Dissertation, Technische Universität München, München
6. Bauer AM, Breitenberger M, Philipp B, Wüchner R, Bletzinger K-U (2016) Nonlinear isogeometric spatial Bernoulli beam. *Comput Methods Appl Mech Eng* 303:101–127
7. Bauer AM, Wüchner R, Bletzinger KU (2020) Weak coupling of nonlinear isogeometric spatial Bernoulli beams. *Comput Methods Appl Mech Eng* 361:112747
8. Benson DJ, Bazilevs Y, Hsu M, Hughes TJR (2010) Isogeometric shell analysis: the Reissner–Mindlin shell. *Comput Methods Appl Mech Eng* 199(5–8):276–289
9. Bettig B, Bapat V (2006) Integrating multiple information representations in a single CAD/CAM/CAE environment. *Eng Comput* 22(1):11–23
10. Bletzinger K-U (1990) Formoptimierung von Flächentragwerken. PhD thesis, Universität Stuttgart
11. Breitenberger M (2016) CAD-integrated design and analysis of shell structures. Dissertation, Technische Universität München, München
12. Breitenberger M, Apostolatos A, Philipp B, Wüchner R, Bletzinger K-U (2015) Analysis in computer aided design: nonlinear isogeometric b-rep analysis of shell structures. *Comput Methods Appl Mech Eng* 284(284):401–457
13. B. CAE. [www.beta-cae.com/ansa.htm](http://www.beta-cae.com/ansa.htm)
14. Carat++. <http://www.cce.ed.tum.de/en/st/software/research/carat/>
15. Chasapi M, Mester L, Simeon B, Klinkel S (2022) Isogeometric analysis of 3D solids in boundary representation for problems in nonlinear solid mechanics and structural dynamics. *Int J Numer Methods Eng* 123(5):1228–1252. <https://doi.org/10.1002/nme.6893>
16. Cohen E, Martin T, Kirby RM, Lyche T, Riesenfeld RF (2010) Analysis-aware modeling: understanding quality considerations in modeling for isogeometric analysis. *Comput Methods Appl Mech Eng* 199:334–356
17. Coradello L, D’Angella D, Carraturo M, Kiendl J, Kollmannsberger S, Rank E, Reali A (2020) Hierarchically refined isogeometric analysis of trimmed shells. *Comput Mech* 66(2):431–447
18. Coradello L, Kiendl J, Buffa A (2021) Coupling of non-conforming trimmed isogeometric Kirchhoff–Love shells via a projected super-penalty approach. *Comput Methods Appl Mech Eng* 387:114187
19. coreform IGA. <https://coreform.com/>
20. Cottrell JA, Reali A, Bazilevs Y, Hughes TJR (2006) Isogeometric analysis of structural vibrations. *Comput Methods Appl Mech Eng* 195(41–43):5257–5296
21. Dadvand P (2007) A framework for developing finite element codes for multi-disciplinary applications. Dissertation, Universidad Politecnica de Catalunya, Barcelona
22. Deng Y-M, Lam YC, Tor SB, Britton GA (2002) A CAD-CAE integrated injection molding design system. *Eng Comput* 18(1):80–92
23. Düster A, Parvizian J, Yang Z, Rank E (2008) The finite cell method for three-dimensional problems of solid mechanics. *Comput Methods Appl Mech Eng* 197(45–48):3768–3782
24. Echter R, Oesterle B, Bischoff M (2013) A hierarchic family of isogeometric shell finite elements. *Comput Methods Appl Mech Eng* 254:170–180
25. Goldbach A-K, Bauer AM, Wüchner R, Bletzinger K-U (2020) CAD-integrated parametric lightweight design with isogeometric b-rep analysis. *Front Built Environ* 6
26. Goldbach A-K, Bauer AM, Wüchner R, Bletzinger K-U (2020) Der cad-integrierte parametrische entwurfsprozess von membrantragwerken. *Stahlbau* 89(8):684–693
27. Goldbach A-K, Bletzinger K-U (2019) CAD-integrated parametric design cycle for structural membranes. *J Int Assoc Shell Spat Struct* 60(202):266–272
28. Grasshopper. <http://www.grasshopper3d.com>
29. G+Smo. <https://github.com/gismo>
30. Guo Y, Ruess M (2015) Weak Dirichlet boundary conditions for trimmed thin isogeometric shells. *Comput Math Appl* 70(7):1425–1440
31. Herrema AJ, Johnson EL, Proserpio D, Wu MC, Kiendl J, Hsu M-C (2019) Penalty coupling of non-matching isogeometric Kirchhoff–Love shell patches with application to

- composite wind turbine blades. *Comput Methods Appl Mech Eng* 346:810–840
32. Hirschler T, Bouclier R, Duval A, Elguedj T, Morlier J (2019) The embedded isogeometric Kirchhoff–Love shell: from design to shape optimization of non-conforming stiffened multipatch structures. *Comput Methods Appl Mech Eng* 349:774–797
  33. Hirschler T, Bouclier R, Duval A, Elguedj T, Morlier J (2019) The embedded isogeometric Kirchhoff–Love shell: from design to shape optimization of non-conforming stiffened multipatch structures. *Comput Methods Appl Mech Eng* 349:774–797
