

# An interactive geometry modeling and parametric design platform for isogeometric analysis

Ming-Chen Hsu<sup>a,\*</sup>, Chenglong Wang<sup>a</sup>, Austin J. Herrema<sup>a</sup>, Dominik Schillinger<sup>b</sup>,  
Anindya Ghoshal<sup>c</sup>, Yuri Bazilevs<sup>d</sup>

<sup>a</sup>*Department of Mechanical Engineering, Iowa State University, 2025 Black Engineering, Ames, IA 50011, USA*

<sup>b</sup>*Department of Civil Engineering, University of Minnesota, 500 Pillsbury Drive S.E., Minneapolis, MN 55455, USA*

<sup>c</sup>*Vehicle Technology Directorate, United States Army Research Laboratory, Aberdeen Proving Ground, MD 21005, USA*

<sup>d</sup>*Department of Structural Engineering, University of California, San Diego, 9500 Gilman Drive, Mail Code 0085, La Jolla, CA 92093, USA*

---

## Abstract

In this paper an interactive parametric design-through-analysis platform is proposed to help design engineers and analysts make more effective use of Isogeometric Analysis (IGA) to improve their product design and performance. We develop several Rhinoceros (Rhino) plug-ins to take input design parameters through a user-friendly interface, generate appropriate surface and/or volumetric models, perform mechanical analysis, and visualize the solution fields, all within the same Computer-Aided Design (CAD) program. As part of this effort we propose and implement graphical generative algorithms for IGA model creation and visualization based on Grasshopper, a visual programming interface to Rhino. The developed platform is demonstrated on two structural mechanics examples—an actual wind turbine blade and a model of an integrally bladed rotor (IBR). In the latter example we demonstrate how the Rhino functionality may be utilized to create conforming volumetric models for IGA.

*Keywords:* Isogeometric analysis; NURBS and T-splines; Parametric design; Rhino and Grasshopper; Visual programming; Graphical generative algorithms

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>IGA design-through-analysis platform</b>	<b>5</b>
2.1	Platform structure . . . . .	5
2.2	Visual programming for IGA modeling: Grasshopper . . . . .	7

---

\*Corresponding author

Email address: [jmchsu@iastate.edu](mailto:jmchsu@iastate.edu) (Ming-Chen Hsu)

2.3	Parametric design and geometry modeling . . . . .	7
2.4	Visualization of NURBS and T-spline analysis results . . . . .	11
2.4.1	NURBS and T-spline surface construction . . . . .	12
2.4.2	Visualization mesh and IGA results . . . . .	12
2.4.3	Wireframe and color contour surface . . . . .	14
<b>3</b>	<b>Example 1: Wind turbine blade</b>	<b>15</b>
3.1	T-spline model . . . . .	15
3.2	Setting material properties, loads, and boundary conditions . . . . .	17
3.3	Simulation results . . . . .	18
3.4	Visualization of IGA results . . . . .	18
3.5	Parametric design modification . . . . .	18
<b>4</b>	<b>Example 2: Integrally bladed rotor</b>	<b>19</b>
4.1	Modeling of the IBR surface geometry . . . . .	19
4.2	Modeling of the IBR solid geometry . . . . .	23
4.3	Structural vibration analysis of an IBR . . . . .	26
<b>5</b>	<b>Conclusion</b>	<b>26</b>

## 1. Introduction

In recent years, the development of Isogeometric Analysis (IGA) [1, 2] has paved a path towards a tighter integration of engineering design and computational analysis. The core idea of IGA is to use the same basis functions for the representation of geometry in Computer-Aided Design (CAD) and the approximation of solution fields in Finite Element Analysis (FEA). As a result, a single geometric model can be utilized directly as the analysis model. This approach bypasses the labor-intensive mesh generation process required for analysis, and has great potential to significantly improve the efficiency of design-through-analysis cycle.

Several computational geometry techniques have been used in IGA. Non-uniform rational B-splines (NURBS) [3–5] are the industry standard for geometry modeling and are most widely used in engineering design. NURBS are convenient for free-form surface modeling and can represent all conic sections exactly. There are many efficient and numerically stable algorithms [4], as well as commercial software packages (e.g., Rhinoceros—typically abbreviated Rhino [6]), that can generate and manipulate NURBS objects. Besides the geometric flexibility, NURBS also have excellent approximation properties [7, 8] necessary for accurate analysis. Therefore, NURBS, a CAD standard representing many years of development, were a natural starting point for IGA [1]. T-splines [9, 10] were introduced in the CAD community as a generalization and extension of

NURBS allowing for local refinement and coarsening, and representation of geometry of arbitrary topological genus. T-splines have been applied successfully in the context of IGA [11–16] and have been further improved to meet the demands of analysis [17–21]. Recent attempts to construct trivariate solid T-splines can be found in [22–29].

IGA has been successfully employed in many areas of engineering and sciences, such as fluid mechanics and turbulence [30–39], solid and structural mechanics [40–56], fluid–structure interaction [13, 57–64], phase-field modeling [65–69], complex fluids [70, 71], Lagrangian shock hydrodynamics [72, 73], contact mechanics [16, 74–81], and optimization [82–85]. In most cases, IGA showed a clear advantage over standard low-order finite elements in terms of solution per-degree-of-freedom accuracy. This enhanced accuracy is in part attributable to the higher-order smoothness of the basis functions employed. Recent research directions in IGA include collocation [86–93], quadrature rules [94–97], trimmed geometries and patch coupling [98–103], analysis-suitable trivariate model [26–29, 104–108], and standardized file formats for data exchange between geometry modeling and computational analysis software [109–113]. For more details of mathematical developments, basis function research, geometry modeling, model quality assessment, and early applications, the reader is referred to [2] and references therein.

With the success of IGA, one research direction has been the realization of IGA design-through-analysis workflow. The concept of IGA design-through-analysis framework was proposed in [14] based on hierarchical refinement of NURBS, T-splines, and Finite Cell Method [114–118]. The framework is suitable for solving three-dimensional solid-geometry problems by immersing the surface model into a background fictitious domain, a methodology recently termed *Immersogeometric Analysis* [64]. References [119, 120] present concrete instantiations of the design-through-analysis workflow implementation in actual CAD software. Reference [119] laid out many of the core ideas for a design-through-analysis workflow for nonlinear shell structures, and, much like the present paper, made use of Rhino as the development platform. Reference [120] presented an Analysis in Computer Aided Design (AiCAD) concept using NURBS-based B-Rep models and isogeometric B-Rep analysis (IBRA) method for nonlinear shell analysis. This more recent work included enhancements such as the ability to perform analysis in the presence of trimmed surfaces and the use of Nitsche’s method for patch coupling. AiCAD has been implemented in two CAD software packages, Rhino and Siemens NX [121].

Despite the progress achieved in the last few years, several challenges remain in effectively using IGA to improve the engineering design process. Perhaps the biggest challenge is the rapid, (semi-)automatic construction of geometric models suitable for analysis. However, the difficulties of constructing designs and the corresponding geometric and analysis models are often overlooked in the engineering literature. It is often a time-consuming and challenging process to construct a baseline IGA model, and to ensure the model has the desired features such as good parameteriza-

tion, sufficient mesh density in the regions of interest, and, most importantly, analysis suitability. In many cases intimate familiarity with CAD technology and advanced programming skills are necessary to successfully build such models. Design engineers, while good professional in their application areas, may not have such skills. Furthermore, in many cases, engineers are only interested in a handful of design parameters and how they affect the product performance. As a result, to help design engineers and analysts make more effective use of IGA, we propose in this work to develop an interactive IGA design-through-analysis platform based on the idea of parametric design and geometry modeling.

The proposed interactive geometry modeling and parametric design platform can streamline the engineering design process by hiding the complex CAD functions in the background through generative algorithms, and letting the user control the design through key design parameters. Since the design concept is integrated with analysis, the design parameters can include not only the geometry parameters, but also quantities such as material properties, loads, and boundary conditions. In this work, the concept of parametric design and geometry modeling is realized through a visual programming interface called Grasshopper [122], which is widely used by designers focusing on exploring new shapes using generative algorithms in Rhino. One advantage of using Grasshopper for parametric design and geometry modeling is that, during the CAD model generation, one can design the generative algorithm to ensure that the resulting IGA model is analysis-suitable.

The concept of *parametric modeling* is central to design in many fields of engineering and beyond (e.g., architecture [123]). Currently parametric modeling is used in conjunction with solid geometry modeling that employs geometric primitives and Boolean operations (e.g., SOLIDWORKS [124]). The use of parametric modeling with modern Spline technology like NURBS or T-splines is not common, and presents a novel research direction in IGA. The proposed work builds on the concept of parametric modeling and provides a fairly general and convenient approach for creating parametric designs, which make use of NURBS and T-spline geometry description, using the visual programming concept. The proposed approach is applicable to a large class of geometries, including surface and volumetric descriptions.

Another novel and unique aspect of this work is the development of the IGA visualization tool directly within Rhino CAD software. Good-quality visualization of the IGA simulation results is not a trivial matter. In many cases this is done by interpolating the IGA solution with low-order finite-element functions, and outputting the results using a standard finite-element data structure for visualization using existing software. However, in this work, we propose a procedure and develop a Rhino plug-in that can be used to visualize NURBS and T-spline analysis results directly in Rhino.

Finally, in the context of template-based modeling [36, 58, 104], we also demonstrate how Rhino functionality may be used to help construct volumetric (trivariate) NURBS models.

The paper is outlined as follows. In Section 2 we describe the salient features of the parametric IGA design-through-analysis platform. In Sections 3 and 4 we present two examples that further describe the inner workings of the proposed platform. In Section 3 we present an example of a wind turbine blade, which is treated as a composite shell structure. We demonstrate how the proposed tool can be used in a visual programming environment to: a. Create the blade geometry; b. Perform the desired mesh refinement; c. Select the regions of the geometry for the imposition of boundary conditions, loads, and assignment of material parameters; d. Perform shell analysis; and e. Visualize the results. In Section 4 we present an example involving an Integrally Blended Rotor (IBR), which is an important mechanical component in modern gas-turbine engines. Here the added complexity with respect to the previous example is the need to model volumetric geometry, and we demonstrate how the proposed tool may be employed to handle such cases. In Section 5 we draw conclusions.

## 2. IGA design-through-analysis platform

In this work we develop an IGA design-through-analysis platform that offers a user-friendly interface for the entire design process, including geometry modeling, assignment of model material parameters, loads, and boundary conditions, computational analysis, and post-processing. We have near-term plans to integrate parametric design as described in [125–127] into the platform in order to enable repetitive design, quality improvement of geometric models [128, 129], and rapid analysis of designs linked by model parameters. Although we target users that have elementary knowledge of geometry modeling, mechanical and structural design, mechanics of solids, structures, and fluids, and numerical methods, the platform can also be used by experts in these areas.

### 2.1. Platform structure

The proposed platform provides a closed-loop design method for engineering applications as depicted in Figure 1. Once a basic model has been designed through CAD software, simulation may be performed directly using this model after distributing material parameters and specifying boundary and load conditions on model surfaces. After inspecting the solution, the designer can then make a judgment where to improve the current design. Because the original geometry is modeled parametrically, changing the design according to the analysis results would consist of

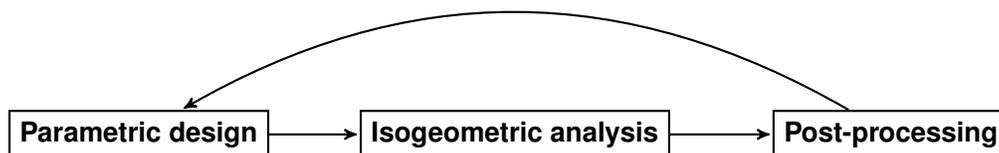


Figure 1: Isogeometric design-through-analysis platform structure.

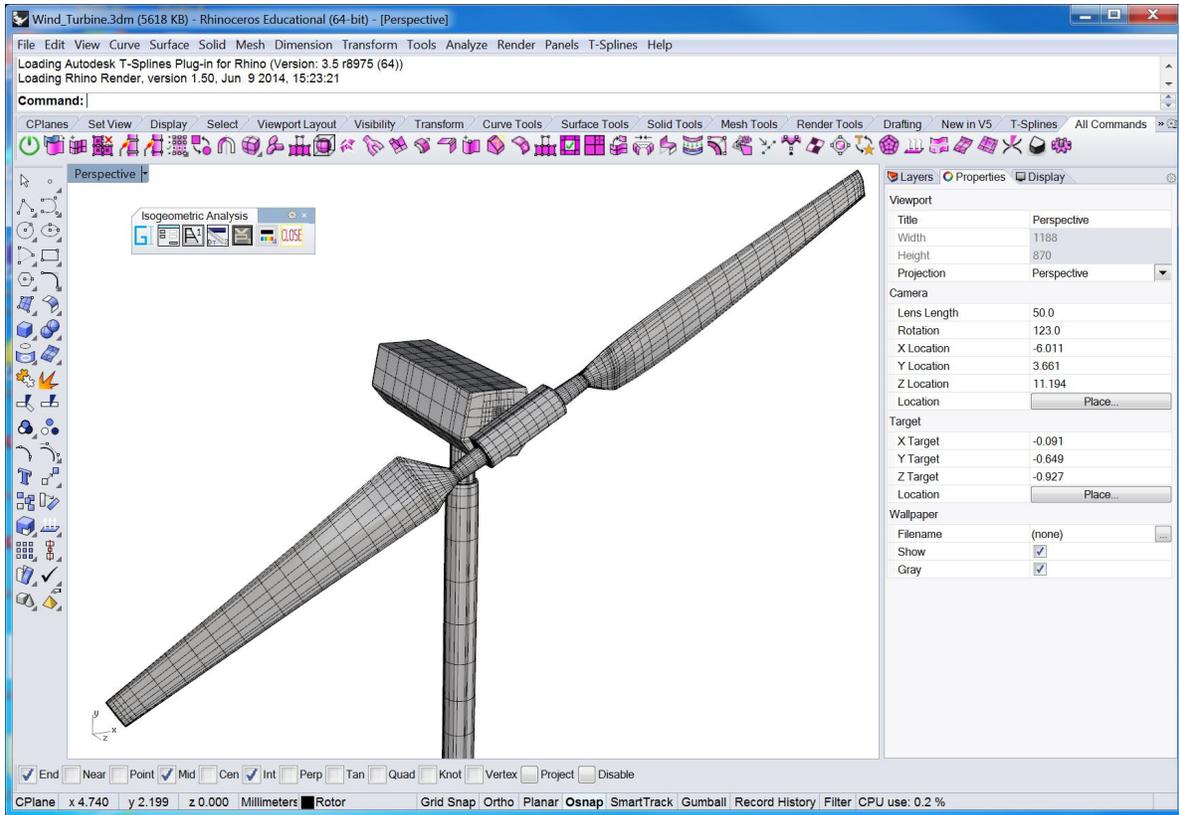


Figure 2: Rhino CAD modeling software with the proposed plug-ins integrated.

