Direct immersogeometric fluid flow analysis using B-rep CAD models

CrossMark

Ming-Chen Hsu *, Chenglong Wang, Fei Xu, Austin J. Herrema, Adarsh Krishnamurthy

Department of Mechanical Engineering, Iowa State University, 2025 Black Engineering, Ames, IA 50011, USA

ARTICLE INFO

ABSTRACT

We present a new method for immersogeometric fluid flow analysis that directly uses the CAD boundary representation (B-rep) of a complex object and immerses it into a locally refined, non-boundary-fitted discretization of the fluid domain. The motivating applications include analyzing the flow over complex geometries, such as moving vehicles, where the detailed geometric features usually require time-consuming, labor-intensive geometry cleanup or mesh manipulation for generating the surrounding boundary-fitted fluid mesh. The proposed method avoids the challenges associated with such procedures. A new method to perform point membership classification of the background mesh quadrature points is also proposed. To faithfully capture the geometry in intersected elements, we implement an adaptive quadrature rule based on the recursive splitting of elements. Dirichlet boundary conditions in intersected elements are enforced weakly in the sense of Nitsche's method. To assess the accuracy of the proposed method, we perform computations of the benchmark problem of flow over a sphere represented using B-rep. Quantities of interest such as drag coefficient are in good agreement with reference values reported in the literature. The results show that the density and distribution of the surface quadrature points are crucial for the weak enforcement of Dirichlet boundary conditions and for obtaining accurate flow solutions. Also, with sufficient levels of surface quadrature element refinement, the quadrature error near the trim curves becomes insignificant. Finally, we demonstrate the effectiveness of our immersogeometric method for high-fidelity industrial scale simulations by performing an aerodynamic analysis of an agricultural tractor directly represented using B-rep.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Immersogeometric analysis was first introduced by Kamensky et al. (2015) as a geometrically flexible technique for solving computational fluid–structure interaction (FSI) problems involving large, complex structural deformation. The method analyzed a surface representation of the structure by immersing it into a non-boundary-fitted discretization of the background fluid domain and focused on accurately capturing the immersed geometry (and hence the name *immersogeometric*) within non-boundary-fitted analysis meshes. The method was successfully applied to the FSI simulation of bioprosthetic heart valves (Kamensky et al., 2015; Hsu et al., 2014, 2015a). The immersogeometric method was further investigated

---

* Corresponding author.
   E-mail address: jmchsu@iastate.edu (M.-C. Hsu).

by Xu et al. (2016) in the context of a tetrahedral finite cell approach (Varduhn et al., 2016) for the simulation of incompressible flow, both laminar and turbulent, around geometrically complex objects. The motivation was to alleviate the difficulties associated with computational fluid dynamics (CFD) mesh generation around complex design geometries. Typical mechanical computer-aided designs (CAD) are so complex that they cannot be handled by CFD mesh generation software independently and automatically, but require human analysts to perform intermediate steps such as defeaturing, geometry cleanup, and mesh manipulation (Marcum and Gaither, 2000; Wang and Srinivasan, 2002; Beall et al., 2004; Lee et al., 2010). The immersogeometric method was proposed to eliminate these labor-intensive procedures from the CFD simulation pipeline while still maintaining high accuracy of the simulation results.

In recent years, the development of isogeometric analysis (IGA) (Hughes et al., 2005; Cottrell et al., 2009) 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 CAD and the approximation of solution fields in finite element analysis (FEA). Aside from its potential to eliminate unnecessary labor from the design-through-analysis pipeline (Schillinger et al., 2012a; Breitenberger et al., 2015; Hsu et al., 2015b), IGA has attracted a great deal of attention due to the improvements in solution quality that follow from incorporation of smooth basis functions into engineering analysis (Cottrell et al., 2007; Akkerman et al., 2008). Over the last decade, IGA has been successfully employed in many areas of engineering and sciences, such as fluid mechanics and turbulence (Bazilevs et al., 2007a; Bazilevs and Akkerman, 2010; Akkerman et al., 2011; Evans and Hughes, 2013; Liu et al., 2013), structural and contact mechanics (Cottrell et al., 2006; Kiendl et al., 2009; Benson et al., 2010b; De Lorenzis et al., 2011; Temizer et al., 2012), fluid–structure interactions (Bazilevs et al., 2008, 2012), phase-field modeling (Gomez et al., 2008; Borden et al., 2014), collocation (Auricchio et al., 2010; Schillinger et al., 2013; Reali and Gómez, 2015), efficient quadrature rules (Hughes et al., 2010; Auricchio et al., 2012; Schillinger et al., 2014), boundary element methods (Simpson et al., 2012; Scott et al., 2013), shape and topology optimization (Dedè et al., 2012; Kiendl et al., 2014; Kostas et al., 2015), finite cell methods (Rank et al., 2012; Schillinger et al., 2012b; Schillinger and Ruess, 2015), trimmed geometries and patch coupling (Schmidt et al., 2012; Ruess et al., 2014; Guo and Ruess, 2015), analysis-suitable trivariate models (Zhang et al., 2007; Stein et al., 2012; Liu et al., 2014), T-splines (Bazilevs et al., 2010a; Scott et al., 2012; Li and Scott, 2014), and standardized file formats for data exchange between CAD and FEA packages (Benson et al., 2010a; Borden et al., 2011; Scott et al., 2011).

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 construction of analysis-suitable geometric models. The typical industry standard for the representation of geometry in mechanical CAD systems is non-uniform rational B-splines (NURBS)-based boundary representations (B-reps). Although B-reps are ubiquitous in the CAD industry, they are not commonly used in analysis due to the challenges associated with directly using the B-rep information to perform geometric operations such as surface integration. Hence, the common practice in mechanical analysis (CFD or FEA) is to preprocess the B-reps by tessellating them into triangles and then using the triangular surface mesh in the simulations. Generating the surface tessellations of complex CAD models is time-consuming and labor intensive, since the geometry needs to be manually checked to avoid creating any intersecting or non-manifold features (such as hanging nodes) during tessellation. In addition, the tessellation of curved surfaces represented using spline surfaces introduces a tessellation error depending on the size of the triangles used to approximate them.

A pioneering work using NURBS-based B-rep models directly in nonlinear isogeometric shell analysis was presented by Breitenberger et al. (2015). However, directly using B-rep models in flow analysis is still limited because for flow simulations, the meshing of the surrounding fluid domain needs to be considered in addition to having the object surface discretization. Generating a high-quality boundary-fitted fluid mesh requires intense manipulation of the surface mesh. Although there have been advances in using analysis-suitable trivariate T-splines (Zhang et al., 2013; Wang et al., 2013; Liu et al., 2014) for volumetric discretization, using T-splines for CAD and CFD meshing continues to be limited by the geometric problems associated with the surface. To overcome these challenges, we present a novel method for immersogeometric fluid flow analysis that directly uses the CAD B-rep of a complex object and immerses it into a non-boundary-fitted discretization of the surrounding fluid domain. This work is inspired by Rank et al. (2012), who proposed to extend the finite cell method (Parvizian et al., 2007; Düster et al., 2008) to use CSG-tree and B-rep information for point membership classification, such that geometric models can be directly used in the finite cell analysis.

The immersogeometric method for CFD is comprised of the following main components. A variational multiscale (VMS) formulation of incompressible flow (Hughes et al., 2000, 2001, 2004; Bazilevs et al., 2007a) is used, which provides accuracy and robustness in both laminar and turbulent flow conditions. The Dirichlet boundary conditions on the surface of the immersed objects are enforced weakly in the sense of Nitsche's method (Nitsche, 1970; Bazilevs and Hughes, 2007). Adaptively refined quadrature rules are used to faithfully capture the flow domain geometry in the discrete problem without modifying the non-boundary-fitted background mesh. It was found by Xu et al. (2016) that the faithful representation of the geometry in intersected elements is critical for accurate immersogeometric fluid flow analysis.

To simulate fluid flow past B-rep CAD models using the immersogeometric method, two key geometric operations need to be performed. The first is to evaluate points on the surface of the solid model to enforce Dirichlet boundary conditions. The second is to perform point membership classification on the points belonging to the background mesh to identify points inside the solid CAD model. Both operations need to be performed adaptively for the CFD analysis to be accurate and converged. In this paper, we have developed new methods to perform these operations directly using the B-rep of the CAD model.
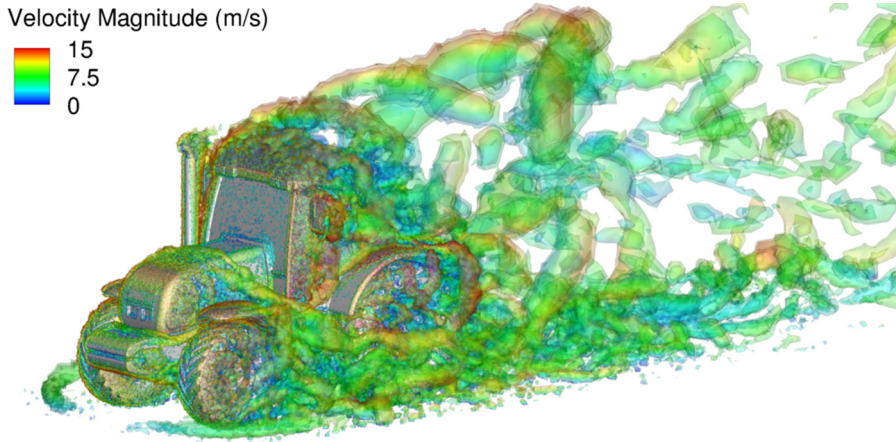
Fig. 1. Visualization of the instantaneous vortical structures of turbulent flow around a tractor colored by velocity magnitude. The B-rep of the tractor consisting of trimmed NURBS surfaces is used directly to perform immersogeometric flow analysis.