  34. Horger T, Kollmannsberger S, Frischmann F, Rank E, Wohlmuth B (2014) A new mortar formulation for modeling elastomer bedded structures with modal-analysis in 3d. *Adv Model Simul Eng Sci* 1:18
  35. Hsu M-C, Wang C, Herrema AJ, Schillinger D, Ghoshal A, Bazilevs Y (2015) An interactive geometry modeling and parametric design platform for isogeometric analysis. *Comput Math Appl* 70(7):1481–1500
  36. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput Methods Appl Mech Eng* 194(39–41):4135–4195
  37. Höllig K, Apprich C, Streit A (2005) Introduction to the web-method and its applications. *Adv Comput Math* 23(1–2):215–237
  38. I.O. for Standardization (2016) Step, iso 10303-21
  39. Jomo J, de Prenter F, Elhaddad M, D'Angella D, Verhoosel C, Kollmannsberger S, Kirschke J, Nübel V, van Brummelen E, Rank E (2019) Robust and parallel scalable iterative solutions for large-scale finite cell analyses. *Finite Elem Anal Des* 163:14–30
  40. Kangaroo3d. <http://kangaroo3d.com>
  41. Karamba3D. <http://www.karamba3d.com>
  42. Kiendl JM, Bazilevs Y, Hsu M-C, Wüchner R, Bletzinger K-U (2010) The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Comput Methods Appl Mech Eng* 199(37–40):2403–2416
  43. Kiendl JM, Bletzinger K-U, Linhard J, Wüchner R (2009) Isogeometric shell analysis with Kirchhoff–Love elements. *Comput Methods Appl Mech Eng* 198(49–52):3902–3914
  44. kiwi!3d. <http://www.kiwi3d.com>
  45. Klinkel S, Chasapi M (2020) Isogeometric analysis of solids in boundary representation. Springer International Publishing, Cham, pp 153–197
  46. Klinkel S, Chen L, Dornisch W (2015) A NURBS based hybrid collocation–Galerkin method for the analysis of boundary represented solids. *Comput Methods Appl Mech Eng* 284:689–711. Isogeometric Analysis Special Issue
  47. KratosMultiphysics. <https://github.com/KratosMultiphysics/Kratos>
  48. Kudela L, Kollmannsberger S, Almac U, Rank E (2020) Direct structural analysis of domains defined by point clouds. *Comput Methods Appl Mech Eng* 358:112581
  49. Larsson K, Kollmannsberger S, Rank E, Larson MG (2022) The finite cell method with least squares stabilized Nitsche boundary conditions. *Comput Methods Appl Mech Eng* 393:114792
  50. Leidinger LF, Breitenberger M, Bauer AM, Hartmann S, Wüchner R, Bletzinger K-U, Duddeck F, Song L (2019) Explicit dynamic isogeometric b-rep analysis of penalty-coupled trimmed NURBS shells. *Comput Methods Appl Mech Eng* 351:891–927
  51. Leonetti L, Liguori FS, Magisano D, Kiendl J, Reali A, Garcea G (2020) A robust penalty coupling of non-matching isogeometric Kirchhoff–Love shell patches in large deformations. *Comput Methods Appl Mech Eng* 371:113289
  52. Liu L, Zhang Y, Hughes TJR, Scott MA, Sederberg TW (2014) Volumetric T-spline construction using Boolean operations. *Eng Comput* 30(4):425–439
  53. LS-DYNA. <http://www.lstc.com/products/ls-dyna>
  54. LS-PrePost. <http://www.lstc.com/products/ls-prepost>
  55. Magisano D, Leonetti L, Garcea G (2021) Isogeometric analysis of 3d beams for arbitrarily large rotations: locking-free and path-independent solution without displacement DOFs inside the patch. *Comput Methods Appl Mech Eng* 373:113437
  56. Nembrini J, Samberger S, Labelle G (2014) Parametric scripting for early design performance simulation. *Energy Build* 68:786–798
  57. Oesterle B, Ramm E, Bischoff M (2016) A shear deformable, rotation-free isogeometric shell formulation. *Comput Methods Appl Mech Eng* 307:235–255
  58. Oesterle B, Sachse R, Ramm E, Bischoff M (2017) Hierarchic isogeometric large rotation shell elements including linearized transverse shear parametrization. *Comput Methods Appl Mech Eng* 321:383–405
  59. Paolini A, Kollmannsberger S, Rank E, Horger T, Wohlmuth B (2019) A mortar formulation including viscoelastic layers for vibration analysis. *Comput Mech* 63(1):23–33
  60. Parvizian J, Düster A, Rank E (2007) Finite cell method. *Comput Mech* 41:121–133
  61. Pasch T, Leidinger LF, Apostolatos A, Wüchner R, Bletzinger KU, Duddeck F (2021) A priori penalty factor determination for (trimmed) NURBS-based shells with Dirichlet and coupling constraints in isogeometric analysis. *Comput Methods Appl Mech Eng* 377:113688