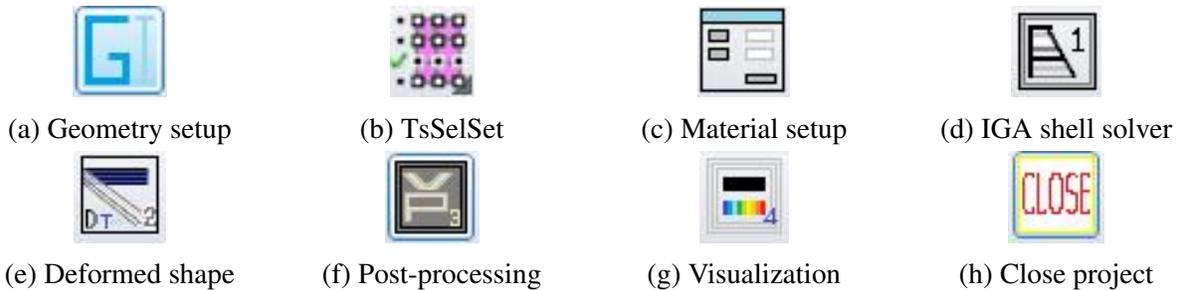


Figure 3: Platform plug-in options.

simply adjusting input parameters. As a result, within this closed-loop design process, the user could conceivably create and optimize designs within a shorter timeframe using a single platform.

In order to fulfill the requirements for our platform outlined above, we base our developments on Rhino [6] CAD software. Rhino gives designers a variety of functions that are required to build complex, multi-patch NURBS surfaces [4]. Recently, additional functionality was added in Rhino to create and manipulate T-spline surfaces [9, 130], which is an important enhancement allowing one to move away from a fairly restrictive NURBS-patch-based geometry design to a completely unstructured, watertight surface definition while respecting all the constraints imposed by analysis [18, 20]. Rhino also features an enhanced *graphic programming design tool* for para-

metric design called Grasshopper [122], and utilizes free and open-source software development kits (SDK) [131] for plug-in development. Furthermore, Rhino is relatively transparent as compared to other CAD software in that it provides the user with the ability to interact with the system through the so-called “plug-in” commands. All of these features are well aligned with our goals, and we exploit them in the design of our IGA-based design-through-analysis platform.

Figure 2 shows a snapshot of the Rhino CAD modeling software interface, with the proposed platform plug-ins integrated. The figure shows a full wind-turbine model represented using T-spline surfaces. Figure 3 shows the developed plug-in commands (or “buttons”), including parametric geometry design, assignment of material parameters, boundary and load conditions imposition, interface to an IGA shell solver, and post-processing and visualization of the analysis result. The details of the plug-in commands and their use will be elucidated in the applications sections of the present paper.

**Remark 1.** Features such as the ability to manipulate complex surfaces and create plug-in commands in Rhino for the implementation of the IGA-based modeling framework were exploited in the earlier work (see [119, 120, 132]). In this paper we focus mainly on the parametric design, and the visual programming interface Grasshopper that enables such design.

## *2.2. Visual programming for IGA modeling: Grasshopper*

As the visual programming interface, Grasshopper moves away from the traditional paradigm of writing a text file with program instructions, which is then fed to an off-the-shelf compiler to produce an executable file. This approach is bug-prone and often requires significant programming education and expertise. In contrast, using Grasshopper, the program is written in terms of “components” with pre-defined functionality, and “wire connections” between the components that serve as conduits of input and output data. As a result, using an intuitive arrangement of components and connections, one can rapidly generate an analysis model, establish parametric control, and link the model to the desired solver and visualization modalities. In the case when new functionality is needed, a traditional programming approach may be employed to create new components, which are then added to the library of the existing ones, and may be flexibly used by the designer. An example of Grasshopper parametric geometry design of a wind turbine blade is shown in Figure 4.

## *2.3. Parametric design and geometry modeling*

During the design cycle, geometric models are often constructed through similar design algorithms. We developed an interactive parametric geometry modeling plug-in that enables rapid construction of analysis-suitable, multi-patch NURBS models using Grasshopper [122]. The plug-in streamlines the engineering design process by hiding the complex CAD functions in the background through generative algorithms, and letting the user control the design through key design

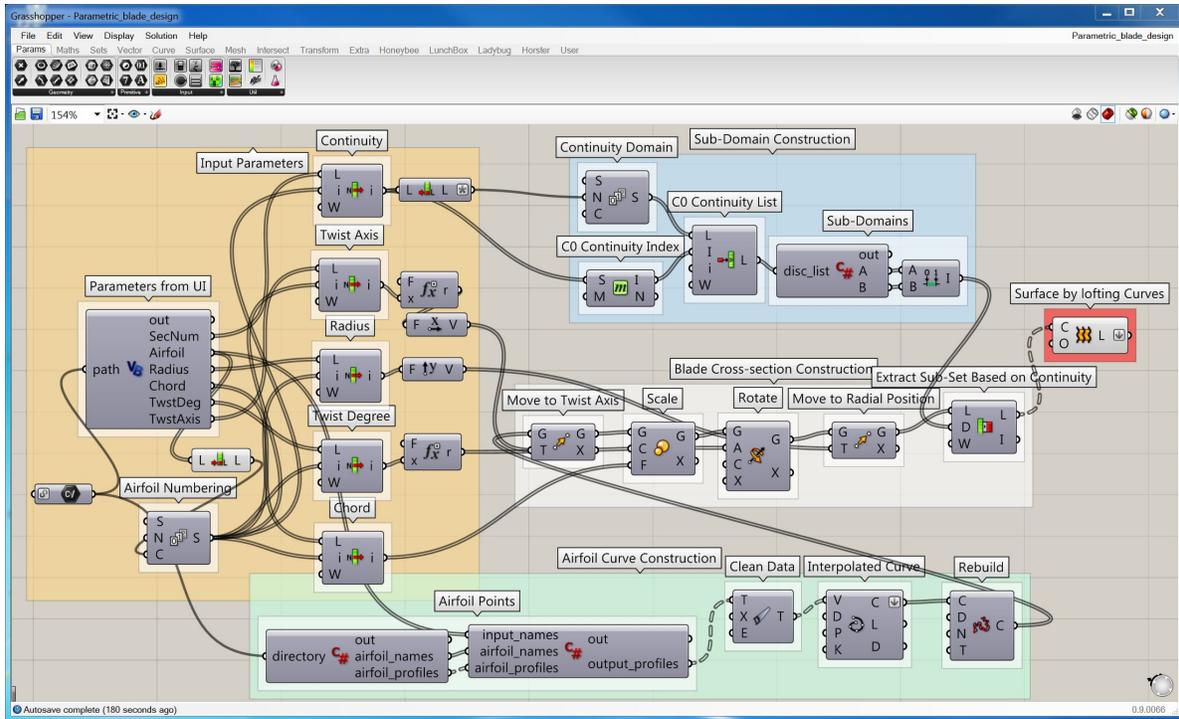
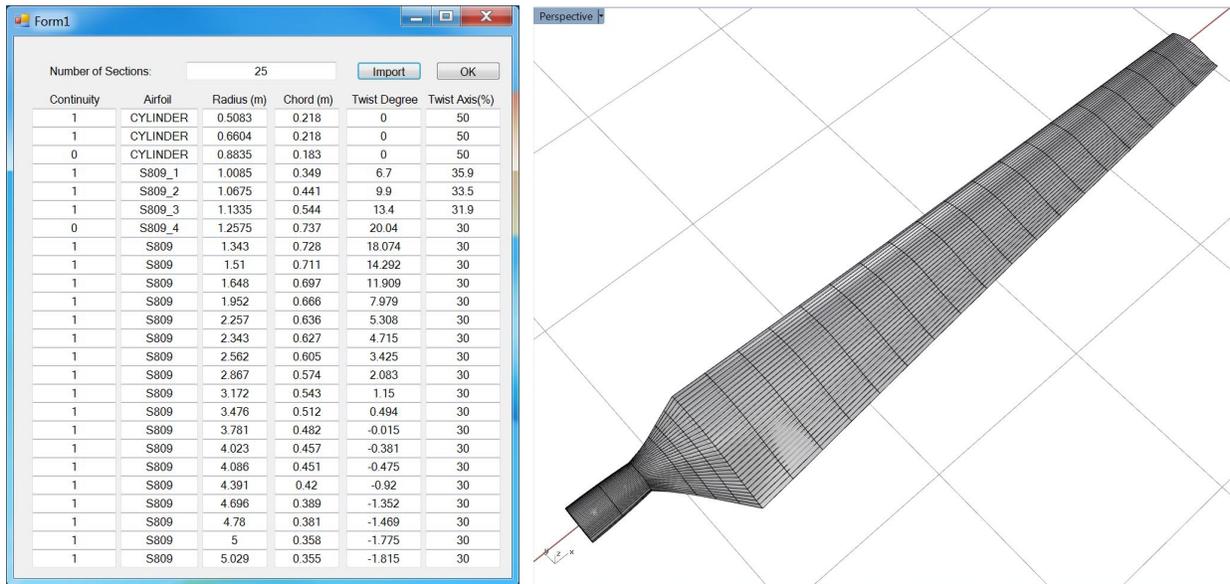


Figure 4: Grasshopper generative algorithm for parametric geometry design of wind turbine blade.



(a) User interface for parametric design.

(b) NREL Phase VI wind turbine blade.

Figure 5: NREL Phase VI wind turbine blade parametric design and geometry modeling.

parameters. An added benefit of using this approach for parametric design and geometry modeling is that, during the CAD model generation, one can ensure that the resulting IGA model is analysis-suitable.

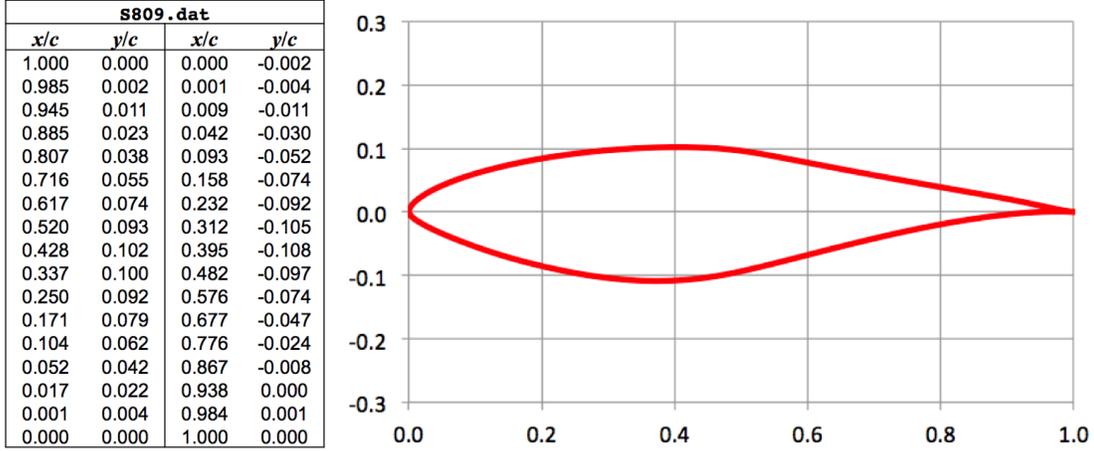
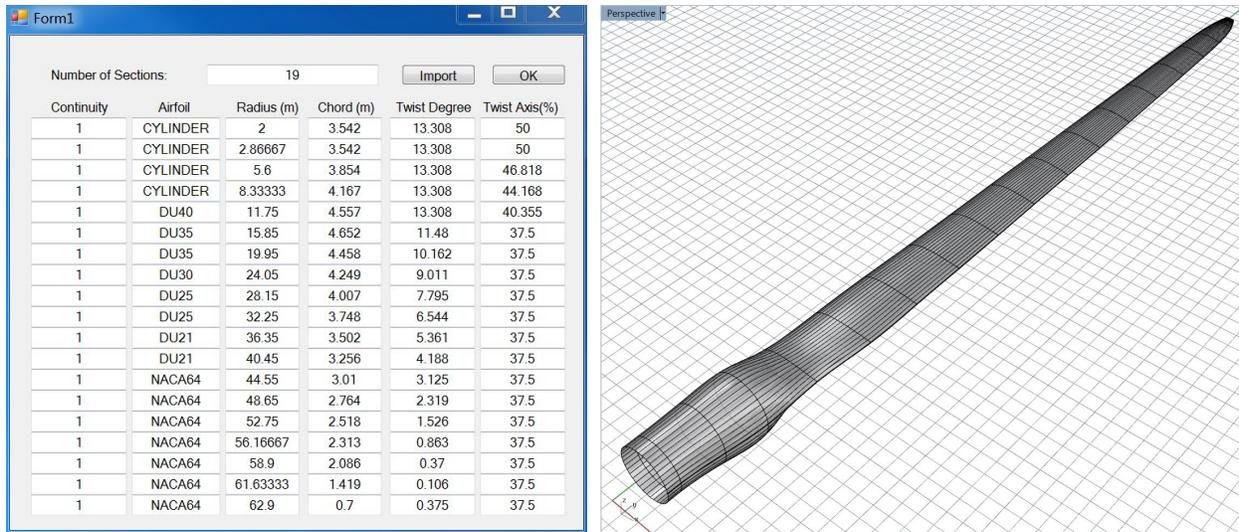


Figure 6: S809 airfoil data and profile [134].

Figure 4 shows an example of using the Grasshopper generative algorithm for parametric geometry design of a wind turbine blade. The leftmost group of components named “Input Parameters” represents operations on input data such as surface continuity, airfoil type, radial position, twist angle and axis, and chord length. The group includes the in-house developed VBScript (Visual Basic Scripting Edition) component for reading these input parameters from a user interface. The user interface is shown in Figure 5(a) and is developed using C# RhinoCommon Plug-in SDK [133]. The group of components named “Airfoil Curve Construction” imports unit-chord-length airfoil data given by the users and constructs smooth NURBS curves interpolating through each set of airfoil data points. The “rebuild” function is then used on these curves to make sure that all the NURBS airfoil profiles have the same number of control points and knot vectors. This operation ensures that the NURBS surface generated by skinning (or lofting) along this series of profile curves will have the desired parameterization.

Based on the input parameters corresponding to blade cross sections, each airfoil profile is relocated such that the twist axis is aligned to the origin, scaled by the chord length, and rotated according to the twist degree. The modified airfoil profiles are then moved to their corresponding radial positions along the twist axis, which is also the blade-pitched axis. This procedure is performed by the group of components named “Blade Cross-section Construction” in Figure 4. Due to the inherent discontinuity of sharp transition between different blade design zones, the input data are separated into different subdomains using the group of components named “Sub-Domain Construction”. After all subsets of blade cross-section curves are prepared, individual NURBS surfaces are generated by skinning (or lofting) along the curves within each subset. This is done by the rightmost component shown in Figure 4. The multi-patch NURBS surface generated through this procedure is conforming between different patches.

**Remark 2.** We note that a geometric model may be constructed using different procedures and



(a) User input design parameters.

(b) NREL 5 MW wind turbine blade.