The boundary of solid models created using CAD systems is usually represented using multiple trimmed NURBS surfaces. Previous implementations of immersogeometric analysis using complex CAD models rely on triangular tessellation of surfaces for analysis. Tessellating these surfaces to create surface triangulations requires the use of adjacency information usually stored in the B-rep of the solid model to prevent gaps; tessellating the surfaces individually can lead to generation of small gaps along the edges of adjacent surface patches. This makes the tessellation process more tedious and requires specialized algorithms. In addition, since these surfaces are usually trimmed, additional surface quadrature points need to be evaluated close to the trim curves. In our method, we directly make use of the trimmed-NURBS surfaces without tessellating them. The surfaces are evaluated uniformly to the required level of precision and the parametric locations close to the trim curves are adaptively refined to include additional surface quadrature points as required.

Point membership classification (e.g., of the vertices of the background mesh) is traditionally performed using a ray-tracing approach (Rossignac and Requicha, 1999; Requicha and Voelcker, 1985). Any point inside the solid model intersects the surface of the model an odd number of times. Performing this operation directly on B-reps consisting of trimmed NURBS surfaces is a compute-intensive operation. Hence, in our method, we create a high-resolution voxelization of the CAD model using GPU rendering of trimmed-NURBS surfaces (Krishnamurthy et al., 2009, 2011). This voxelization is then used to perform point membership classification on vertices of the background mesh. The same operation is repeated to perform point membership classification on the volume quadrature points in the background mesh.

This paper is organized as follows. In Section 2, we briefly review the variational problem and solution strategies for immersogeometric analysis. In Section 3, we discuss the B-rep data structure and surface quadrature rules for B-rep-based immersogeometric analysis. In Section 4, we present a GPU-accelerated point membership classification technique for inside–outside test against B-rep models. In Section 5, we apply our proposed methods to the simulations of flow around a sphere as a benchmark study to investigate the influence of trimmed surfaces, and a full-scale agriculture tractor (Fig. 1) to demonstrate the ability of our B-rep-based immersogeometric method for the aerodynamic analysis of industrial-scale problems. In Section 6, we draw conclusions.

2. Immersogeometric analysis

In this section, we present the key features of immersogeometric fluid flow analysis. The components include a variational multiscale (VMS) formulation of incompressible flow (Hughes et al., 2000, 2001, 2004; Bazilevs et al., 2007a), the weakly enforced Dirichlet boundary conditions on the surface of the immersed objects (Bazilevs and Hughes, 2007; Xu et al., 2016), and the adaptively refined quadrature rules for faithfully capturing the geometry in intersected background elements. The last is critical for accurate immersogeometric fluid flow analysis, as shown in Xu et al. (2016).

2.1. Variational multiscale formulation

Let Ω (subsets of $\mathbb{R}^d$, $d \in \{2, 3\}$) denote the spatial domain and Γ be its boundary. Consider a collection of disjoint elements $\{\Omega^e\}$, $\cup_e \Omega^e \subset \mathbb{R}^d$, with closures covering the fluid domain: $\Omega \subset \cup_e \overline{\Omega^e}$. Note that Ω^e is not necessarily a subset of Ω. Let $\mathcal{V}_u^h$ and $\mathcal{V}_p^h$ be the discrete velocity and pressure spaces of functions supported on these elements. The VMS discretization of the Navier–Stokes equations of incompressible flows is stated as: Find fluid velocity $\mathbf{u}^h \in \mathcal{V}_u^h$ and pressure $p^h \in \mathcal{V}_p^h$ such that for all test functions $\mathbf{w}^h \in \mathcal{V}_u^h$ and $q^h \in \mathcal{V}_p^h$:

$$B^{\mathrm{VMS}}\left(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}\right) - F^{\mathrm{VMS}}\left(\{\mathbf{w}^h, q^h\}\right) = 0,$$

(1)

where

$$B^{\text{VMS}}\left(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}\right) = \int_\Omega \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h\right) d\Omega + \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right) d\Omega + \int_\Omega q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h d\Omega$$

$$- \sum_e \int_{\Omega^e \cap \Omega} \left(\mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h + \frac{\boldsymbol{\nabla} q^h}{\rho}\right) \cdot \mathbf{u}' d\Omega - \sum_e \int_{\Omega^e \cap \Omega} p' \boldsymbol{\nabla} \cdot \mathbf{w}^h d\Omega$$

$$+ \sum_e \int_{\Omega^e \cap \Omega} \mathbf{w}^h \cdot (\mathbf{u}' \cdot \boldsymbol{\nabla} \mathbf{u}^h) d\Omega - \sum_e \int_{\Omega^e \cap \Omega} \frac{\boldsymbol{\nabla} \mathbf{w}^h}{\rho} : (\mathbf{u}' \otimes \mathbf{u}') d\Omega$$

$$+ \sum_e \int_{\Omega^e \cap \Omega} \left(\mathbf{u}' \cdot \boldsymbol{\nabla} \mathbf{w}^h\right) \overline{\tau} \cdot \left(\mathbf{u}' \cdot \boldsymbol{\nabla} \mathbf{u}^h\right) d\Omega, \tag{2}$$

and

$$F^{\text{VMS}}\left(\{\mathbf{w}^h, q^h\}\right) = \int_\Omega \mathbf{w}^h \cdot \rho \, \mathbf{f} d\Omega + \int_{\Gamma^N} \mathbf{w}^h \cdot \mathbf{h} d\Gamma. \tag{3}$$

In (2), $\mathbf{u}'$ is defined as

$$\mathbf{u}' = -\tau_{\text{M}} \left(\rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}\right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right)\right) \tag{4}$$

and p' is given by

$$p' = -\rho \, \tau_{\text{C}} \boldsymbol{\nabla} \cdot \mathbf{u}^h. \tag{5}$$

In the above equations, ρ is the density of the fluid, $\mathbf{f}$ is the external force per unit mass, $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ are the stress and strain-rate tensors, respectively, and $\mathbf{h}$ is the traction vector at the Neumann boundary Γ^N. The terms integrated over element interiors may be interpreted both as stabilization and as a turbulence model (Brooks and Hughes, 1982; Tezduyar, 1992; Tezduyar and Osawa, 2000; Hughes et al., 2001; Bazilevs et al., 2007a; Hsu et al., 2010). τ_{M}, τ_{C} and $\overline{\tau}$ are the stabilization parameters. Their detailed expression used in this work can be found in Xu et al. (2016).

2.2. Variationally consistent weak boundary conditions

The standard way of imposing Dirichlet boundary conditions in Eq. (1) is to enforce them strongly by ensuring that they are satisfied by all trial solution functions. This is not feasible in immersed methods. Instead, the strong enforcement is replaced by weakly enforced Dirichlet boundary conditions proposed by Bazilevs et al. (Bazilevs and Hughes, 2007; Bazilevs et al., 2007b, 2010b). The semi-discrete problem becomes

$$B^{\text{VMS}}\left(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}\right) - F^{\text{VMS}}\left(\{\mathbf{w}^h, q^h\}\right)$$

$$- \int_{\Gamma^D} \mathbf{w}^h \cdot \left(-p^h \mathbf{n} + 2\mu \boldsymbol{\varepsilon}(\mathbf{u}^h) \mathbf{n}\right) d\Gamma - \int_{\Gamma^D} \left(2\mu \boldsymbol{\varepsilon}(\mathbf{w}^h) \mathbf{n} + q^h \mathbf{n}\right) \cdot \left(\mathbf{u}^h - \mathbf{g}\right) d\Gamma$$

$$- \int_{\Gamma^{D,-}} \mathbf{w}^h \cdot \rho \left(\mathbf{u}^h \cdot \mathbf{n}\right) \left(\mathbf{u}^h - \mathbf{g}\right) d\Gamma$$

$$+ \int_{\Gamma^D} \tau_{\text{TAN}}^B \left(\mathbf{w}^h - \left(\mathbf{w}^h \cdot \mathbf{n}\right) \mathbf{n}\right) \cdot \left(\left(\mathbf{u}^h - \mathbf{g}\right) - \left(\left(\mathbf{u}^h - \mathbf{g}\right) \cdot \mathbf{n}\right) \mathbf{n}\right) d\Gamma$$

$$+ \int_{\Gamma^D} \tau_{\text{NOR}}^B \left(\mathbf{w}^h \cdot \mathbf{n}\right) \left(\left(\mathbf{u}^h - \mathbf{g}\right) \cdot \mathbf{n}\right) d\Gamma = 0, \tag{6}$$

where Γ^D is the Dirichlet boundary that may cut through element interiors, $\Gamma^{D,-}$ is the *inflow* part of Γ^D, on which $\mathbf{u}^h \cdot \mathbf{n} < 0$, $\mathbf{g}$ is the prescribed velocity on Γ^D, τ_{TAN}^B and τ_{NOR}^B are stabilization parameters that need to be chosen element-wise as a compromise between the conditioning of the stiffness matrix, variational consistency, and the stability of the formulation.