  62. Perdata A, Putanowicz R (2019) Tools and techniques for building models for isogeometric analysis. *Adv Eng Softw* 127:70–81
  63. Philipp B, Breitenberger M, D'Auria I, Wüchner R, Bletzinger K-U (2016) Integrated design and analysis of structural membranes using the isogeometric b-rep analysis. *Comput Methods Appl Mech Eng* 303:312–340
  64. Piegl LA, Tiller W (1997) The NURBS book, 2nd edn. Springer, New York
  65. Rank E, Ruess M, Kollmannsberger S, Schillinger D, Düster A (2012) Geometric modeling, isogeometric analysis and the finite cell method. *Comput Methods Appl Mech Eng* 249–252:104–115
  66. Rhinoceros. <http://www.rhino3d.com>
  67. Robinson TT, Armstrong CG, Chua HS (2013) Determining the parametric effectiveness of a CAD model. *Eng Comput* 29(1):111–126
  68. Rorris L, Nagy A, Hartmann S, Chalkidis I, Vafeidis A (2019) The ANSA/LS-DYNA approach for isogeometric analysis simulations. In: 8th BEFORE REALITY conference
  69. Sanches R, Bornemann P, Cirak F (2011) Immersed b-spline (i-spline) finite element method for geometrically complex domains. *Comput Methods Appl Mech Eng* 200(13):1432–1445
  70. Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR (2011) Isogeometric finite element data structures based on Bézier extraction of T-splines. *Int J Numer Methods Eng* 88(2):126–156
  71. Scott MA, Li X, Sederberg TW, Hughes TJR (2012) Local refinement of analysis-suitable T-splines. *Comput Methods Appl Mech Eng* 213–216:206–222
  72. Stahl A, Kvamsdal T, Schellewald C (2017) Post-processing and visualization techniques for isogeometric analysis results. *Comput Methods Appl Mech Eng* 316:880–943. Special Issue on Isogeometric Analysis: Progress and Challenges
  73. Stroud I (2016) Boundary representation modelling techniques, 1st edn. Springer, Berlin
  74. Teschemacher T, Bauer AM, Oberbichler T, Breitenberger M, Rossi R, Wüchner R, Bletzinger K-U (2018) Realization of CAD-integrated shell simulation based on isogeometric B-Rep analysis. *Adv Model Simul Eng Sci* 5(1):1–54
  75. Teschemacher T, Bauer AM, Aristio R, Messmer M, Bletzinger K-U. Cocodrilo. <https://github.com/CocodriloCAD/Cocodrilo>
  76. Teschemacher T, Celigueta Jordana MÁ, Casas González G, Wüchner R, Bletzinger K-U (2019) Particle-structure interaction using cad-based boundary descriptions and isogeometric B-REP analysis (IBRA). In: The VI international conference on particle-based methods (PARTICLES 2019)

77. U.N.B. of Standards (1980) Initial graphics exchange specification (IGES), nbsir 80-1978, digital representation for communication of product definition data
78. Vassberg J, Dehaan M, Rivers M, Wahls R (2012) Development of a common research model for applied CFD validation studies. In: 26th AIAA applied aerodynamics conference
79. Wassermann B, Kollmannsberger S, Bog T, Rank E (2017) From geometric design to numerical analysis: a direct approach using the finite cell method on constructive solid geometry. *Comput Math Appl* 74(7):1703–1726
80. Wassermann B, Kollmannsberger S, Yin S, Kudela L, Rank E (2019) Integrating cad and numerical analysis: ‘dirty geometry’ handling using the finite cell method. *Comput Methods Appl Mech Eng* 351:808–835
81. Weeger O, Wever U, Simeon B (2013) Isogeometric analysis of nonlinear Euler–Bernoulli beam vibrations. *Nonlinear Dyn* 72:01
82. Wei X, Li X, Qian K, Hughes TJ, Zhang YJ, Casquero H (2022) Analysis-suitable unstructured T-splines: multiple extraordinary points per face. *Comput Methods Appl Mech Eng* 391:114494
83. Wei X, Zhang Y, Hughes TJ, Scott MA (2015) Truncated hierarchical Catmull–Clark subdivision with local refinement. *Comput Methods Appl Mech Eng* 291:1–20
84. Zander N, Bog T, Kollmannsberger S, Schillinger D, Rank E (2015) Multi-level hp-adaptivity: high-order mesh adaptivity without the difficulties of constraining hanging nodes. *Comput Mech* 55(3):499–517
85. Zhang Y, Wang W, Hughes TJ (2012) Solid T-spline construction from boundary representations for genus-zero geometry. *Comput Methods Appl Mech Eng* 249–252:185–197. *Higher Order Finite Element and Isogeometric Methods*

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.