Figure 7: NREL 5 MW offshore wind turbine blade parametric geometry design.

algorithms depending on the designer’s preferences and the level of familiarity with the tool. The example shown in Figure 4 is one of the many ways of achieving the same goal.

The proposed concept is applied to the parametric geometry design of an NREL Phase VI wind turbine blade [135, 136] that requires a considerable number of parametric inputs, including the geometric continuity of each cross section, airfoil type, radial airfoil location, and chord length. This NREL wind turbine blade has 25 airfoil cross sections. It gradually changes from a cylindrical cross section at the hub center to an S809 airfoil [134] cross section along the blade to the blade tip. The main input parameters are shown in Figure 5(a). The S809 airfoil data points shown in Figure 6 can be stored in a text file and imported via the interface. After entering all the information, a multi-patch NURBS surface of the wind turbine blade is generated as shown in Figure 5(b).

To demonstrate the applicability of our parametric design interface, we construct another complex 5 MW offshore wind turbine blade [36, 137] the same way we constructed the previous example; namely, through our user interface, accessed by selecting the command “Geometry User Interface” shown in Figure 3(a). The input design parameters and constructed NURBS blade geometry are shown in Figure 7.

**Remark 3.** Since the IGA design concept is integrated with analysis, the design parameters can include not only the geometry parameters, but also quantities such as material properties, loads, and boundary conditions. This is demonstrated through the wind turbine blade example in Section 3.

**Remark 4.** The surface geometry may be converted to a T-spline representation (see, e.g., Figure 2) using the Autodesk T-Splines Plug-in for Rhino [130] to have better modeling features, such as local refinement and coarsening.

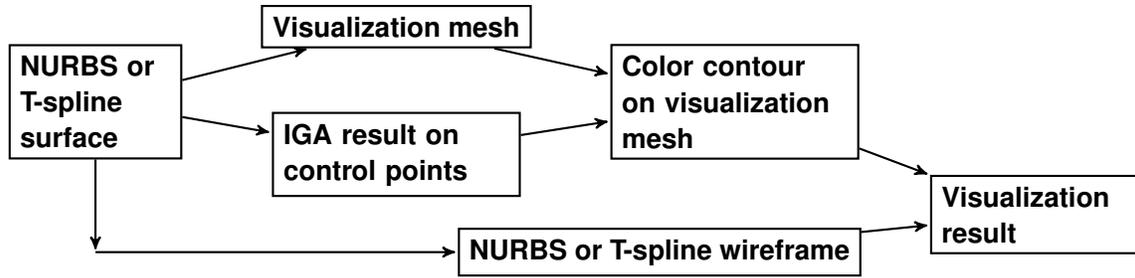


Figure 8: Visualization procedure for NURBS or T-spline IGA result.

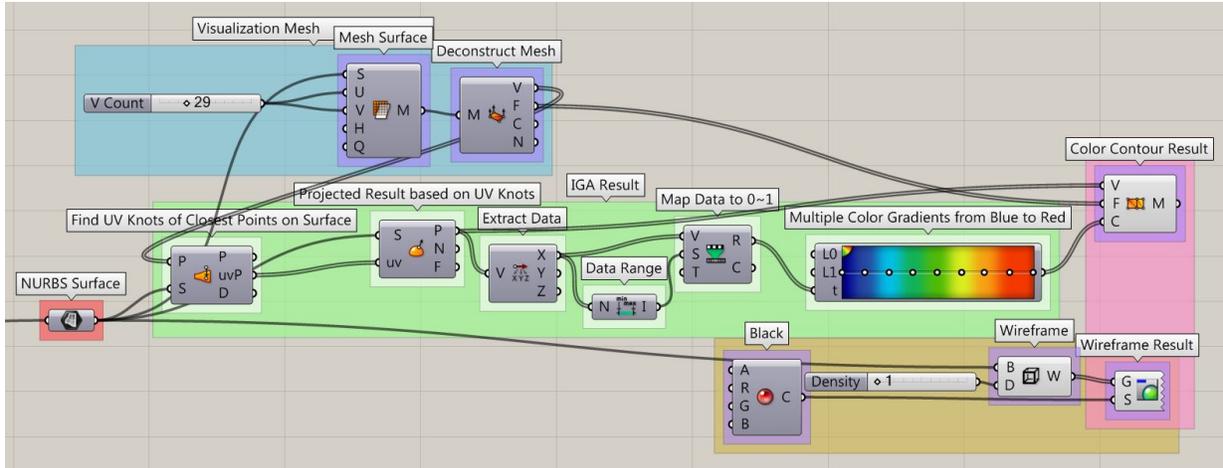


Figure 9: Grasshopper components for visualization of NURBS analysis result.

**Remark 5.** Quite often a volumetric representation of the geometry is needed for analysis, as in the case of an IBR. We discuss possible extensions of the current geometry modeling platform to handle volumes in the application section describing the IBR model.

#### 2.4. Visualization of NURBS and T-spline analysis results

After solving the IGA problem, the control variables (or degrees of freedom) for the solution fields (e.g., displacement, velocity, temperature, etc.) are obtained. These control variables are defined on the control points, which are typically not located on the physical geometry. When coupled with basis functions these give continuous solution fields on the geometry. Visualization of the IGA results is an integral part of the design-through-analysis framework, and presents some challenges that we address in this section. Here we focus on visualization of the solution fields defined in terms of NURBS and T-spline functions performed within the same IGA design-through-analysis platform.

Figure 8 shows a conceptual diagram of the visualization procedure for NURBS and T-spline IGA results. A visualization mesh is constructed directly from the NURBS or T-spline surface in order to visualize color contours of the solution fields on the geometry using the control variables, control points and basis functions information. The visualization mesh points require the solution

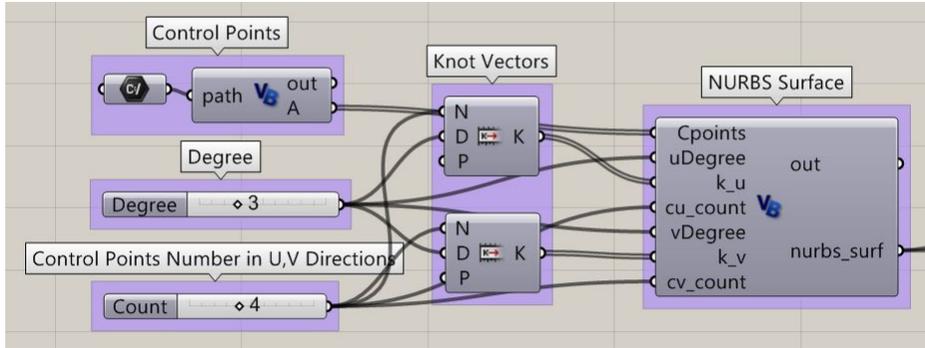


Figure 10: NURBS surface construction in Grasshopper using parametric inputs.

values, which can be evaluated at their closest points on the NURBS or T-spline surface. The color contours of the visualization mesh is then overlapped with the wireframe<sup>1</sup> extracted from the NURBS or T-spline surface. The combination of these two inputs provides a novel way of visualizing the IGA results directly within the CAD software.

An implementation of this idea as a Rhino plug-in is shown in Figure 9, which is a Grasshopper generative algorithm for visualizing IGA results directly in Rhino. More details about the plug-in are given in what follows.

#### 2.4.1. NURBS and T-spline surface construction

Figure 10 shows an example of constructing a NURBS surface in Grasshopper using parametric inputs including control point information, degree, and knot vector. The example model has 16 control points, four in each parametric direction. The polynomial degrees are cubic in both directions, leading to a one-element mesh. The VBScript component includes all the necessary functions to build a NURBS surface based on the user inputs. The constructed NURBS surface is shown in Figure 11, where the control points and control polygon are also visualized. This surface is used for testing and demonstrating the color contour visualization concept and procedure. A T-spline surface can be created based on the NURBS surface, as shown in Figure 12, which has one element locally refined.

#### 2.4.2. Visualization mesh and IGA results

Currently, Rhino does not provide color contours directly visualized on NURBS or T-spline surface. In this work we propose to construct a visualization mesh from the NURBS or T-spline surface for the purpose of visualizing color contours on the physical geometry. This is shown as the top left group of components named “Visualization Mesh” in Figure 9. In this case, a NURBS sur-

<sup>1</sup>The wireframe is a set of crossing isoparametric curves, which are curves of constant parametric value mapped into physical space, on the NURBS or T-spline surface. By default in Rhino these parametric values used to generate the wireframe are selected at knot locations.

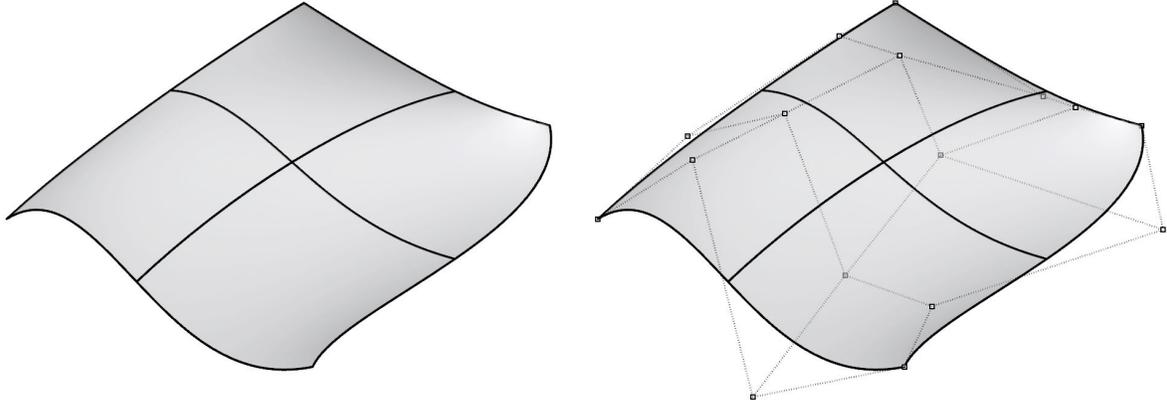


Figure 11: NURBS surface and control points.

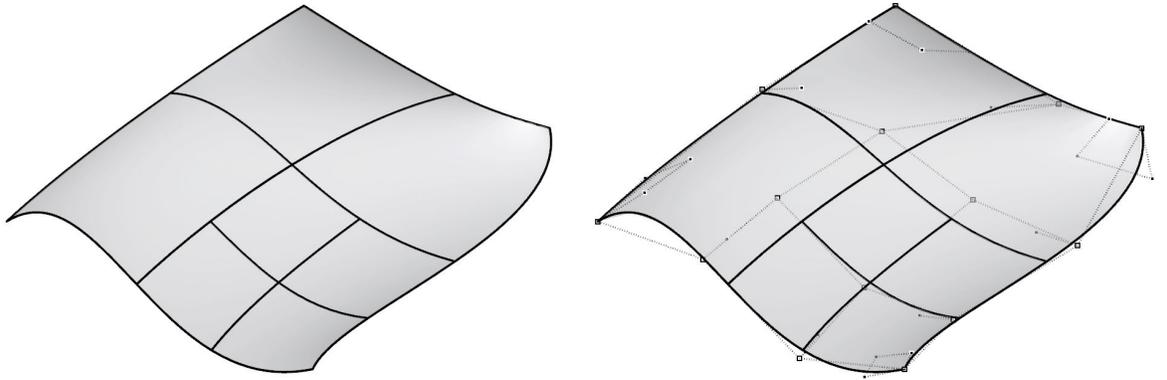


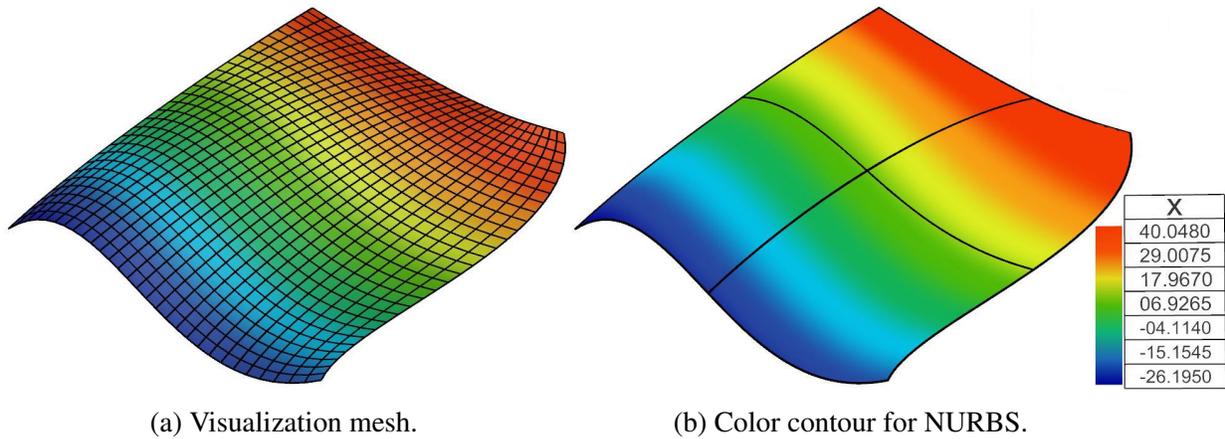
Figure 12: T-spline surface and control points.

face is evenly divided into several segments in both parametric coordinates. A denser mesh could be generated by increasing the number of segment if needed. Examples of the visualization mesh are shown in Figures 13(a) and 14(a). The T-spline visualization mesh is generated by first converting the T-spline surface back to NURBS patches, and then applying the same aforementioned procedure for NURBS surface.

**Remark 6.** It should be noted that the proposed method is not the only way to construct the visualization mesh. One can also, for example, directly mesh the NURBS or T-spline surface with triangles and/or quadrilaterals using the “mesh command” in Rhino to achieve the same purpose.

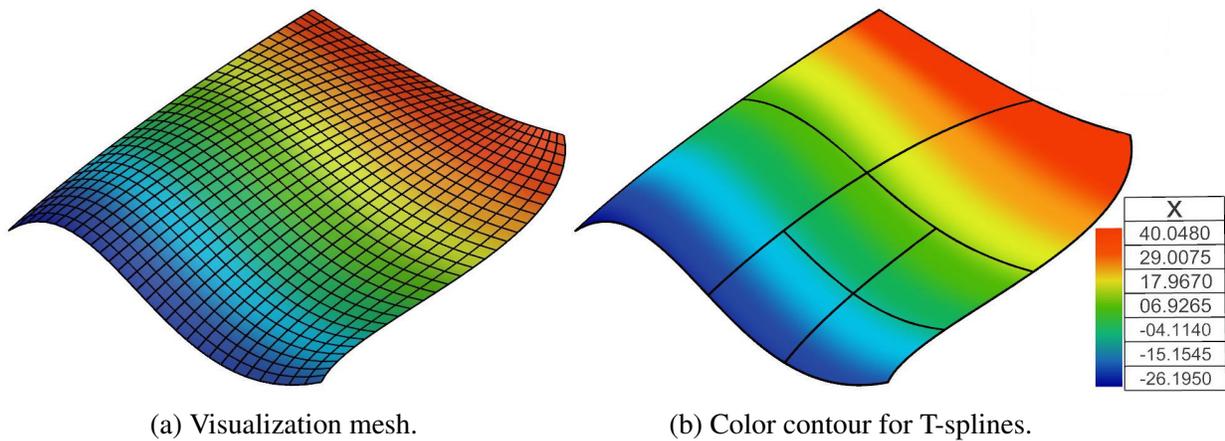
The control variables for the solution field of the IGA results are defined on the control points, which are usually not located on the geometry. These results need to be transferred to the visualization mesh points. In our implementation this is done by feeding the coordinates of the mesh points to a Grasshopper component<sup>2</sup> that finds the closest points on the NURBS or T-spline surface. The

<sup>2</sup>The component could be a Grasshopper built-in function (works for NURBS surfaces) or an in-house code [13] (works for both NURBS and T-spline surfaces).