For immersogeometric methods, weakly enforced boundary conditions are particularly attractive as the additional Nitsche terms (the third to last terms on the left-hand side of Eq. (6)) are formulated independently of the mesh. In contrast to
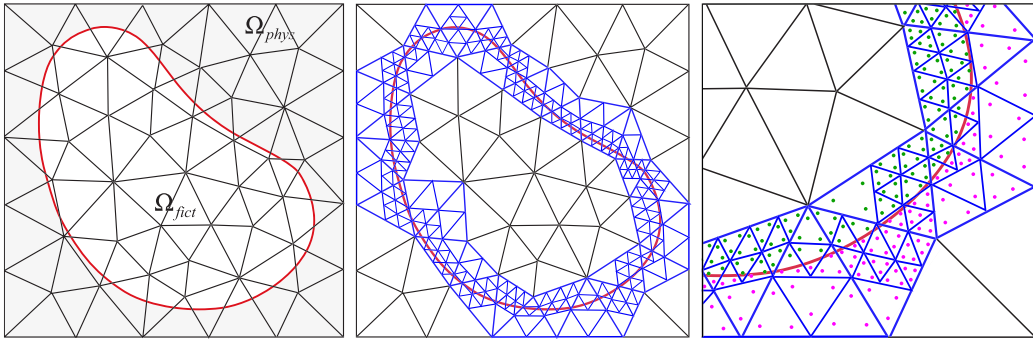
Fig. 2. The concept of physical and fictitious domains and quadrature scheme based on adaptive sub-cells (blue lines). Quadrature points within the fluid domain (marked in pink) are used in the numerical integration. Quadrature points outside (marked in green) are discarded. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

strong enforcement, which relies on boundary-fitted meshes to impose Dirichlet boundary conditions on the discrete solution space, the Nitsche terms also hold for intersected elements, where the domain boundary does not coincide with element boundaries. All that is needed is a separate discretization of the domain boundary with quadrature rules whose position of the quadrature points in intersected elements is known or can be determined. In Xu et al. (2016), the geometry of the object was described by the stereolithography (STL) format, which uses polygons (mostly triangles) to discretize (tessellate) the object surface. However, modern CAD models are typically described using B-rep, and the conversion from B-rep to STL is not trivial, especially when the geometry is not "watertight". In this work we tackle this issue by performing the surface integration of the weak boundary conditions directly using B-rep model information. This novel approach eliminates the need for a different discretization of the object surface and allows use of the actual CAD model directly for the purpose of immersogeometric analysis. This approach also shares the same philosophy with isogeometric analysis (Hughes et al., 2005; Cottrell et al., 2009)—bridging the gap between design and analysis.

Another advantage of weakly enforced Dirichlet boundary conditions is the release of the point-wise no-slip condition at the boundary of the fluid domain. Although maybe counterintuitive at first sight, some violation of the no-slip boundary condition is in fact desirable, as it allows the flow to slip on the solid surface and imitates the presence of the thin boundary layer that typically needs to be resolved with spatial refinement. It was shown in Bazilevs et al. (2010b) and Hsu et al. (2012) that weak boundary conditions allow for an accurate overall flow solution even if the mesh size in the wall-normal direction is relatively large. Weak enforcement of Dirichlet boundary conditions also provides special benefits in turbulent flow simulation (Bazilevs et al., 2007b, 2010b). In the immersogeometric method, the fluid mesh is arbitrarily cut by the object boundary, leaving a boundary layer discretization of inferior quality compared to the boundary-fitted counterpart. However, it was shown in Xu et al. (2016) that accurate laminar and turbulent flow solutions were obtained using the immersogeometric method with a mesh resolution and refinement pattern comparable to the boundary-fitted mesh used to obtain the reference values. We believe this is partially attributed to using the weak boundary condition formulation.

2.3. Sub-cell-based adaptive quadrature

The immersogeometric method introduces elements that are intersected by the geometric boundary, which creates complex, discontinuous integration domains in intersected elements. To ensure geometrically accurate evaluation of volume integrals in intersected elements, we use a sub-cell-based adaptive quadrature scheme (Düster et al., 2008; Xu et al., 2016). The basic concept is to increase the number of quadrature points around immersed geometric boundaries so that arbitrary integration domains resulting from the intersecting boundary can be taken into account accurately. This is achieved by recursively splitting intersected cells into sub-elements. At each level, only those sub-elements intersected by the boundary are further split. For clarity, we illustrate the quadrature scheme based on adaptive sub-cells for triangles in 2D in Fig. 2. We emphasize that splitting is performed on the quadrature level only and does not affect the basis functions, which are still defined on the original tetrahedral element.

2.4. Time integration and solution strategies

We complete the discretization of Eq. (6) by a time integration scheme from the family of generalized-α methods (Chung and Hulbert, 1993; Jansen et al., 2000; Bazilevs et al., 2008), which is a fully-implicit, second-order accurate method with control over the dissipation of high-frequency modes. At each time step, the solution of a nonlinear algebraic problem is required. We directly apply Newton–Raphson iterations (with an approximate tangent) to converge the residual of this algebraic problem. For each Newton–Raphson iteration, the linear system is solved using a block-diagonal preconditioned GMRES method (Saad and Schultz, 1986; Shakib et al., 1989). The computations reported in this work are carried out in a parallel computing environment on Linux clusters. A description of our parallelization strategy can be found in Hsu et al. (2011)
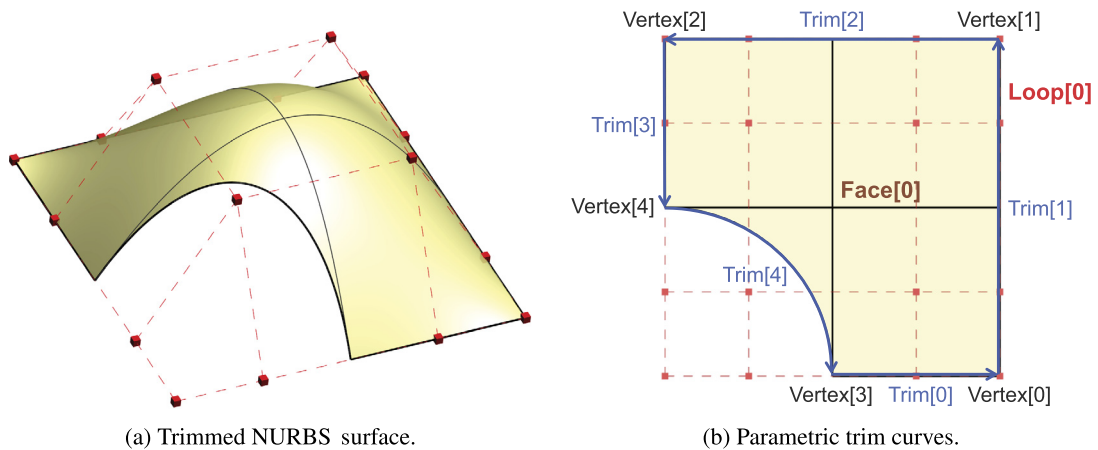
(a) Trimmed NURBS surface.   (b) Parametric trim curves.

Fig. 3. Example surface represented using trimmed NURBS. The trim curves are defined in the parametric space and, depending on the orientation of the curves, part of the NURBS surface is trimmed.

and a strong linear scaling of the immersogeometric method was shown in Xu et al. (2016). The mesh is partitioned into subdomains using METIS (Karypis and Kumar, 1999), and each subdomain is assigned to a processor core.

3. Boundary representation (B-rep)

In B-rep, the boundary of a CAD model is represented using a set of faces. The geometry of the faces is usually represented using tensor product NURBS surface patches. In addition, the B-rep data also includes information regarding the topology of the CAD model: the connectivity between the faces. This information is used to create watertight, 2-manifold, tessellations of the CAD model.

Representing B-rep faces using traditional NURBS surfaces restricts them to topologically rectangular sheets; they are not very flexible, especially when it comes to representing surfaces that are not rectangular or those with holes or complex local geometries that arise due to Boolean operations. Therefore, the NURBS patches are trimmed, discarding a portion of the surface defined in the parametric domain. An example of a trimmed NURBS surface in a CAD model is shown in Fig. 3(a). The trimming information is defined in the 2D parametric domain of the surface (Fig. 3(b)). Typically, trim curves are represented as directed closed loops; the direction of the loop determines which side of the trim curve to cut away. There can also be multiple loops per surface, one defining the boundary and others defining interior holes, or even holes within holes. In conventional CAD B-rep, there is at least one trim curve that bounds the valid surface region for every surface (even if is the trivial outer boundary) in order to have a consistent representation.

3.1. Surface quadrature in trimmed NURBS patches

The Dirichlet boundary conditions on the surface are imposed in the weak sense in our immersogeometric method. This requires the evaluation of the surface integrals of Eq. (6) on the object surface. These weak boundary conditions are evaluated at the surface Gaussian quadrature points.[1] For untrimmed NURBS surfaces, the surface quadrature points can be directly evaluated based on standard Gaussian quadrature rules. However, in the presence of trimmed NURBS surfaces, the surface quadrature points need to be checked if they lie inside the trim curves of the surface.[2] In addition, for accurate evaluation of the function near the trim curve, the surface needs to be adaptively sampled around the trim curve.

The adaptive quadrature for trimmed NURBS is implemented by first dividing the base untrimmed NURBS surface patches into quadrature elements for each knot span. The Gauss quadrature points for each quadrature element are generated in the parametric space. The parametric location of each Gauss point is then tested using the trim curves to classify them as inside or outside the trimmed-out section of the quadrature element. Those quadrature elements that have some Gauss points outside the trimmed-out section (inside the trimmed surface) are recursively refined using the sub-cell approach and the process is repeated with new Gauss quadrature points generated inside each refined quadrature element. This approach is illustrated in Fig. 4.

3.2. Implementation

We use Rhinoceros (Rhino) and Grasshopper to efficiently generate the surface quadrature rules directly from B-rep models. Grasshopper is a visual programming interface to Rhino. Using Grasshopper, the program is written in terms of

---

[1] We typically use $(p+1)$ Gaussian integration points per parametric direction for each knot span. p is the degree of the NURBS.

[2] The regions inside and outside the trim curves are called the *trimmed surface* and *trimmed-out section*, respectively.
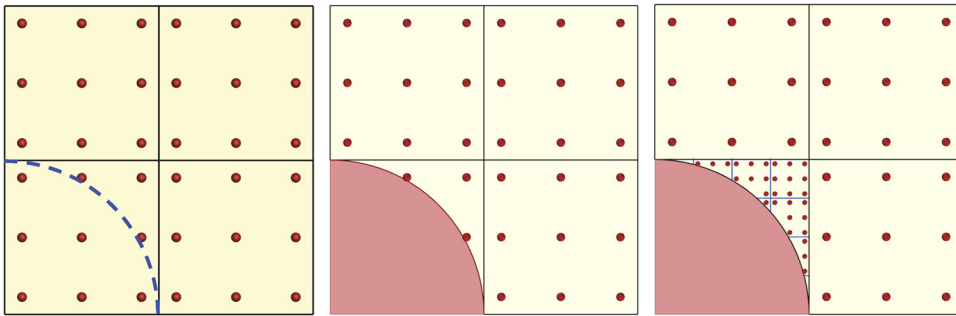
Fig. 4. Sub-cell-based adaptive quadrature is used to more accurately evaluate the integrals near the trim curves. In each level, the Gaussian quadrature points (shown in red) that lie outside the trim curves are discarded. The Gaussian quadrature points after two levels of adaptive quadrature sub-cell refinement are shown on the right. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
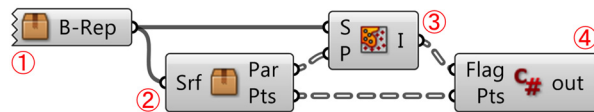

Fig. 5. Grasshopper implementation for generating surface quadrature rules directly from B-rep models.

"components" with pre-defined or user-defined functionality and "wire connections" between the components that serve as conduits of input and output data. The large selection of pre-defined components gives users access to complex geometric modeling functionality. A Grasshopper implementation for generating Gaussian quadrature rules on untrimmed and trimmed patches is shown in Fig. 5. The visual program executes the following steps:

- Component 1 reads in the NURBS-based B-rep model. Note that the orientation of the surfaces should be facing towards the interior of the object (so that the normal vector will be pointing outward of the fluid domain).
- Component 2 takes the B-rep surfaces from 1 (Srf), builds quadrature elements on each untrimmed NURBS patch, adaptively refines quadrature elements near trim curves, and applies standard Gaussian quadrature rules on each quadrature element. The component then calculates the physical location, Gauss weight, Jacobian determinant of the parameter-to-physical-space mapping, and the normal vector on each Gauss point (Pts), and finds the location of each Gaussian quadrature point in the parametric space (Par).
- Component 3, which is a Grasshopper built-in function called *Point in Trim*, takes the B-rep data from 1 (S) and parametric location of the Gaussian quadrature points from 2 (P), and determines whether the points are inside or outside the trim curves using the input B-rep data. A flag will be assigned to each point as a result of the inside–outside test. Using this pre-defined component, the users do not need to know the details of the B-rep data structure.
- Component 4 takes the Gaussian quadrature rules from 2 (Pts) and the corresponding inside–outside flags from 3, then only outputs the surface quadrature points *inside* the trim curves with the required information for immersogeometric analysis. The output data on each Gauss point includes its physical location (used to locate the point in the background mesh), Gauss weight, Jacobian determinant of the mapping, and the normal vector (pointing towards the interior of the object). This information allows us to integrate over the trimmed surfaces of the B-rep CAD model.

4. GPU-accelerated point membership classification

To perform sub-cell-based adaptive quadrature (Section 2.3) for immersogeometric analysis, the mesh vertices of the background mesh as well as the 3D quadrature points need to be classified as being inside or outside the immersed object. We perform this operation by creating a finely sampled voxelization of the CAD object as a preprocessing step. We then consider these voxels as individual axis-aligned bounding-boxes (AABBs), which can then be used to perform the point membership classification during immersogeometric analysis.

To create the voxelized representation of the CAD model, we construct a grid of voxels in the region occupied by the object. We then make use of a rendering-based approach to classify the voxel centers as being inside or outside the object. Using this method on the GPU, a fine voxelization of the model (up to 1 billion voxels) with a relative voxel size of 0.001, can be generated (Fig. 6). This resolution is fine enough to perform immersogeometric point membership classification for standard CAD models.

A 2D example of the method is shown in Fig. 7; the method directly extends to 3D. The CAD model is rendered slice-by-slice by clipping it while rendering. Each pixel of this clipped model is then used to classify the voxel corresponding to the slice as being inside or outside the CAD model. This is performed by counting the number of fragments that were rendered in each pixel using the stencil buffer on the GPU. After the clipped model has been rendered, an odd value in the stencil
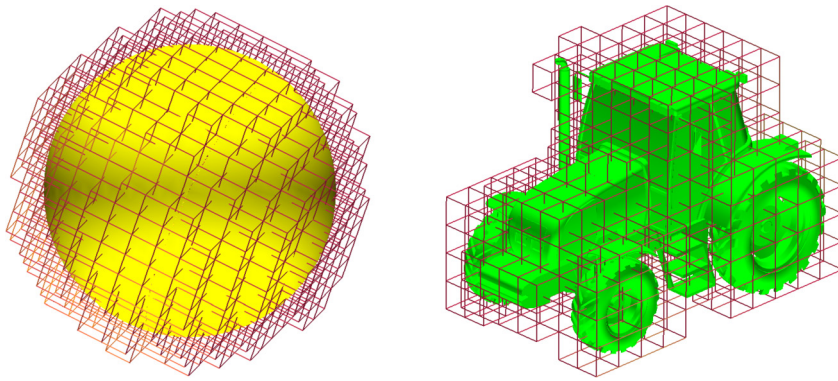
Fig. 6. Visualization of coarse voxelization of CAD models. A fine voxelization is used for point membership classification.
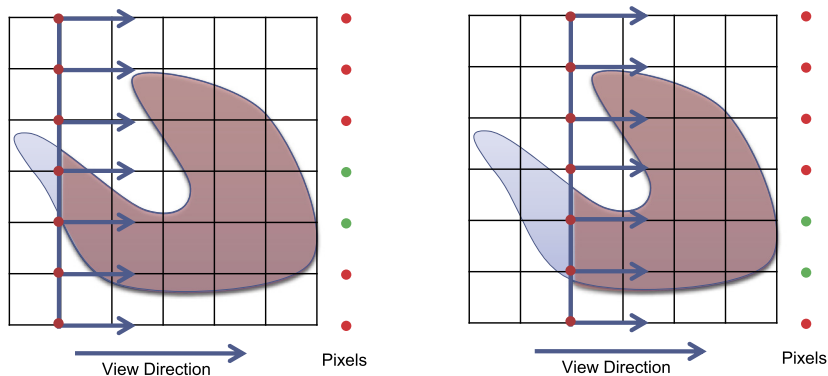

Fig. 7. Performing point membership classification in 2D using GPU rendering. A clipped CAD model is rendered slice-by-slice and the number of rendered pixels is counted. The pixels that are rendered an odd number of times in each slice are inside the object.

buffer indicates that the voxel on the particular slice is inside the CAD model, and vice versa. The process is then repeated by clipping the model with a plane that is offset by the voxel size. Once all the slices have been classified, we can get the complete voxelized representation of the CAD model.