(a) Visualization mesh. (b) Color contour for NURBS.

Figure 13: Visualization of color contour and wireframe for NURBS.



(a) Visualization mesh. (b) Color contour for T-splines.

Figure 14: Visualization of color contour and wireframe for T-splines.

component also returns the parametric coordinates of these closest points. Combining with control variables, control points and basis functions information, the solution values can then be evaluated and transferred back to the visualization mesh points. These data are extracted and mapped to a range between 0 and 1 that the linear color gradient is based on. The group of components named “IGA Result” in the middle of Figure 9 is used for this procedure.

### 2.4.3. Wireframe and color contour surface

In Figure 9, the group of components named “Wireframe” is for extracting wireframe from the NURBS or T-spline surface. The wireframe density is set to 1, which displays one isoparametric curve on each knot value. Increasing the wireframe density adds extract isoparametric curves on each knot interval. If the surface is a single knot-span surface (e.g., only one element), in the current version of Rhino, isoparametric curves (one in each parametric direction) will be displayed in the middle of the surface by default. (This explains the extra mesh lines in Figure 11.) The final visualization result is constructed in the rightmost group of components in Figure 9. The color

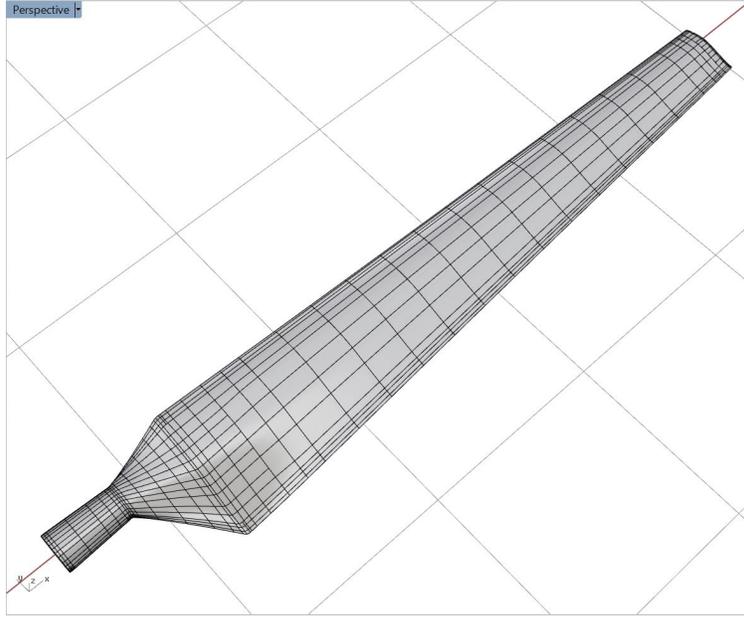


Figure 15: T-spline surface of the NREL Phase VI wind turbine blade.

contour surface from the visualization mesh is overlapped with the wireframe extracted from the original NURBS or T-spline surface as shown in Figures 13(b) and 14(b), respectively.

### 3. Example 1: Wind turbine blade

We apply the proposed IGA design-through-analysis platform to the modeling of wind turbine blades. In what follows, we make use of the rotation-free Kirchhoff–Love thin shell formulation [43, 45] combined with classical laminated plate theory [61, 138] to model the composite blade structural mechanics.

#### 3.1. T-spline model

The parametric design and geometry modeling of the NREL Phase VI wind turbine blade has been discussed in detail in Section 2.3 and a multi-patch NURBS surface was generated. It is converted to a single T-spline surface using the Autodesk T-Splines Plug-in for Rhino [112, 130] to have better modeling features, such as local refinement and coarsening. Figure 15 shows the T-spline surface of the wind turbine blade to which local refinement has already been added, and from which unwanted knots have been removed.

**Remark 7.** Geometric model may have far fewer degrees of freedom than what are required in analysis to get reliable solutions. It may be necessary to first refine the geometry before performing the analysis in order to obtain accurate simulation results.

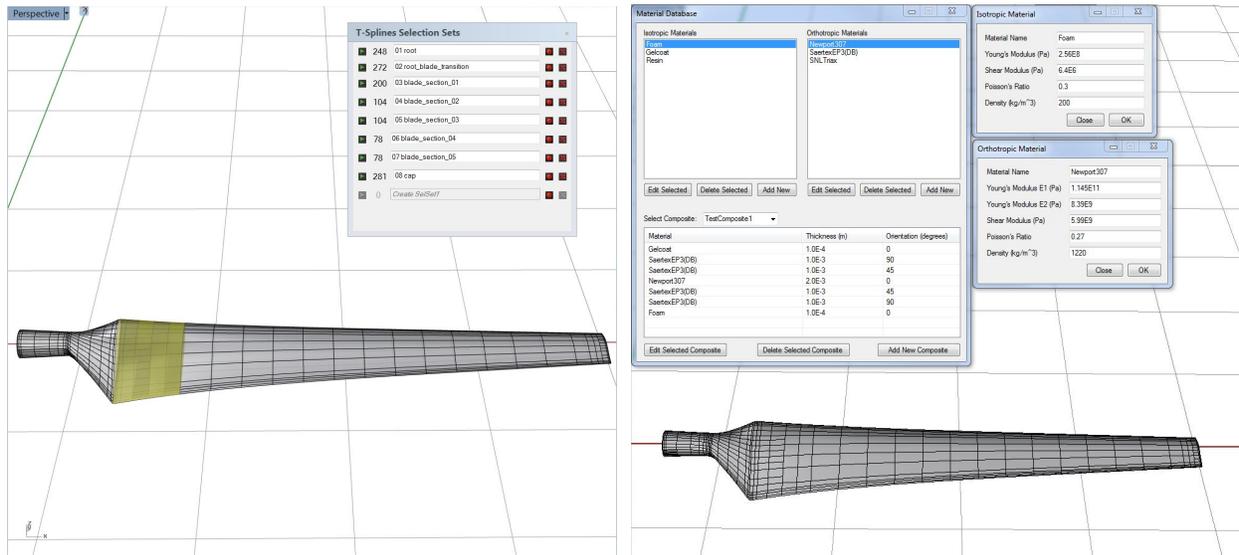


Figure 16: Left: Select and assign elements to different sets using “TsSelSet” command. Right: Composite layups can be defined and assigned to T-spline sets using the in-house developed user interface.

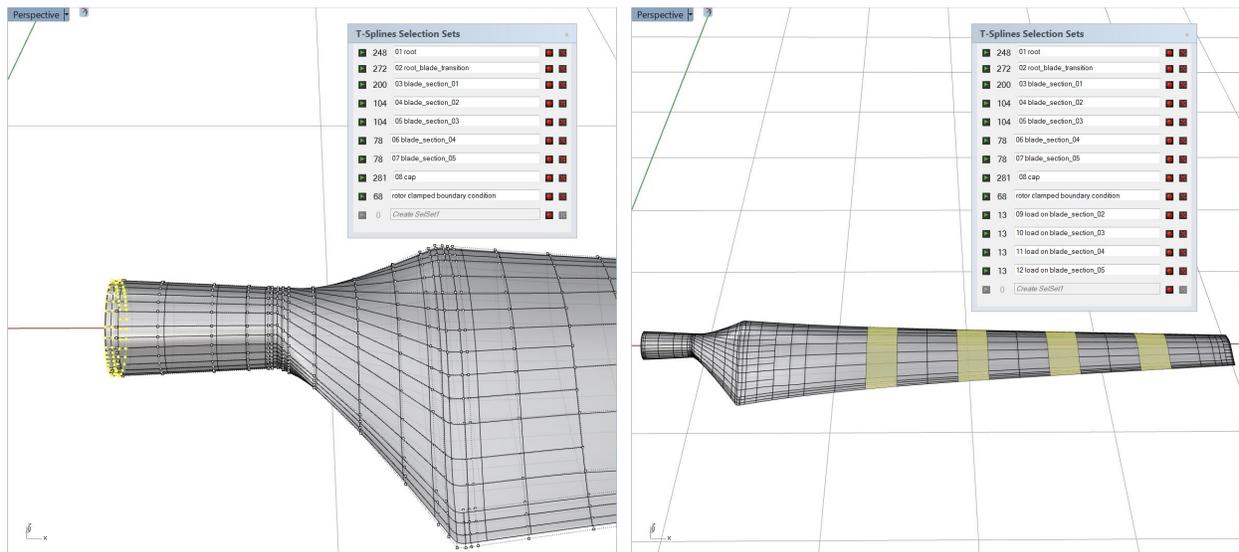


Figure 17: Left: Select control points to set clamped boundary condition. Right: Select elements to assign pressure loads on several upper-surface areas.

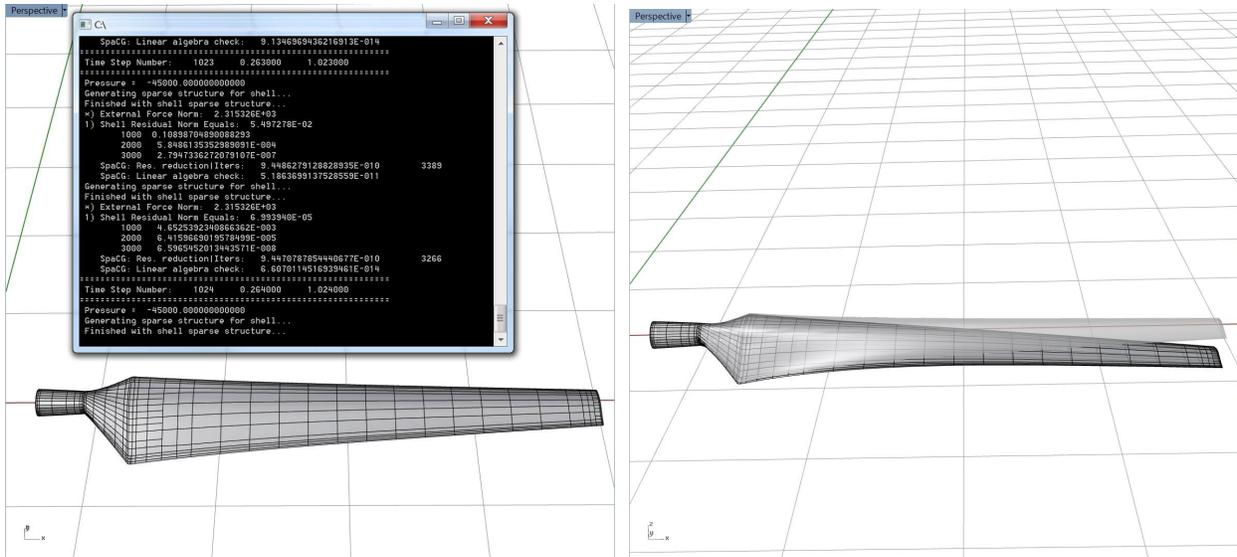


Figure 18: Left: Isogeometric shell analysis. Right: Deformed T-spline surface.

### 3.2. Setting material properties, loads, and boundary conditions

The next logical step towards analysis is to define the material properties, loads, and boundary conditions. The user interface for setting these properties and conditions depends on the selection and assignment of T-spline surface elements. By using T-spline “TsSelSet” [112] command in Figure 3(b), one can select groups of elements and define a different set to each group, as shown in Figure 16. After setting up these element sets, the composite material properties can be entered and assigned to each set by using the “Material setup” command in Figure 3(c). The user interface shown in Figure 16 allows the user to define composite layups consisting of any number of layers of user-defined isotropic or orthotropic materials. Each layer is assigned a material, thickness, and fiber orientation. This approach to composite definition is similar to that employed by Sandia National Laboratories’ Numerical Manufacturing And Design Tool (NuMAD) [139]. The resulting composite layup designs can then be assigned to specific T-spline surface elements. We use the classical laminated plate theory [61, 140] to homogenize the material through-thickness constitutive behavior for a given composite layup. For the sake of simplicity, the material we use in the following analysis is aluminum, which is isotropic and has Young’s modulus of 70 GPa and Poisson’s ratio of 0.35. The blade is assumed to have eight regions of constant thickness, which decreases from root to tip. Finally, we apply the clamped boundary condition at the root by selecting two rows of control points, and select four different pressure load areas on the pressure side of the blade surface, as shown in Figure 17.

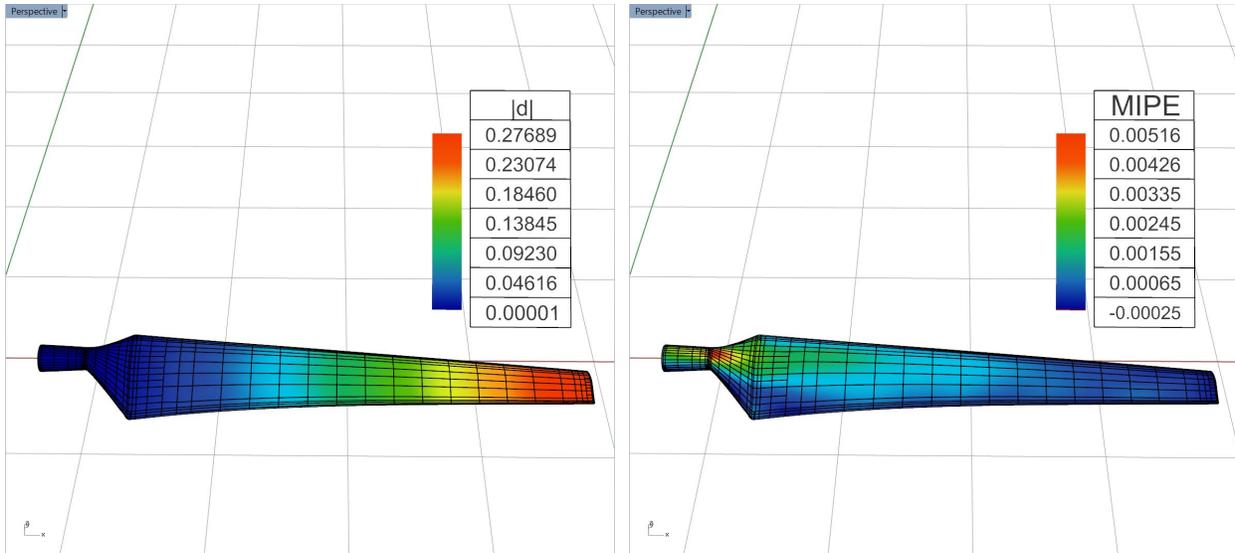


Figure 19: Displacement (left) and MIPE (right) contours of the isogeometric shell analysis result.