We make use of the method developed by Krishnamurthy et al. (2009) to directly evaluate and render the NURBS surfaces in the model using the GPU. The B-rep model is first decomposed into its component surfaces. If the surface is a flat or a cylindrical surface, it is converted into triangles with a very fine resolution that is less than one-tenth of the voxelization resolution. All other surfaces are converted into NURBS and are evaluated using the GPU while rendering. Each surface is rendered successively to an off-screen framebuffer and the stencil buffer is used to classify the voxels corresponding to the slice as being inside or outside as explained above. Hence the model is rendered once for each slice to classify the voxelization.

The time taken to perform the classification is the sum of the time taken to evaluate the NURBS surfaces in the model once and the total time taken to render each slice. As an example, the total time taken to evaluate the NURBS surfaces in the tractor model considered in Section 5.2 is 3.80 seconds, while the time taken to classify 558,458,880 ($1120 \times 784 \times 636$) voxels is 22.93 seconds. These timings are obtained by running our classification algorithm on a MacBook Pro with a 2.7 GHz processor, 16 GB RAM, and an NVIDIA GeForce GT 650M GPU.

5. B-rep-based immersogeometric analysis

In this section, we apply our B-rep-based immersogeometric method to the simulations of flow around bluff bodies. We first consider simple B-rep models of a sphere as a benchmark study to investigate the influence of trimmed NURBS surfaces on the immersogeometric fluid flow analysis. We then use a native B-rep CAD model of a full-scale agriculture tractor to demonstrate the accuracy and efficiency of our B-rep-based immersogeometric approach for the analysis of industrial-scale ground vehicle aerodynamics.

5.1. Flow around a sphere

We use the flow around a sphere at Reynolds numbers $Re = 100$ as a benchmark to assess the accuracy of our B-rep-based immersogeometric method. The problem setup, computational domain, and boundary conditions are identical to those
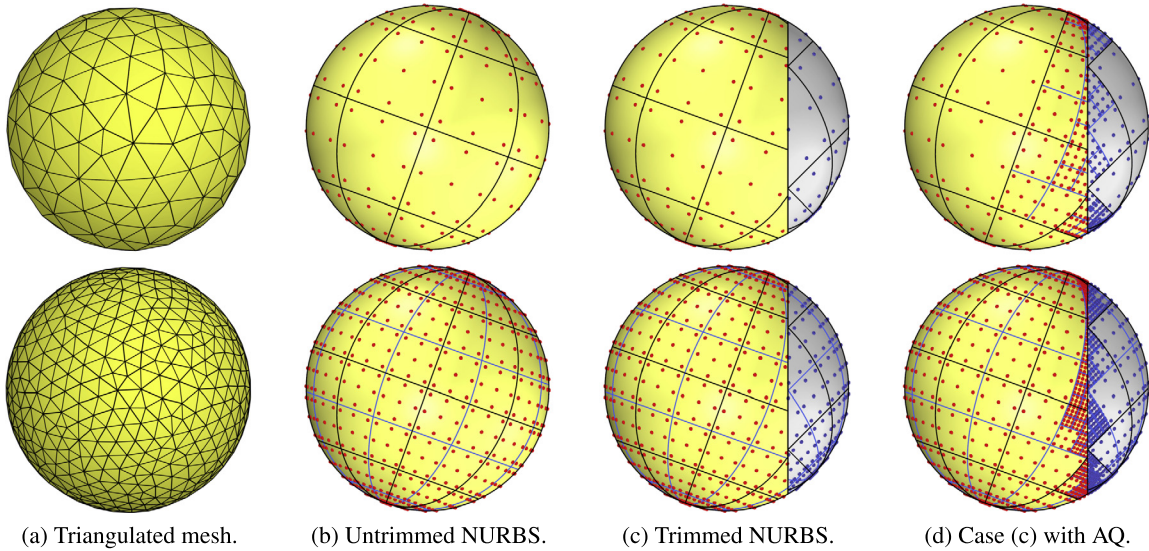
(a) Triangulated mesh.   (b) Untrimmed NURBS.   (c) Trimmed NURBS.   (d) Case (c) with AQ.

Fig. 8. (a)–(c) The three different B-rep models of a sphere considered in this study. The first row is the Level 1 quadrature element level and the second row is Level 2 quadrature element refinement. The triangulated mesh sizes are 0.16 and 0.08 and three quadrature points are used for each linear triangle. Nine quadrature points (three in each parametric direction) are used for each untrimmed quadrature element on the NURBS surface. (d) Addition of two levels of adaptive quadrature (AQ) sub-cells to the quadrature elements near the trim curves.

of used in Xu et al. (2016). The background fluid mesh into which the sphere is immersed is discretized using linear tetrahedral elements and is identical to the "IM1" used in Xu et al. (2016). The main advantage of using tetrahedral elements is their flexibility in providing locally refined three-dimensional discretizations for complex domains. Local refinement is essential in this work for boundary layer resolution.

Remark 1. It should be noted that the immersogeometric concept is independent of a specific basis and can be used with any basis function technology and element type. For example, Hsu et al. (2014, 2015a) successfully immersed NURBS and T-spline geometries into trivariate NURBS background mesh and Xu et al. (2016) immersed a triangulated surface into tetrahedral background elements.

Sub-cell-based adaptive quadrature was used in Kamensky et al. (2015) and Xu et al. (2016) to accurately integrate the *volume* integrals and faithfully capture the geometry in background intersected elements. This is a key requirement for obtaining accurate flow solutions with the immersogeometric method. Xu et al. (2016) showed that two levels of adaptive quadrature sub-cells in background intersected elements present a good balance between computational cost and solution accuracy. In this work we adopt the same strategy and use two levels of adaptive quadrature sub-cells for all our computations.

The weak boundary conditions formulated in Eq. (6) require that we evaluate surface integrals of traces of functions defined on the background mesh. This is done by breaking the immersed surface into quadrature elements independently of the background mesh, and defining a quadrature rule on these elements. The surface element quadrature points must then be located in the parameter space of the background elements in which they fall. Kamensky et al. (2015) suggested that even though the surface parameterization need not be informed by the background fluid discretization, the physical space density of surface quadrature points should be reasonably high with respect to the fluid element size. Xu et al. (2016) used a surface element size half of the background element size near the immersed object boundary to ensure sufficient resolution.

Fig. 8(a)–(c) shows the three different B-rep models of a sphere considered in this study. These models will be directly used to generate the surface quadrature rules without using a different discretization. In the first column, the surface of the sphere is tessellated with linear triangles and the geometry accuracy depends on the surface element size. The second column shows the sphere modeled using one untrimmed quadratic NURBS patch. Note that the sphere geometry is exact at the coarsest parametric level. The third column shows the sphere modeled using two trimmed quadratic NURBS surfaces. This allows us to study the effect of using trimmed patches in immersogeometric flow analysis.

To investigate the required surface quadrature density for the flow around a sphere problem, we considered different levels of quadrature element refinement. The first row in Fig. 8(a)–(c) shows the coarsest quadrature level (Level 1) considered for each B-rep model. The cases shown in the second row are after one level of quadrature element refinement (Level 2). For the tessellated surface, the quadrature element refinement requires remeshing the surface using finer triangles to improve the geometry resolution. However, for the NURBS models, since the geometry is exact at the coarsest level, we refine the quadrature elements using a sub-cell approach without further discretization. At each new level, a quadrature element is
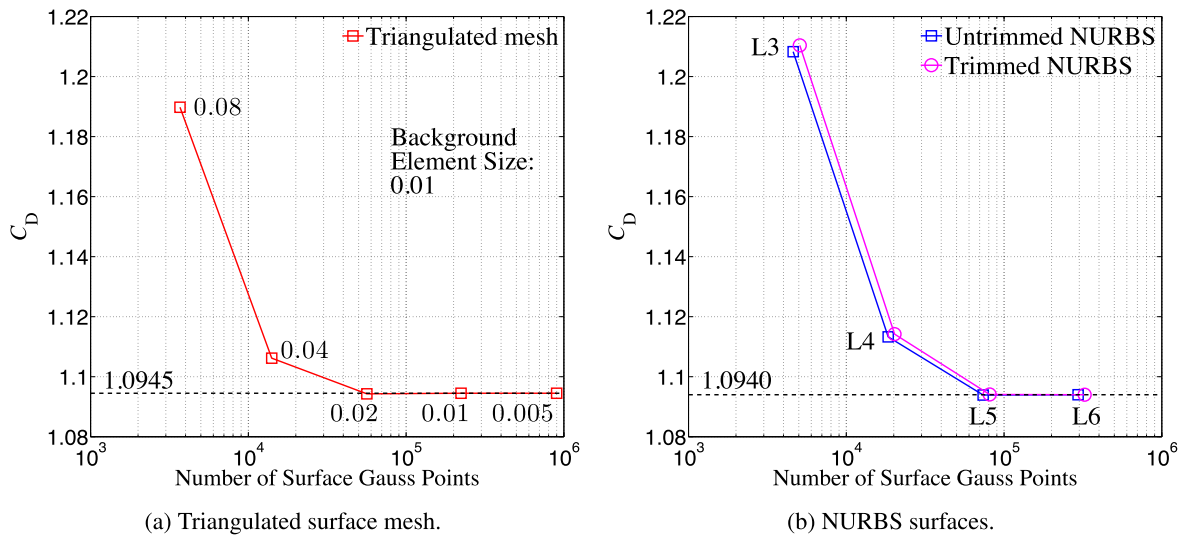
Fig. 9. Drag coefficient convergence associated with surface quadrature element densities. The triangulated surfaces cover mesh sizes from 0.08 to 0.005. The drag coefficients converge to 1.0945. The untrimmed and trimmed NURBS surface cases considered include Level 3 to Level 6 quadrature element refinements. The drag coefficients converge to 1.0940.

split into four sub-cells, each assigned as the new quadrature elements. Standard Gaussian quadrature rule is applied to the quadrature elements. This approach is consistent and can be combined with the adaptive quadrature presented in Section 3.1. We first uniformly refine the quadrature elements on the entire patch, i.e., we apply the same quadrature element refinement level to all of the NURBS elements (defined by knot spans). This is due to the necessity of having sufficient surface quadrature point density for immersogeometric analysis. One can then adaptively refine the quadrature elements near the trim curves for improving the surface integration accuracy lost due to trimming. Fig. 8(d) shows the Level 1 and Level 2 quadrature elements of the trimmed NURBS sphere with two levels of adaptive quadrature sub-cells applied near the trim curves.

Remark 2. For multi-patch scenarios, the refinement level can be different for each patch based on the physical patch size to ensure similar Gauss point distribution over the entire B-rep model. Determining the refinement level based on the patch size was chosen due to its simplicity of implementation. A more sophisticated approach is to choose different refinement levels in each parametric direction based on the physical length of the knot span (element size). This approach may produce more evenly distributed quadrature points over the entire patch surface. We plan to investigate this in the future.

Fig. 9 shows the drag coefficient C_D for flow around a sphere at $Re = 100$, computed with our immersogeometric method using different B-rep models of the sphere, each with several levels of surface quadrature refinement. The drag force was evaluated using the variationally consistent conservative definition of traction (Bazilevs and Akkerman, 2010; van Brummelen et al., 2011; Xu et al., 2016). The triangulated surfaces cover mesh sizes from 0.08 to 0.005. The results show that taking more levels of surface quadrature element refinement converges the C_D to 1.0945, which is consistent with the value reported in Xu et al. (2016). Fig. 9(a) also shows that choosing a similar element size between the immersed surface and the background mesh may be sufficient when a triangulated surface mesh and tetrahedral background elements are used. For the untrimmed and trimmed NURBS spheres, Level 3 to Level 6 quadrature element refinements without adaptivity near the trim curves are considered. Fig. 9(b) shows that even though the drag coefficients between untrimmed and trimmed NURBS are slightly different at Level 3 and Level 4, this discrepancy disappears at Level 5 and Level 6. The drag coefficients for both spheres converge to 1.0940.

Comparing Fig. 9(a) and (b), the C_D convergence curves show a very similar trend when comparable numbers of surface quadrature points are used. In fact, the triangulated surface converges faster than the NURBS surfaces. We believe this is because triangulated sphere has more evenly distributed surface elements while, due to the degenerate points and surface curvature, NURBS spheres have larger size variation between elements. The results show that the immersogeometric method is sensitive to the density and distribution of the surface quadrature points in the background mesh. It may be more efficient to refine the quadrature elements based on their physical element size. The comparison between Fig. 9(a) and (b) also reveals that when sufficiently dense surface quadrature points are used, the converged drag coefficients are slightly different when triangulated or NURBS surfaces are used. Since the background meshes are identical in all cases, this difference is likely caused by the geometric error when triangular tessellation of the curved surfaces is used (note that the NURBS geometry is exact). We plan to investigate this in the future.

Using denser and better distributed surface quadrature points increases the accuracy of the surface integration in the background mesh, which directly links to the accurate evaluation of the weak enforcement of Dirichlet boundary conditions.
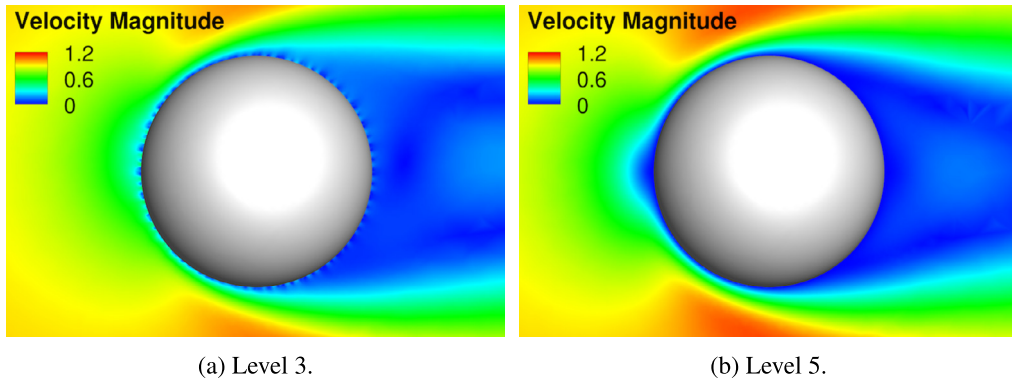
(a) Level 3.   (b) Level 5.

Fig. 10. Velocity magnitude contours of the immersogeometric result computed using untrimmed NURBS B-rep model with three and five levels of surface quadrature element refinement.
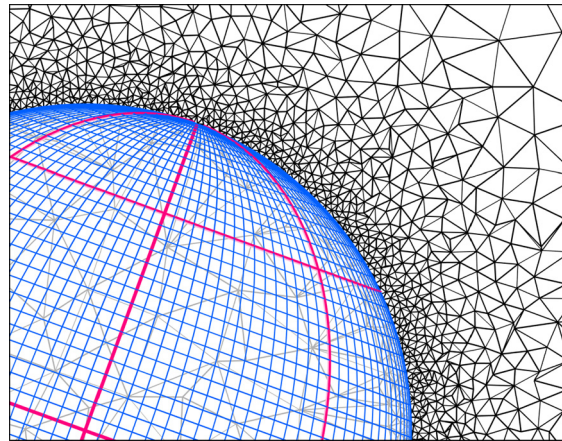

Fig. 11. NURBS surface parameterization (pink) of the untrimmed sphere, Level 5 surface quadrature elements (blue), and a planar cut of the background tetrahedral elements (black) into which the sphere is immersed. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

This can be seen in Fig. 10, which shows the velocity magnitude contour of the immersogeometric results computed using untrimmed NURBS B-rep model with three and five levels of surface quadrature element refinement. Fig. 9(b) shows that Level 5 produces a correct prediction of C_D while Level 3 presents a large error. It can be clearly seen from Fig. 10(a) that due to the lack of surface quadrature point resolution in the background fluid mesh, the (weakly enforced) Dirichlet boundary conditions are satisfied poorly and the flow solutions near the boundary layer are unstable. Fig. 10(b) shows the correct flow field obtained when sufficiently fine surface quadrature points are used. We conclude that an increased surface quadrature density is crucial to achieving accurate flow solutions in the immersogeometric analysis.

Remark 3. An example of the surface element resolution relative to the background fluid mesh is shown in Fig. 11. The figure shows the NURBS surface parameterization (pink) of the untrimmed sphere, Level 5 surface quadrature elements (blue), and a planar cut of the background tetrahedral elements into which the sphere is immersed. The background element size near the sphere boundary is 0.01.

The comparison between untrimmed and trimmed NURBS results in Fig. 9(b) suggests that with the sufficiently fine surface quadrature point density to obtain accurate flow solution, the quadrature error introduced by trimming may be insignificant. To verify this, we add 1, 2 and 4 levels of adaptive quadrature sub-cells near the trim curves to improve the surface integration accuracy lost due to trimming. The results are shown in Fig. 12. At Level 4, it can be clearly seen that using adaptive quadrature does influence the results. In fact, the drag coefficient at Level 4 converges under the refinement of adaptive quadrature sub-cells, but towards a value worse than the unrefined case. This may seem counterintuitive at first, but the result implies that the unrefined case was "inaccurate" due to the surface quadrature error caused by the trimming. The "correct" drag coefficient at this particular level for the trimmed NURBS sphere is the one obtained using more accurate surface integration, improved by the adaptive quadrature sub-cells.