### 3.3. Simulation results

The pressure load of 45 kPa is applied on the selected zones of the blade surface as shown in Figure 17. The resultant force due to the pressure load is 2.315 kN. The blade is clamped at the root and, in addition, loaded by gravity. Dynamic simulation is employed with a time-step size of 0.001 s. The isogeometric Kirchhoff–Love shell solver is called by the “IGA Shell Solver” command shown in Figure 3(d). The analysis results are shown in Figure 18. The deformed T-spline surface is visualized by adding the displacement field to the control point coordinates. The command used to perform this function is “Deformed shape” shown in Figure 3(e).

### 3.4. Visualization of IGA results

To have a deeper understanding of the analysis results, one may perform post-processing of quantities of interest such as the maximum in-plane principal Green–Lagrange strain (MIPE) from shell displacement by using the “Post-processing” command shown in Figure 3(f). “Visualization” command shown in Figure 3(g) can then be executed to visualize the color contours of either displacement magnitude or MIPE on the blade surface for a chosen time step. The steady-state results are shown in Figure 19. The higher MIPE area is concentrated around the sharp transition from the cylindrical root to the airfoil cross sections. This analysis result could provide guidance for potential design improvement. Finally, the “Close project” command shown in Figure 3(h) is used to close all the plug-ins.

### 3.5. Parametric design modification

The design of the NREL Phase VI wind turbine blade can be easily modified to have a larger root by using the parametric design user interface as shown in Figures 20. Figure 21 shows an

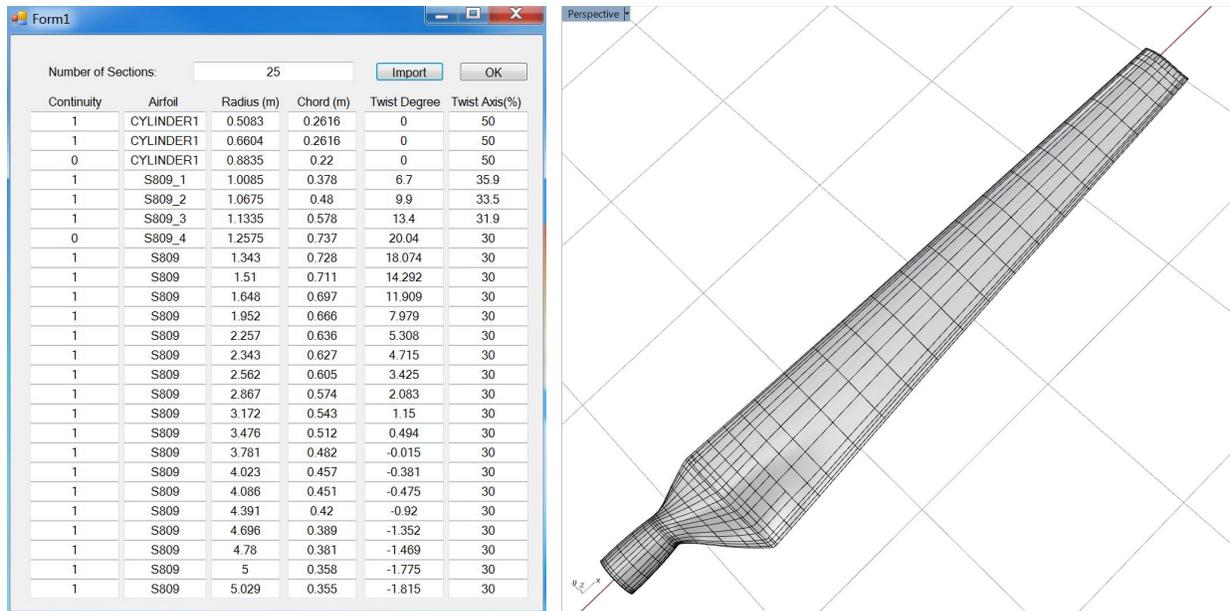


Figure 20: New design parameters and the corresponding modified wind turbine blade geometry.

overlapping of the original and modified designs, where the red wireframe represents the new geometry, and the black wireframe represents the original geometry. After following the same platform steps, the new deformation and MIPE results are shown in Figure 22. Figure 23 shows the comparison of MIPE between the redesigned and original cases. The maximum value of MIPE of the whole blade drops by 18.4% compared to the original design, and the maximum displacement decreases by 11.7%. This illustrates how the structural design improvements may be achieved within the same IGA design-through-analysis platform.

#### 4. Example 2: Integrally bladed rotor

We apply the proposed IGA design-through-analysis platform to the modeling of an IBR. We first use the IGA design-through-analysis platform to build the surface model, and then convert it to the solid NURBS representation. A template-based approach [36, 58, 104] is employed here. The approach is capable of generating good-quality volumetric models, but for a specific object, or a family of objects. Although modeling of solid geometry to this day presents a big research challenge in IGA, some innovative solutions are currently emerging that can handle geometries of greater generality [25–28, 104–108, 141].

##### 4.1. Modeling of the IBR surface geometry

The Grasshopper generative algorithm for parametric geometry design of IBR surface model is shown in Figure 24. The leftmost component represents our C# script for importing the design parameters given through the user interface. The input parameters include inner radius, outer radius

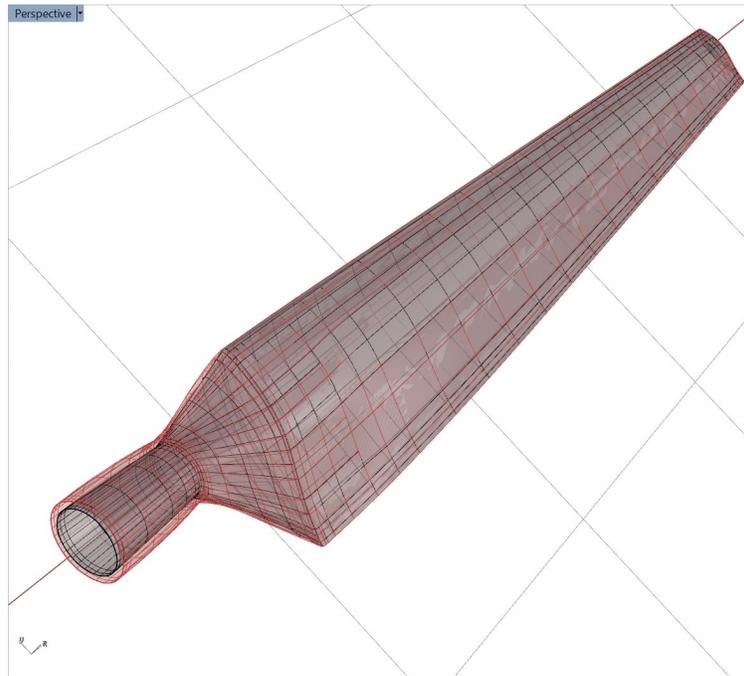


Figure 21: The comparison between the original (black wireframe) and modified (red wireframe) geometry.

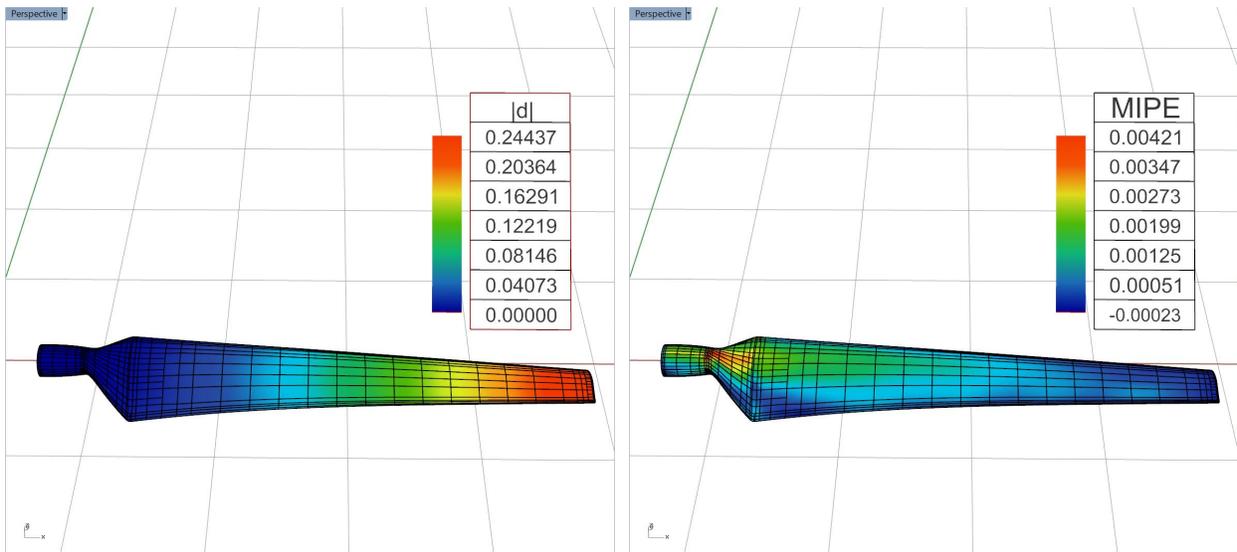


Figure 22: Displacement (left) and MIPE (right) contours of the isogeometric shell analysis results of the modified wind turbine blade.

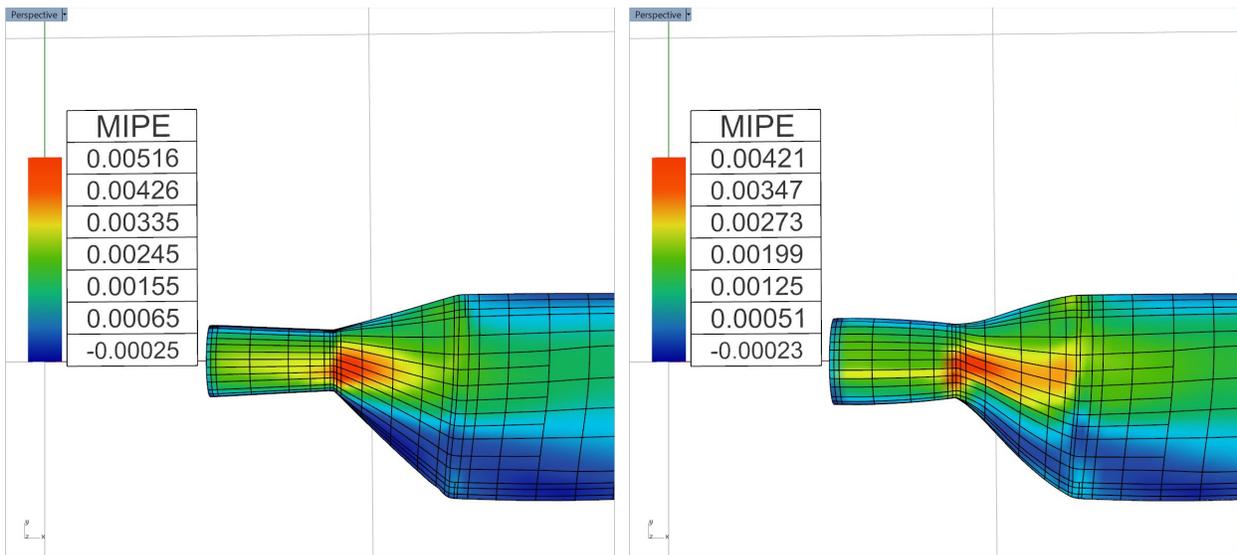


Figure 23: The comparison of the MIPE contour distributions of the original (left) and modified (right) designs.

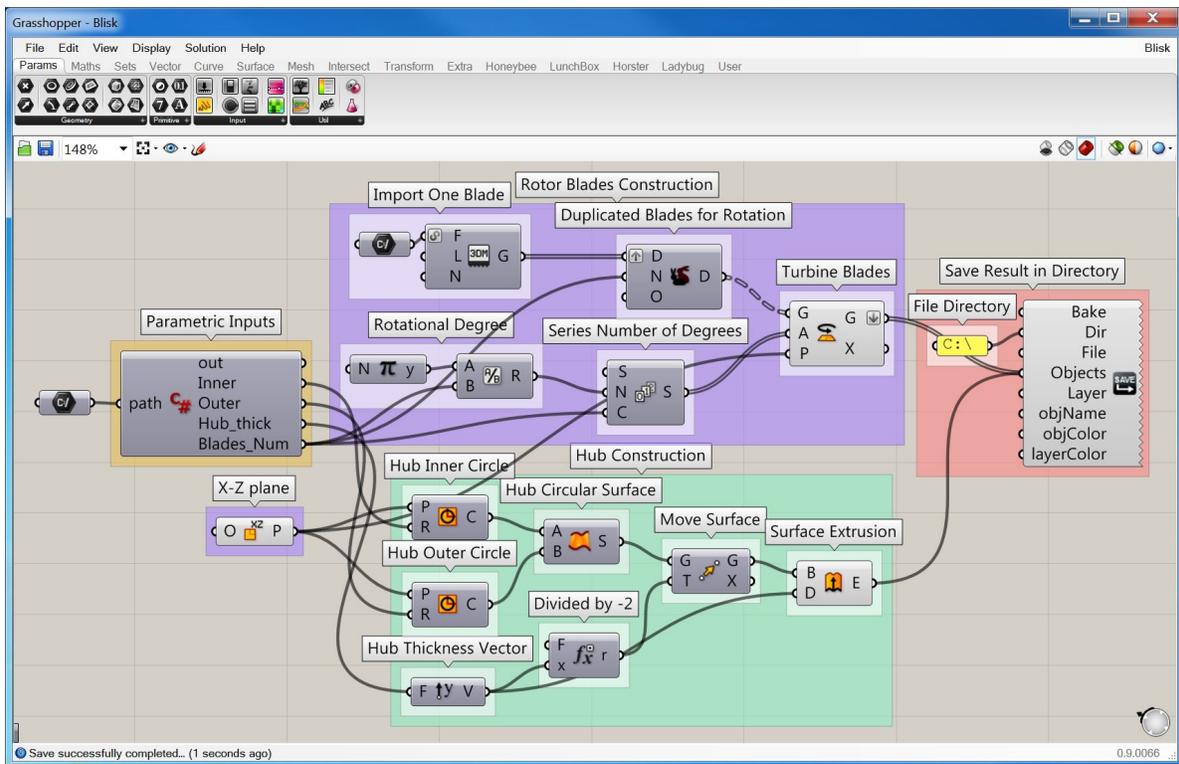


Figure 24: Grasshopper generative algorithm for parametric geometry design of IBR.