At Level 5, when the drag coefficients are converged, Fig. 12 shows that adding surface adaptive quadrature sub-cells near the trim curves does not change the results. This confirms that with the necessary density and distribution of surface
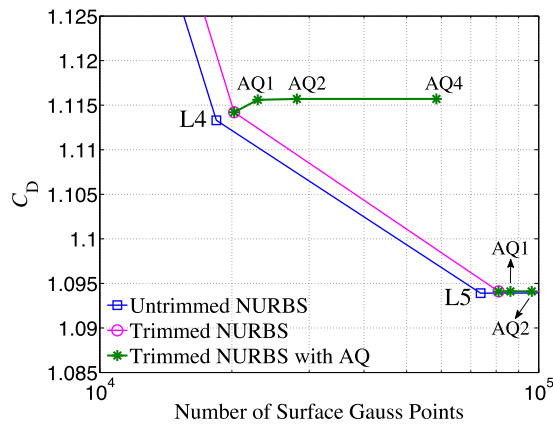
Fig. 12. Drag coefficient predicted with our immersogeometric method using Level 4 and Level 5 quadrature elements with 1, 2 and 4 levels of adaptive quadrature (AQ) sub-cells added near the trim curves.
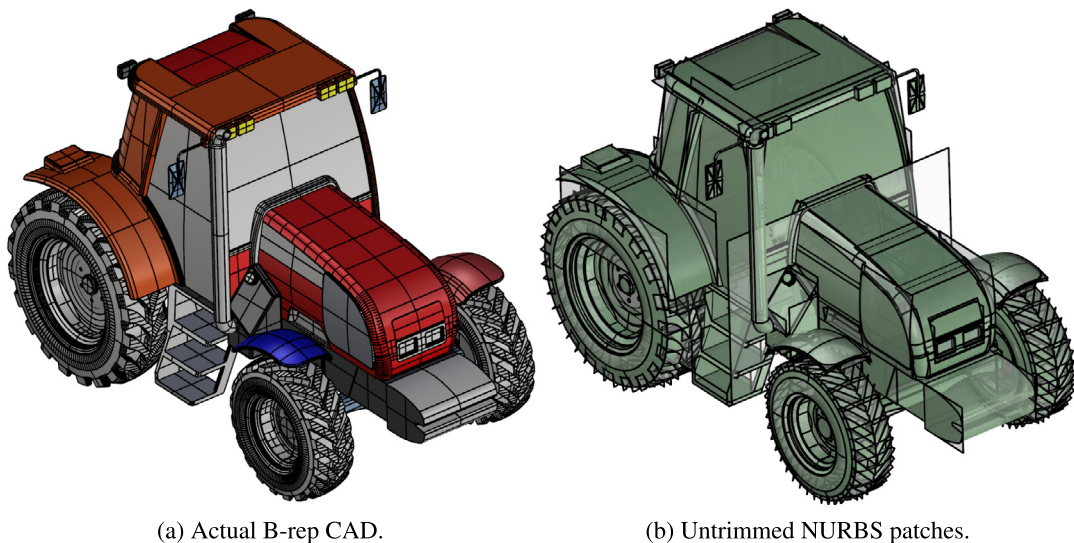

(a) Actual B-rep CAD.   (b) Untrimmed NURBS patches.

Fig. 13. B-rep CAD of an agricultural tractor model. The complexity of the design can be seen.

quadrature points for accurate weak enforcement of Dirichlet boundary conditions, the quadrature error near the trim curves is negligible. As a result, it may not be necessary to use the adaptive quadrature to improve the surface integration accuracy lost due to trimming, in the context of immersogeometric fluid flow analysis. Using adaptive quadrature near the trim curves does improve the accuracy of the surface integration. However, it also increases the evaluation time. Since using the required level of surface quadrature element refinement already generates an accurate prediction of the drag coefficient, we therefore do not use adaptive quadrature on the surface in favor of the better efficiency.

Finally, the study presented in this section shows that B-rep CAD models, with surface quadrature rules generated using the method proposed in Section 3.2, are successfully and directly employed in our immersogeometric fluid flow analysis.

5.2. Airflow around a tractor

Typical vehicle designs lead to very complex fluid domain boundaries due to the detailed geometric features of the models. This complexity constitutes a major barrier to the transfer of fluid domains into boundary-fitted computational meshes. An example is the B-rep model of the tractor shown in Fig. 13(a), which incorporates many geometrically complex details (e.g., tires). A common practice is to preprocess the B-rep by tessellating it into triangles, and then using the triangular surface mesh for CFD mesh generation. However, generating the surface tessellations of complex CAD models is time-consuming and labor intensive, since the geometry needs to be manually checked to avoid creating any intersecting or non-manifold features during tessellation. Furthermore, the CFD mesh quality heavily depends on the surface mesh design, and, as a result, human analysts are required to perform intermediate steps such as defeaturing, geometry cleanup, and mesh manipulation (Marcum and Gaither, 2000; Wang and Srinivasan, 2002; Beall et al., 2004; Lee et al., 2010).
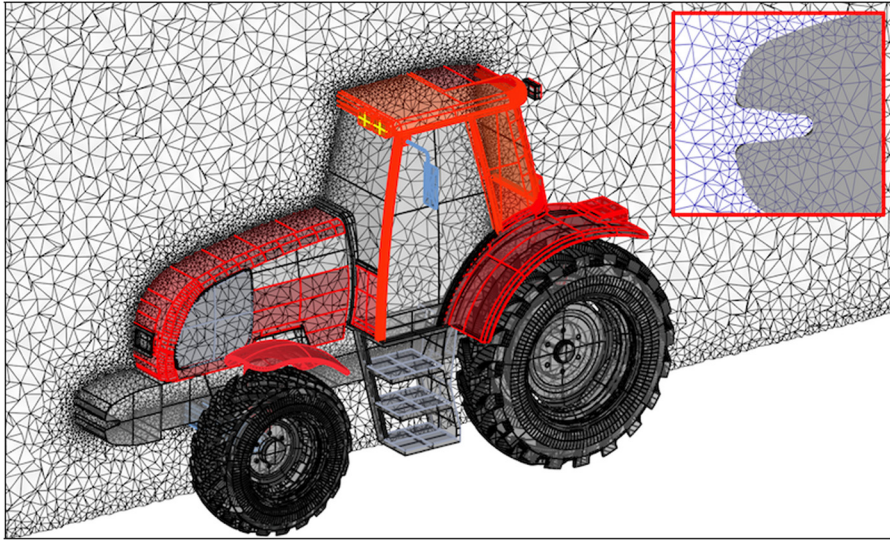
Fig. 14. Locally refined tetrahedral meshes of the fluid domain for aerodynamic analysis of the tractor. We show the mesh cut along a plane in flow direction. The tractor model is represented directly using B-rep.

The immersogeometric method was proposed to eliminate these labor-intensive procedures from the CFD simulation pipeline while still maintaining high accuracy of the simulation results. In this section, we demonstrate how the tractor design based on B-rep can be directly immersed into a non-boundary-fitted discretization of the surrounding fluid domain and perform aerodynamic analysis in the context of large-scale industrial applications.

In Xu et al. (2016), the tractor geometry was tessellated and described by STL format. However, conversion from B-rep to STL is not trivial, especially when the geometry is not "watertight". Here we perform the surface integration of the weak boundary conditions directly using B-rep model information. In Section 3, we discussed how B-rep involves trimmed NURBS surfaces. Fig. 13(b) shows the untrimmed NURBS patches used to model the tractor. We use our approach proposed in Section 3.2 to directly access the B-rep data and trimming information and generate a proper surface quadrature rule for the actual tractor design (Fig. 13(a)).

The problem setup, computational domain, boundary conditions, and background fluid mesh are identical to those of used in Xu et al. (2016). A zoom of the immersogeometric mesh is shown in Fig. 14. The B-rep model of the tractor is directly immersed into the tetrahedral background mesh. Two levels of adaptive quadrature sub-cells are added in all intersected background elements to accurately integrate the volume integrals. Sufficiently accurate integration in intersected elements is essential to faithfully capture the geometry of the tractor. The inside–outside classification of the volume quadrature points are carried out using the GPU-accelerated point membership classification technique presented in Section 4.

We note that a significant advantage of the immersogeometric method is its geometric flexibility. For example, it enables us to impose a uniform mesh size along the immersed tractor surface regardless of fine-scale geometric features. It is not necessary to "defeature" those geometrically complex objects. Taking the tractor model as an example, the tires are typically smoothed due to the challenges associated with boundary-layer mesh generation. In the immersogeometric approach, this geometry manipulation is not necessary and the fluid mesh can be generated directly.