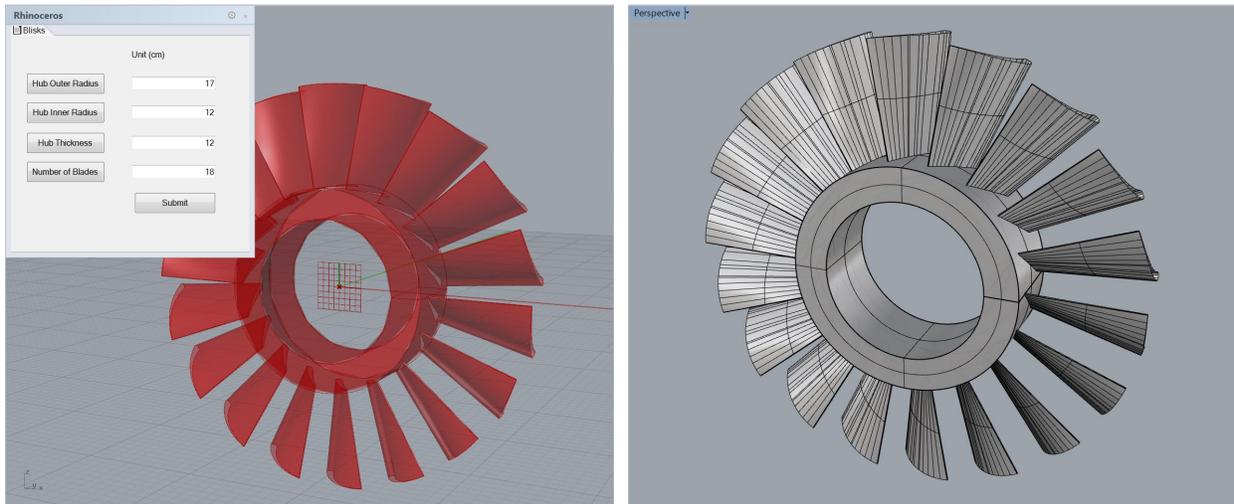


Figure 25: Parametric geometry design (left) and the NURBS surface model (right) of the 18-bladed IBR.

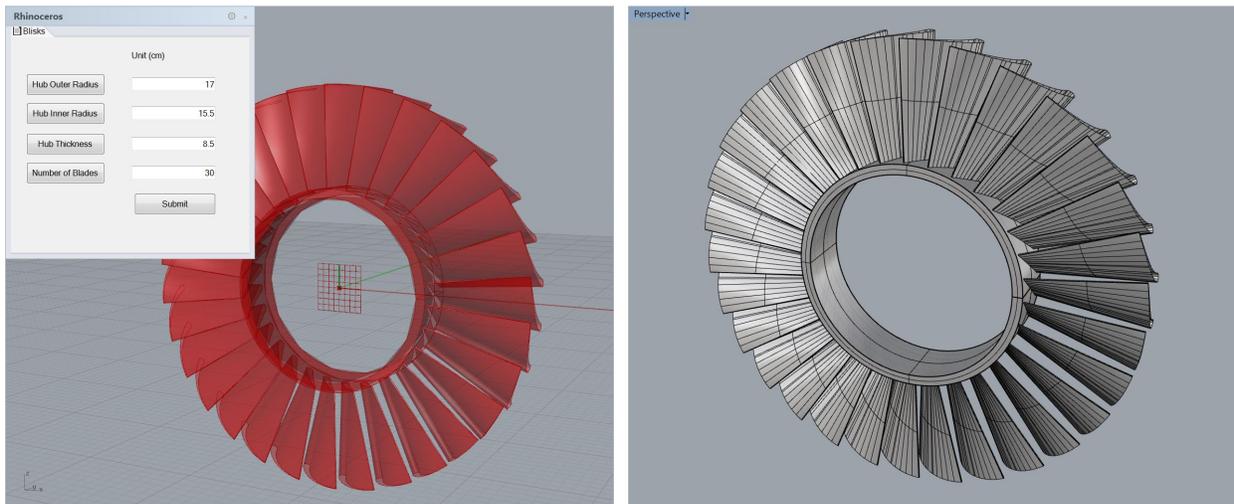


Figure 26: Parametric geometry design (left) and the NURBS surface model (right) of the 30-bladed IBR.

and thickness of the rotor hub, and the number of IBR blades. The group of components named “Rotor Blades Construction” in Figure 24 is for constructing the IBR blades. After importing an existing blade model, this imported object will be duplicated multiple times according to the number of blades that is specified by the user. Each blade will be rotated about the rotor axis by a degree from a series that has an increment of  $2\pi$  divided by the number of blades. Through this approach, the radially symmetric IBR blades are constructed.

The IBR hub is built through the group of components named “Hub Construction” in Figure 24. An annular surface is generated by using the inner and outer circles with user-defined radii. The annular surface is duplicated and translated based on the hub thickness, and block extrusion of these two surfaces generates the IBR hub surface model.

Figures 25 and 26 show NURBS surface models of 18-bladed and 30-bladed IBR, respectively,

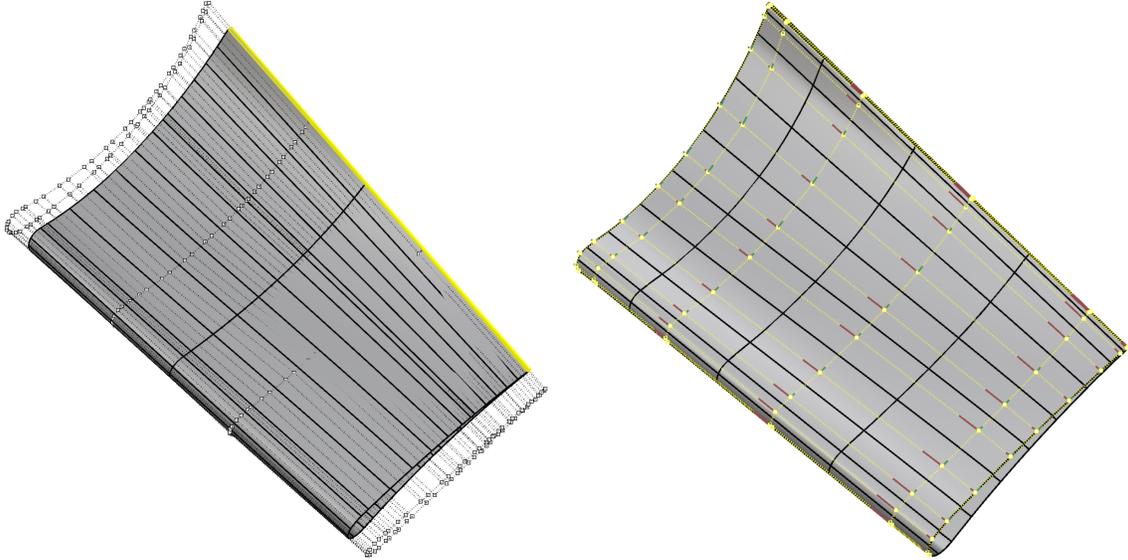


Figure 27: Left: IBR blade surface from the CAD file, which includes trimming curves. Right: IBR blade surface after pre-processing in Rhino, which includes applying the trim explicitly, splitting and merging of NURBS patches, and removing knots.

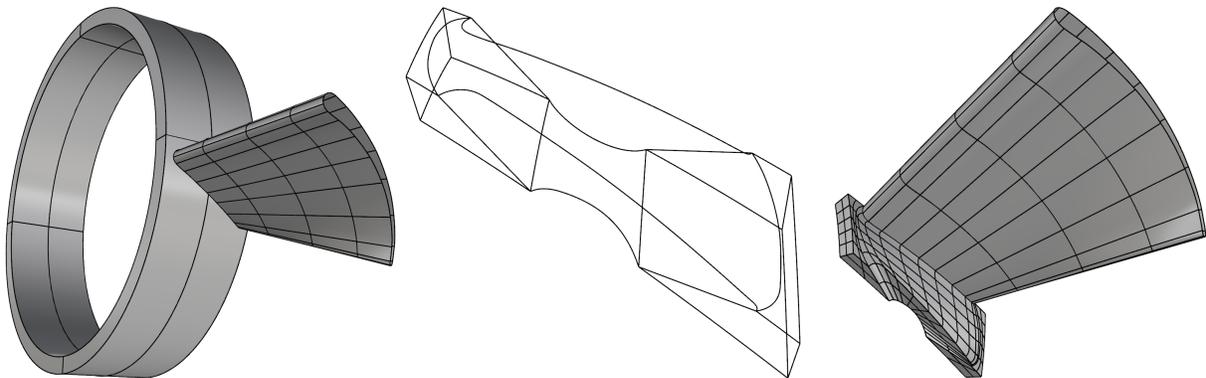


Figure 28: Left: Blade and hub with a nonmatching interface. Middle: Wireframe of the reparameterized blade-hub intersection. Right: Blade and hub with a matching interface.

designed with different hub inner radius and thickness, and number of blades. The models are generated using the proposed parametric geometry design plug-in. The 30-bladed IBR model is used for the following studies.

#### 4.2. Modeling of the IBR solid geometry

As a point of departure we started with a surface-based CAD model of a generic 30-bladed IBR shown in Figure 26. The rotor diameter is 0.68 m, the inner and outer hub diameters are 0.31 m and 0.34 m, respectively. As a first step, one of the IBR blades was pre-processed using the functionality of Rhino software, where the trimming operations were explicitly performed, patches split and merged, and knots removed for a coarser representation (See Figure 27 for details).

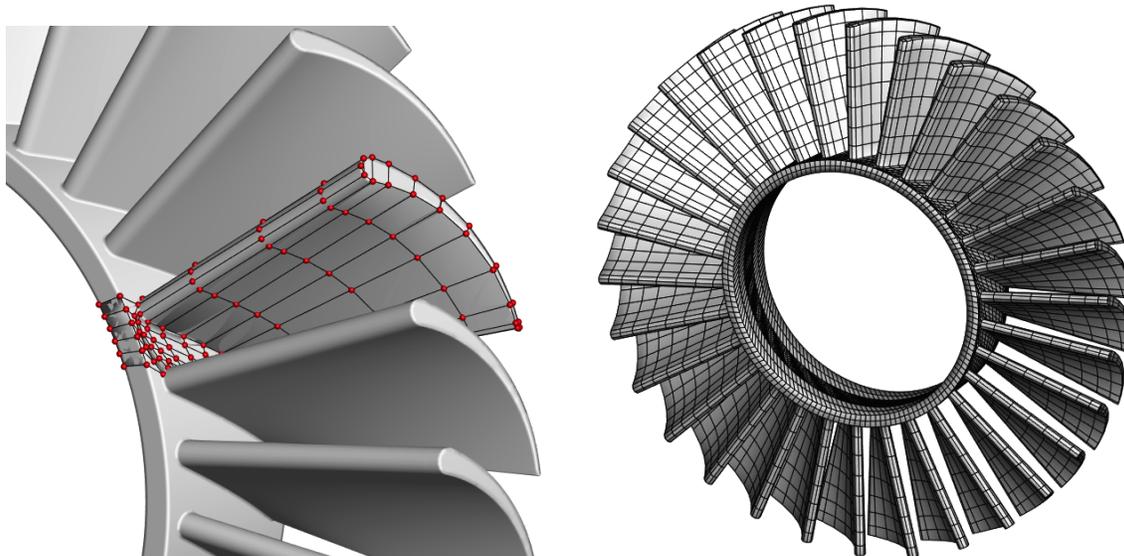


Figure 29: Left: Volumetric (trivariate) NURBS model of a periodic section of an IBR. Right: Volumetric NURBS model of the full IBR. The models are analysis-suitable.

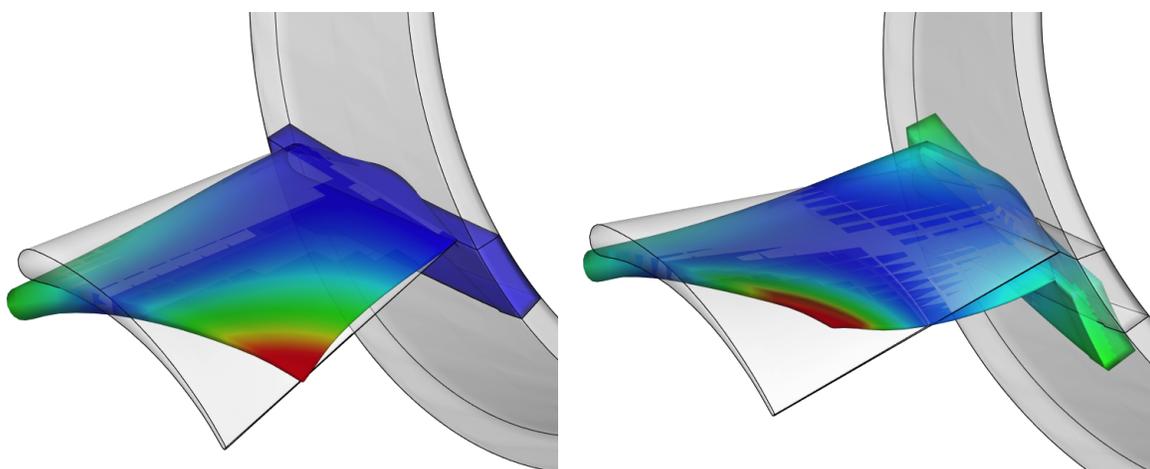


Figure 30: Twisting modes of the single-blade model superposed on the undeformed configuration. Color indicates displacement magnitude (red-high, blue-low). Left: Clamped hub. Right: Free hub. (No color scale is provided since the displacements are scaled to clearly depict the mode shapes.)

With this new representation, we can proceed with the 3D solid model construction. We first define the blade and hub intersection. Starting with a nonmatching intersection of the blade and hub, Rhino was used to reparameterize the intersection and extract its wireframe. Using the wireframe, the blade was seamlessly blended with the hub in a  $C^0$ -continuous fashion. Figure 28 shows the steps involved in the definition and reparameterization of the blade–hub intersection.

At this point the surfaces are defined in such a way that creating a volumetric (trivariate) NURBS mesh of the IBR becomes a straightforward task. Figure 29 shows the volumetric NURBS model of the 1/30 of the IBR domain superposed on the original IBR geometry as well as the full

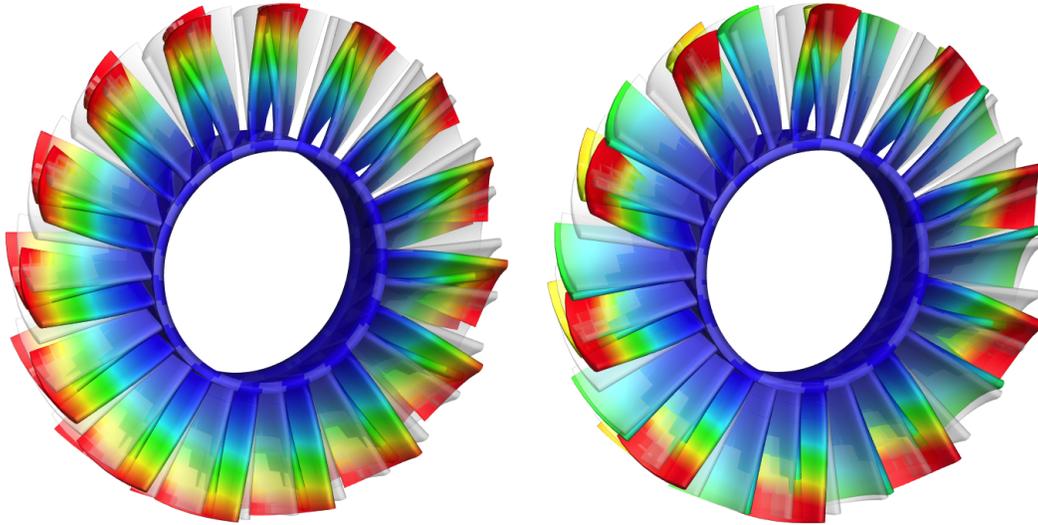


Figure 31: A selection of mode shapes from the full-IBR model with clamped hub. Color indicates displacement magnitude (red-high, blue-low). Both modes correspond to the natural frequency of 255.3 Hz. The mode shapes are superposed on the undeformed configuration. Note the symmetries present in the modes. (No color scale is provided since the displacements are scaled to clearly depict the mode shapes.)

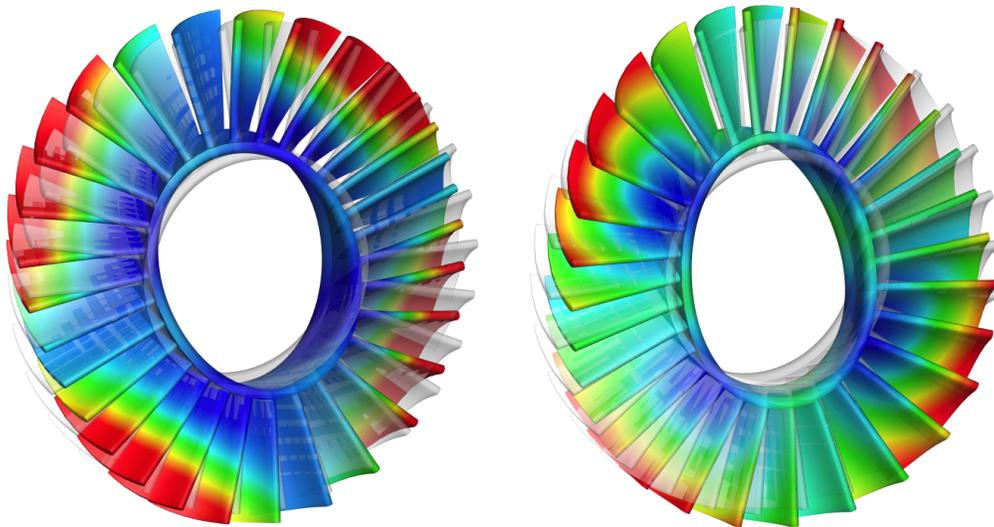


Figure 32: A selection of mode shapes from the full-IBR model with free hub. Color indicates displacement magnitude (red-high, blue-low). Mode on the left has a natural frequency of 159.3 Hz, while mode on the right has a natural frequency of 210.1 Hz. The mode shapes are superposed on the undeformed configuration. Note the symmetries present in the modes. (No color scale is provided since the displacements are scaled to clearly depict the mode shapes.)

model. The single blade model has 370 quadratic NURBS elements and 1,132 control points, while the full model is comprised of 11,100 quadratic NURBS elements and 31,680 control points. The models are analysis suitable.

#### 4.3. Structural vibration analysis of an IBR

Using the above volumetric NURBS representation we compute natural frequencies and mode shapes of the IBR structure. The rotor is made of AM355 stainless steel with Young's modulus of 210 GPa, Poisson's ratio of 0.3, and density of 8030 kg/m<sup>3</sup>. We consider a single-blade and full-IBR models and, for each one, perform natural frequency and mode shape calculations for the cases when the hub is either clamped or free. ARPACK [142], a generalized eigenvalue problem solver, is employed in the computations. The single-blade twisting modes of the single-blade model for the clamped- and free-hub cases are shown in Figure 30, and a selection of eigenmodes for the full-IBR case is shown in Figures 31 and 32.

### 5. Conclusion

We presented an isogeometric design-through-analysis platform created to help design engineers and analysts to make more effective use of IGA. The platform is built on top of Rhino CAD software, and features several plug-ins to facilitate analysis model creation for IGA, as part of a complete design-through-analysis feedback-control loop. We showed how Grasshopper, a visual programming interface, may be effectively employed to create parametric geometry designs without a burden of writing lengthy and "bug-prone" computer programs. In addition, we presented a method to enable direct visualization of NURBS and T-spline meshes and solutions defined on these meshes directly in Rhino. Finally, we demonstrated how to use the visual programming interface in combination with other existing capabilities in Rhino to create volumetric models for IGA based on the concept of template-based modeling.

The work shown in this paper presents a first step in making IGA accessible to design engineers and analysts. Much remains to be done in order to make the proposed parametric design-through-analysis platform more versatile and robust to handle realistic engineering designs in all of their complexity.

### Acknowledgements

This work is partially supported by the ARO grant No. W911NF-14-1-0296, Program Manager Dr. Joseph Myers. The work on IBR modeling was supported by NAVAIR, Program Manager Dr. Nam Phan. This material is also partially based upon work supported by the National Science Foundation under Grant No. CNS-1156841. We thank Matthew Getch, Mariama Wilson, and Alexis Moreno for their initial work on this research. A.J. Herrema is supported by the U.S. National Science Foundation Grant No. DGE-1069283 which funds the activities of the Integrative Graduate Education and Research Traineeship (IGERT) in Wind Energy Science, Engineering and Policy (WESEP) at Iowa State University.

## References

- [1] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
- [2] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, Chichester, 2009.
- [3] G. E. Farin. *NURBS Curves and Surfaces: From Projective Geometry to Practical Use*. A. K. Peters, Ltd., Natick, MA, 1995.
- [4] L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication)*, 2nd ed. Springer-Verlag, New York, 1997.
- [5] D. F. Rogers. *An Introduction to NURBS With Historical Perspective*. Academic Press, San Diego, CA, 2001.
- [6] Rhinoceros. <http://www.rhino3d.com/>. 2015.
- [7] Y. Bazilevs, L. Beirao da Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli. Isogeometric analysis: Approximation, stability and error estimates for  $h$ -refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16:1031–1090, 2006.
- [8] T. J. R. Hughes, J. A. Evans, and A. Reali. Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems. *Computer Methods in Applied Mechanics and Engineering*, 272:290–320, 2014.
- [9] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCS. *ACM Transactions on Graphics*, 22(3):477–484, 2003.
- [10] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. *ACM Transactions on Graphics*, 23(3):276–283, 2004.
- [11] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:229–263, 2010.
- [12] M. R. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local  $h$ -refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:264–275, 2010.

- [13] Y. Bazilevs, M.-C. Hsu, and M. A. Scott. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249–252:28–41, 2012.
- [14] D. Schillinger, L. Dedè, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, and T. J. R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249–252:116–150, 2012.
- [15] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. P. A. Bordas, T. J. R. Hughes, and T. W. Sederberg. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197–221, 2013.
- [16] R. Dimitri, L. De Lorenzis, M. A. Scott, P. Wriggers, R. L. Taylor, and G. Zavarise. Isogeometric large deformation frictionless contact using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 269:394–414, 2014.
- [17] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, and M. A. Scott. On linear independence of T-spline blending functions. *Computer Aided Geometric Design*, 29(1):63–76, 2012.
- [18] M. A. Scott, X. Li, T. W. Sederberg, and T. J. R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213–216:206–222, 2012.
- [19] L. Beirão da Veiga, A. Buffa, D. Cho, and G. Sangalli. Analysis-Suitable T-splines are Dual-Compatible. *Computer Methods in Applied Mechanics and Engineering*, 249–252:42–51, 2012.
- [20] Xin Li and M. A. Scott. Analysis-suitable T-splines: Characterization, refineability, and approximation. *Mathematical Models and Methods in Applied Sciences*, 24(06):1141–1164, 2014.
- [21] E. J. Evans, M. A. Scott, X. Li, and D. C. Thomas. Hierarchical T-splines: Analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:1–20, 2015.
- [22] J. M. Escobar, J. M. Cascón, E. Rodríguez, and R. Montenegro. A new approach to solid modeling with trivariate T-splines based on mesh optimization. *Computer Methods in Applied Mechanics and Engineering*, 200(45–46):3210–3222, 2011.

- [23] W. Wang, Y. Zhang, M. A. Scott, and T. J. R. Hughes. Converting an unstructured quadrilateral mesh to a standard T-spline surface. *Computational Mechanics*, 48:477–498, 2011.
- [24] W. Wang, Y. Zhang, M. A. Scott, and T. J. R. Hughes. Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline. *Computational Mechanics*, 50:65–84, 2012.
- [25] Y. Zhang, W. Wang, and T. J. R. Hughes. Solid T-spline construction from boundary representations for genus-zero geometry. *Computer Methods in Applied Mechanics and Engineering*, 249-252:185–197, 2012.
- [26] Y. Zhang, W. Wang, and T. J. R. Hughes. Conformal solid T-spline construction from boundary T-spline representations. *Computational Mechanics*, 51(6):1051–1059, 2013.
- [27] W. Wang, Y. Zhang, L. Liu, and T. J. R. Hughes. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. *Computer-Aided Design*, 45(2):351–360, 2013.
- [28] L. Liu, Y. Zhang, T. J. R. Hughes, M. A. Scott, and T. W. Sederberg. Volumetric T-spline construction using Boolean operations. *Engineering with Computers*, 30(4):425–439, 2014.
- [29] L. Liu, Y. Zhang, Y. Liu, and W. Wang. Feature-preserving T-mesh construction using skeleton-based polycubes. *Computer-Aided Design*, 58:162–172, 2015.
- [30] Y. Bazilevs, V. M. Calo, J. A. Cottrell, T. J. R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007.
- [31] Y. Bazilevs, C. Michler, V. M. Calo, and T. J. R. Hughes. Weak Dirichlet boundary conditions for wall-bounded turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 196:4853–4862, 2007.
- [32] I. Akkerman, Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and S. Hulshoff. The role of continuity in residual-based variational multiscale modeling of turbulence. *Computational Mechanics*, 41:371–378, 2008.
- [33] Y. Bazilevs, C. Michler, V. M. Calo, and T. J. R. Hughes. Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. *Computer Methods in Applied Mechanics and Engineering*, 199:780–790, 2010.

- [34] M.-C. Hsu, Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199:828–840, 2010.
- [35] Y. Bazilevs and I. Akkerman. Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual–based variational multiscale method. *Journal of Computational Physics*, 229:3402–3414, 2010.
- [36] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. E. Tezduyar. 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids*, 65:207–235, 2011.
- [37] I. Akkerman, Y. Bazilevs, C. E. Kees, and M. W. Farthing. Isogeometric analysis of free-surface flow. *Journal of Computational Physics*, 230:4137–4152, 2011.
- [38] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Computers & Fluids*, 49:93–100, 2011.
- [39] J. A. Evans and T. J. R. Hughes. Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations. *Journal of Computational Physics*, 241:141–167, 2013.
- [40] J. A. Cottrell, A. Reali, Y. Bazilevs, and T. J. R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5297, 2006.
- [41] T. Elguedj, Y. Bazilevs, V. M. Calo, and T. J. R. Hughes.  $\bar{\mathbf{B}}$  and  $\bar{\mathbf{F}}$  projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer Methods in Applied Mechanics and Engineering*, 197:2732–2762, 2008.
- [42] S. Lipton, J. A. Evans, Y. Bazilevs, T. Elguedj, and T. J. R. Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199:357–373, 2010.
- [43] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.
- [44] D. J. Benson, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. Isogeometric shell analysis: The Reissner–Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199:276–289, 2010.

- [45] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199:2403–2416, 2010.
- [46] C. V. Verhoosel, M. A. Scott, T. J. R. Hughes, and R. de Borst. An isogeometric analysis approach to gradient damage models. *International Journal for Numerical Methods in Engineering*, 86:115–134, 2011.
- [47] Y. Bazilevs, M.-C. Hsu, J. Kiendl, and D. J. Benson. A computational procedure for prebending of wind turbine blades. *International Journal for Numerical Methods in Engineering*, 89:323–336, 2012.
- [48] D. J. Benson, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. A large deformation, rotation-free, isogeometric shell. *Computer Methods in Applied Mechanics and Engineering*, 200:1367–1378, 2011.
- [49] E. De Luycker, D. J. Benson, T. Belytschko, Y. Bazilevs, and M.-C. Hsu. X-FEM in isogeometric analysis for linear fracture mechanics. *International Journal for Numerical Methods in Engineering*, 87:541–565, 2011.
- [50] D. J. Benson, S. Hartmann, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. Blended isogeometric shells. *Computer Methods in Applied Mechanics and Engineering*, 255:133–146, 2013.
- [51] S. Hosseini, J. J. C. Remmers, C. V. Verhoosel, and R. de Borst. An isogeometric continuum shell element for non-linear analysis. *Computer Methods in Applied Mechanics and Engineering*, 271:1–22, 2014.
- [52] P. Stein and B. Xu. 3D Isogeometric Analysis of intercalation-induced stresses in Li-ion battery electrode particles. *Computer Methods in Applied Mechanics and Engineering*, 268:225–244, 2014.
- [53] S. Morganti, F. Auricchio, D. J. Benson, F. I. Gambarin, S. Hartmann, T. J. R. Hughes, and A. Reali. Patient-specific isogeometric structural analysis of aortic valve closure. *Computer Methods in Applied Mechanics and Engineering*, 284:508–520, 2015.
- [54] C. V. Verhoosel, G. J. van Zwieten, B. van Rietbergen, and R. de Borst. Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone. *Computer Methods in Applied Mechanics and Engineering*, 284:138–164, 2015.

- [55] X. Deng, A. Korobenko, J. Yan, and Y. Bazilevs. Isogeometric analysis of continuum damage in rotation-free composite shells. *Computer Methods in Applied Mechanics and Engineering*, 284:349–372, 2015.
- [56] J. Kiendl, M.-C. Hsu, M. C. H. Wu, and A. Reali. Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering*, 2015. <http://dx.doi.org/10.1016/j.cma.2015.03.010>.
- [57] Y. Bazilevs, V. M. Calo, Y. Zhang, and T. J. R. Hughes. Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38:310–322, 2006.
- [58] Y. Zhang, Y. Bazilevs, S. Goswami, C. Bajaj, and T. J. R. Hughes. Patient-specific vascular nurbs modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196:2943–2959, 2007.
- [59] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid–structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43:3–37, 2008.
- [60] J. G. Isaksen, Y. Bazilevs, T. Kvamsdal, Y. Zhang, J. H. Kaspersen, K. Waterloo, B. Romner, and T. Ingebrigtsen. Determination of wall tension in cerebral artery aneurysms by numerical simulation. *Stroke*, 39:3172–3178, 2008.
- [61] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65:236–253, 2011.
- [62] M.-C. Hsu and Y. Bazilevs. Fluid–structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50:821–833, 2012.
- [63] M.-C. Hsu, D. Kamensky, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. *Computational Mechanics*, 54(4):1055–1071, 2014.
- [64] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.
- [65] H. Gomez, V. M. Calo, Y. Bazilevs, and T. J. R. Hughes. Isogeometric analysis of the Cahn–Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197:4333–4352, 2008.

- [66] M. J. Borden, M. A. Scott, C. V. Verhoosel, T. J. R. Hughes, and C. M. Landis. A phase-field description of dynamic brittle fracture. *Computer Methods in Applied Mechanics and Engineering*, 217–220:77–95, 2012.
- [67] G. Vilanova, I. Colominas, and H. Gómez. Capillary networks in tumor angiogenesis: From discrete endothelial cells to phase-field averaged descriptions via isogeometric analysis. *International Journal for Numerical Methods in Biomedical Engineering*, 29(10):1015–1037, 2013.
- [68] M. J. Borden, T. J. R. Hughes, C. M. Landis, and C. V. Verhoosel. A higher-order phase-field model for brittle fracture: Formulation and analysis within the isogeometric analysis framework. *Computer Methods in Applied Mechanics and Engineering*, 273:100–118, 2014.
- [69] R. P. Dhote, H. Gomez, R. N. V. Melnik, and J. Zu. Isogeometric analysis of a dynamic thermo-mechanical phase-field model applied to shape memory alloys. *Computational Mechanics*, 53(6):1235–1250, 2014.
- [70] J. Liu, L. Dedè, J. A. Evans, M. J. Borden, and T. J. R. Hughes. Isogeometric analysis of the advective Cahn–Hilliard equation: Spinodal decomposition under shear flow. *Journal of Computational Physics*, 242:321–350, 2013.
- [71] J. Liu, H. Gómez, J. A. Evans, T. J. R. Hughes, and C. M. Landis. Functional entropy variables: A new methodology for deriving thermodynamically consistent algorithms for complex fluids, with particular reference to the isothermal Navier–Stokes–Korteweg equations. *Journal of Computational Physics*, 248:47–86, 2013.
- [72] Y. Bazilevs, I. Akkerman, D. J. Benson, G. Scovazzi, and M. J. Shashkov. Isogeometric analysis of Lagrangian hydrodynamics. *Journal of Computational Physics*, 243:224–243, 2013.
- [73] Y. Bazilevs, C. C. Long, I. Akkerman, D. J. Benson, and M. J. Shashkov. Isogeometric analysis of Lagrangian hydrodynamics: Axisymmetric formulation in the  $rz$ -cylindrical coordinates. *Journal of Computational Physics*, 262:244–261, 2014.
- [74] L. De Lorenzis, P. Wriggers, and G. Zavarise. A mortar formulation for 3D large deformation contact using NURBS-based isogeometric analysis and the augmented Lagrangian method. *Computational Mechanics*, 49(1):1–20, 2011.
- [75] L. De Lorenzis, İ. Temizer, P. Wriggers, and G. Zavarise. A large deformation frictional contact formulation using NURBS-based isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 87:1278–1300, 2011.

- [76] Í. Temizer, P. Wriggers, and T. J. R. Hughes. Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):1100–1112, 2011.
- [77] Í. Temizer, P. Wriggers, and T. J. R. Hughes. Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, 209-212:115–128, 2012.
- [78] M. Dittmann, M. Franke, Í. Temizer, and C. Hesch. Isogeometric Analysis and thermomechanical Mortar contact problems. *Computer Methods in Applied Mechanics and Engineering*, 274:192–212, 2014.
- [79] C. J. Corbett and R. A. Sauer. NURBS-enriched contact finite elements. *Computer Methods in Applied Mechanics and Engineering*, 275:55–75, 2014.
- [80] J. Lu and C. Zheng. Dynamic cloth simulation by isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 268:475–493, 2014.
- [81] L. De Lorenzis, P. Wriggers, and T. J. R. Hughes. Isogeometric contact: a review. *GAMM-Mitteilungen*, 37(1):85–123, 2014.
- [82] W. A. Wall, M. A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197:2976–2988, 2008.
- [83] Y. D. Seo, H. J. Kim, and S. K. Youn. Isogeometric topology optimization using trimmed spline surfaces. *Computer Methods in Applied Mechanics and Engineering*, 199:3270–3296, 2010.
- [84] J. Kiendl, R. Schmidt, R. Wüchner, and K.-U. Bletzinger. Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. *Computer Methods in Applied Mechanics and Engineering*, 274:148–167, 2014.
- [85] K. V. Kostas, A. I. Ginnis, C. G. Politis, and P. D. Kaklis. Ship-hull shape optimization with a T-spline based BEM-isogeometric solver. *Computer Methods in Applied Mechanics and Engineering*, 284:611–622, 2015.
- [86] F. Auricchio, L. Beirão da Veiga, T. J. R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation for elastostatics and explicit dynamics. *Computer Methods in Applied Mechanics and Engineering*, 249-252:2–14, 2012.

- [87] D. Schillinger, J. A. Evans, A. Reali, M. A. Scott, and T. J. R. Hughes. Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Computer Methods in Applied Mechanics and Engineering*, 267:170–232, 2013.
- [88] F. Auricchio, L. Beirão da Veiga, J. Kiendl, C. Lovadina, and A. Reali. Locking-free isogeometric collocation methods for spatial Timoshenko rods. *Computer Methods in Applied Mechanics and Engineering*, 263:113–126, 2013.
- [89] H. Gómez, A. Reali, and G. Sangalli. Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models. *Journal of Computational Physics*, 262:153–171, 2014.
- [90] A. Reali and H. Gómez. An isogeometric collocation approach for Bernoulli–Euler beams and Kirchhoff plates. *Computer Methods in Applied Mechanics and Engineering*, 284:623–636, 2015.
- [91] L. De Lorenzis, J. A. Evans, T. J. R. Hughes, and A. Reali. Isogeometric collocation: Neumann boundary conditions and contact. *Computer Methods in Applied Mechanics and Engineering*, 284:21–54, 2015.
- [92] D. Schillinger, M. J. Borden, and H. K. Stolarski. Isogeometric collocation for phase-field fracture models. *Computer Methods in Applied Mechanics and Engineering*, 284:583–610, 2015.
- [93] J. Kiendl, F. Auricchio, L. Beirão da Veiga, C. Lovadina, and A. Reali. Isogeometric collocation methods for the Reissner–Mindlin plate problem. *Computer Methods in Applied Mechanics and Engineering*, 284:489–507, 2015.
- [94] F. Auricchio, F. Calabrò, T. J. R. Hughes, A. Reali, and G. Sangalli. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 249-252:15–27, 2012.
- [95] D. Schillinger, S. J. Hossain, and T. J. R. Hughes. Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 277:1–45, 2014.
- [96] A. P. Nagy and D. J. Benson. On the numerical integration of trimmed isogeometric elements. *Computer Methods in Applied Mechanics and Engineering*, 284:165–185, 2015.

- [97] M. Hillman, J. S. Chen, and Y. Bazilevs. Variationally consistent domain integration for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:521–540, 2015.
- [98] R. Schmidt, R. Wüchner, and K.-U. Bletzinger. Isogeometric analysis of trimmed NURBS geometries. *Computer Methods in Applied Mechanics and Engineering*, 241-244:93–111, 2012.
- [99] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank. Weakly enforced essential boundary conditions for nurbs-embedded and trimmed nurbs geometries on the basis of the finite cell method. *International Journal for Numerical Methods in Engineering*, 95:811–846, 2013.
- [100] V. P. Nguyen, P. Kerfriden, M. Brino, S. P. A. Bordas, and E. Bonisoli. Nitsche’s method for two and three dimensional NURBS patch coupling. *Computational Mechanics*, 53(6):1163–1182, 2014.
- [101] A. Apostolatos, R. Schmidt, R. Wüchner, and K.-U. Bletzinger. A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 97(7):473–504, 2014.
- [102] M. Ruess, D. Schillinger, A. I. Özcan, and E. Rank. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 269:46–731, 2014.
- [103] Y. Guo and M. Ruess. Nitsche’s method for a coupling of isogeometric thin shells and blended shell structures. *Computer Methods in Applied Mechanics and Engineering*, 284:881–905, 2015.
- [104] P. Stein and K. Beucke M.-C. Hsu, Y. Bazilevs. Operator- and template-based modeling of solid geometry for isogeometric analysis with application to vertical axis wind turbine simulation. *Computer Methods in Applied Mechanics and Engineering*, 213–216:71–83, 2012.
- [105] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. *Computer-Aided Design*, 45(2):395–404, 2013.
- [106] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method. *Journal of Computational Physics*, 252:275–289, 2013.

- [107] G. Xu, B. Mourrain, A. Galligo, and T. Rabczuk. High-quality construction of analysis-suitable trivariate NURBS solids by reparameterization methods. *Computational Mechanics*, 54(5):1303–1313, 2014.
- [108] B.-Q. Zuo, Z.-D. Huang, Y.-W. Wang, and Z.-J. Wu. Isogeometric analysis for CSG models. *Computer Methods in Applied Mechanics and Engineering*, 285:102–124, 2015.
- [109] D. J. Benson, Y. Bazilevs, E. De Luycker, M.-C. Hsu, M. Scott, T. J. R. Hughes, and T. Be-lytschko. A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM. *International Journal for Numerical Methods in Engineering*, 83:765–785, 2010.
- [110] M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87:15–47, 2011.
- [111] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88:126–156, 2011.
- [112] M. A. Scott, T. J. R. Hughes, T. W. Sederberg, and M. T. Sederberg. An integrated approach to engineering design and analysis using the Autodesk T-spline plugin for Rhino3d. ICES REPORT 14-33, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, September 2014, 2014.
- [113] D. C. Thomas, M. A. Scott, J. A. Evans, K. Tew, and E. J. Evans. Bézier projection: A unified approach for local projection and quadrature-free refinement and coarsening of NURBS and T-splines with particular application to isogeometric design and analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:55–105, 2015.
- [114] J. Parvizian, A. Düster, and E. Rank. Finite cell method. *Computational Mechanics*, 41(1):121–133, 2007.
- [115] A. Düster, J. Parvizian, Z. Yang, and E. Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197(45–48):3768–3782, 2008.
- [116] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster. Geometric modeling, isogeometric analysis and the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 249-252:104–115, 2012.

- [117] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, and E. Rank. Small and large deformation analysis with the  $p$ - and B-spline versions of the Finite Cell Method. *Computational Mechanics*, 50(4):445–478, 2012.
- [118] D. Schillinger and M. Ruess. The Finite Cell Method: A review in the context of higher-order structural analysis of CAD and image-based geometric models. *Archives of Computational Methods in Engineering*, 2014. <http://dx.doi.org/10.1007/s11831-014-9115-y>.
- [119] Y. Bazilevs, M.-C. Hsu, K.-U. Bletzinger, M. Breitenberger, R. Wüchner, J. Kiendl, and R. Schmidt. Three-dimensional geometric design, analysis, and optimization of shell structures, 2012. U.S. Patent Application Publication No.: US 2012/0310604 A1.
- [120] M. Breitenberger, A. Apostolatos, B. Philipp, R. Wüchner, and K.-U. Bletzinger. Analysis in computer aided design: Nonlinear isogeometric B-Rep analysis of shell structures. *Computer Methods in Applied Mechanics and Engineering*, 284:401–457, 2015.
- [121] Siemens NX. <http://www.plm.automation.siemens.com/>. 2015.
- [122] Grasshopper. <http://www.grasshopper3d.com/>. 2015.
- [123] M. Stavric and O. Marina. Parametric modeling for advanced architecture. *International Journal of Applied Mathematics and Informatics*, 5(1):9–16, 2011.
- [124] SOLIDWORKS. <http://www.solidworks.com/>. 2015.
- [125] W. F. LaBozetta and P. E. Cole. Interactive graphics for geometry generation—A program with a contemporary design. *Journal of Aircraft*, 22:1054–1058, 1985.
- [126] J. C. Trapp and H. Sobieszky. Interactive parametric geometry design. In *Proceedings of 37th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 99-0829, Reno, NV, 1999.
- [127] A. Azamatov, J.-W. Lee, and Y.-H. Byun. Comprehensive aircraft configuration design tool for integrated product and process development. *Advances in Engineering Software*, 42:35–49, 2011.
- [128] E. Cohen, T. Martin, R. M. Kirby, T. Lyche, and R. F. Riesenfeld. Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:334–356, 2010.
- [129] W. Wang and Y. Zhang. Wavelets-based NURBS simplification and fairing. *Computer Methods in Applied Mechanics and Engineering*, 199:290–300, 2010.

- [130] Autodesk T-Splines Plug-in for Rhino. <http://www.tsplines.com/products/tsplines-for-rhino.html>. 2015.
- [131] Rhino Developer Tools. <http://wiki.mcneel.com/developer/home>. 2015.
- [132] R. Schmidt, J. Kiendl, K.-U. Bletzinger, and R. Wüchner. Realization of an integrated structural design process: analysis-suitable geometric modelling and isogeometric analysis. *Computing and Visualization in Science*, 13(7):315–330, 2011.
- [133] RhinoCommon Plug-in SDK. <http://wiki.mcneel.com/developer/rhinocommon/>. 2015.
- [134] D. M. Somers. Design and experimental results for the S809 airfoil. Technical Report NREL/SR-440-6918, National Renewable Energy Laboratory, Golden, CO, 1997.
- [135] M. M. Hand, D. A. Simms, L. J. Fingersh, D. W. Jager, J. R. Cotrell, S. Schreck, and S. M. Larwood. Unsteady aerodynamics experiment phase VI: Wind tunnel test configurations and available data campaigns. Technical Report NREL/TP-500-29955, National Renewable Energy Laboratory, Golden, CO, 2001.
- [136] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. Finite element simulation of wind turbine aerodynamics: Validation study using NREL Phase VI experiment. *Wind Energy*, 17:461–481, 2014.
- [137] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, 2009.
- [138] J. Kiendl. *Isogeometric Analysis and Shape Optimal Design of Shell Structures*. PhD thesis, Lehrstuhl für Statik, Technische Universität München, 2011.
- [139] Numerical Manufacturing And Design Tool (NuMAD). <http://energy.sandia.gov/energy/renewable-energy/wind-power/rotor-innovation/numerical-manufacturing-and-design-tool-numad/>. 2015.
- [140] I. M. Daniel and O. Ishai. *Engineering Mechanics of Composite Materials, 2nd ed.* Oxford University Press, New York, NY, 2006.
- [141] X. Wang and X. Qian. An optimization approach for constructing trivariate B-spline solids. *Computer-Aided Design*, 46:179–191, 2014.
- [142] ARPACK. <http://www.caam.rice.edu/software/arpack/>. 2014.