Fig. 1 shows the immersogeometric result of the instantaneous vortical structures of the highly turbulent flow around the tractor. To assess the accuracy of the B-rep-based immersogeometric results, we first compute the time-averaged drag coefficient $\overline{C}_D = 2\overline{F}_D / \rho U^2 A$, where U is the inflow velocity, $\overline{F}_D$ is the time-averaged drag force, and A is the area of the frontal tractor surface projected onto a plane perpendicular to the main flow direction. We compare our simulation result to the reference values reported by Xu et al. (2016), where a triangulated tractor surface was used for the computation. The values of $\overline{C}_D$ are 0.851 and 0.864 for the triangulated-surface and NURBS B-rep computations, respectively. The results are in very good agreement between the two models. We also plot the distribution of the time-averaged pressure coefficient $\overline{C}_P$ along curves over the tractor top and bottom surfaces in Fig. 15. An overall good agreement is again observed, which shows the effectiveness and accuracy of our direct immersogeometric fluid flow analysis using B-rep CAD models for complex-geometry problems.

Finally, the tractor analysis presented in this section indicates that our B-rep-based immersogeometric method can greatly simplify the mesh generation process for industrial turbulent flow problems without sacrificing solution accuracy.

6. Conclusions

In this paper, we presented a new method for immersogeometric fluid flow analysis that directly makes use of the CAD B-rep of a complex object and immerses it into a locally refined, non-boundary-fitted background fluid mesh to avoid the
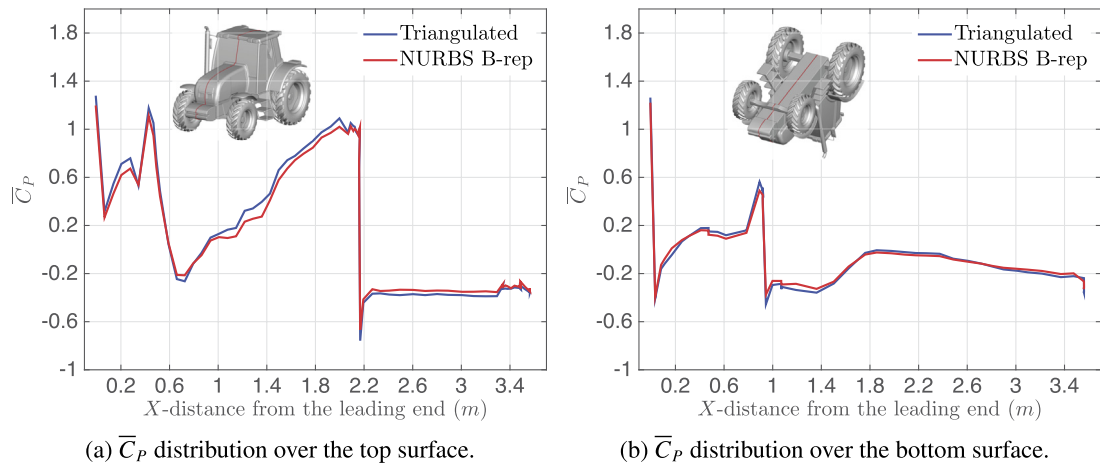
(a) $\overline{C}_P$ distribution over the top surface.

(b) $\overline{C}_P$ distribution over the bottom surface.

Fig. 15. Time-averaged pressure coefficient $\overline{C}_P$ plotted along center curves over the tractor surface.

challenges associated with geometry cleanup and mesh generation and manipulation. To assess the accuracy of the proposed method, we performed computations of the benchmark problem of flow over a sphere represented using B-rep. Quantities of interest such as drag coefficient were in good agreement with reference values reported in the literature. The results showed that the density and distribution of the surface quadrature points are crucial for accurate high-fidelity flow analysis. Also, with sufficient levels of surface quadrature element refinement, the quadrature error near the trim curves became insignificant. Finally, we demonstrated the effectiveness of our immersogeometric method for industrial scale simulations by performing an aerodynamic analysis of an agricultural tractor directly represented using B-rep.

Acknowledgements

References

Akkerman, I., Bazilevs, Y., Calo, V.M., Hughes, T.J.R., Hulshoff, S., 2008. The role of continuity in residual-based variational multiscale modeling of turbulence. Comput. Mech. 41, 371–378.

Akkerman, I., Bazilevs, Y., Kees, C.E., Farthing, M.W., 2011. Isogeometric analysis of free-surface flow. J. Comput. Phys. 230, 4137–4152.

Auricchio, F., Beirão da Veiga, L., Hughes, T.J.R., Reali, A., Sangalli, G., 2010. Isogeometric collocation methods. Math. Models Methods Appl. Sci. 20, 2075–2107.

Auricchio, F., Calabrò, F., Hughes, T.J.R., Reali, A., Sangalli, G., 2012. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. Comput. Methods Appl. Mech. Eng. 249–252, 15–27.

Bazilevs, Y., Akkerman, I., 2010. Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual-based variational multiscale method. J. Comput. Phys. 229, 3402–3414.

Bazilevs, Y., Calo, V.M., Cottrel, J.A., Hughes, T.J.R., Reali, A., Scovazzi, G., 2007a. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. Comput. Methods Appl. Mech. Eng. 197, 173–201.

Bazilevs, Y., Calo, V.M., Cottrell, J.A., Evans, J.A., Hughes, T.J.R., Lipton, S., Scott, M.A., Sederberg, T.W., 2010a. Isogeometric analysis using T-splines. Comput. Methods Appl. Mech. Eng. 199, 229–263.

Bazilevs, Y., Calo, V.M., Hughes, T.J.R., Zhang, Y., 2008. Isogeometric fluid–structure interaction: theory, algorithms, and computations. Comput. Mech. 43, 3–37.

Bazilevs, Y., Hsu, M.C., Scott, M.A., 2012. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. Comput. Methods Appl. Mech. Eng. 249–252, 28–41.

Bazilevs, Y., Hughes, T.J.R., 2007. Weak imposition of Dirichlet boundary conditions in fluid mechanics. Comput. Fluids 36, 12–26.

Bazilevs, Y., Michler, C., Calo, V.M., Hughes, T.J.R., 2007b. Weak Dirichlet boundary conditions for wall-bounded turbulent flows. Comput. Methods Appl. Mech. Eng. 196, 4853–4862.

Bazilevs, Y., Michler, C., Calo, V.M., Hughes, T.J.R., 2010b. Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. Comput. Methods Appl. Mech. Eng. 199, 780–790.

Beall, M.W., Walsh, J., Shephard, M.S., 2004. A comparison of techniques for geometry access related to mesh generation. Eng. Comput. 20, 210–221.

Benson, D.J., Bazilevs, Y., De Luycker, E., Hsu, M.C., Scott, M., Hughes, T.J.R., Belytschko, T., 2010a. A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM. Int. J. Numer. Methods Eng. 83, 765–785.

Benson, D.J., Bazilevs, Y., Hsu, M.C., Hughes, T.J.R., 2010b. Isogeometric shell analysis: the Reissner–Mindlin shell. Comput. Methods Appl. Mech. Eng. 199, 276–289.

Borden, M.J., Hughes, T.J.R., Landis, C.M., Verhoosel, C.V., 2014. A higher-order phase-field model for brittle fracture: formulation and analysis within the isogeometric analysis framework. Comput. Methods Appl. Mech. Eng. 273, 100–118.

Borden, M.J., Scott, M.A., Evans, J.A., Hughes, T.J.R., 2011. Isogeometric finite element data structures based on Bézier extraction of NURBS. Int. J. Numer. Methods Eng. 87, 15–47.

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, 401–457.

Brooks, A.N., Hughes, T.J.R., 1982. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. Comput. Methods Appl. Mech. Eng. 32, 199–259.

van Brummelen, E.H., van der Zee, K.G., Garg, V.V., Prudhomme, S., 2011. Flux evaluation in primal and dual boundary-coupled problems. J. Appl. Mech. 79, 010904.

Chung, J., Hulbert, G.M., 1993. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. J. Appl. Mech. 60, 371–375.

Cottrell, J.A., Hughes, T.J.R., Bazilevs, Y., 2009. Isogeometric Analysis: Toward Integration of CAD and FEA. John Wiley & Sons, Chichester.

Cottrell, J.A., Hughes, T.J.R., Reali, A., 2007. Studies of refinement and continuity in isogeometric structural analysis. Comput. Methods Appl. Mech. Eng. 196, 4160–4183.

Cottrell, J.A., Reali, A., Bazilevs, Y., Hughes, T.J.R., 2006. Isogeometric analysis of structural vibrations. Comput. Methods Appl. Mech. Eng. 195, 5257–5297.

De Lorenzis, L., Temizer, İ., Wriggers, P., Zavarise, G., 2011. A large deformation frictional contact formulation using NURBS-based isogeometric analysis. Int. J. Numer. Methods Eng. 87, 1278–1300.

Dedè, L., Borden, M.J., Hughes, T.J.R., 2012. Isogeometric analysis for topology optimization with a phase field model. Arch. Comput. Methods Eng. 19, 427–465.

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, 3768–3782.

Evans, J.A., Hughes, T.J.R., 2013. Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations. J. Comput. Phys. 241, 141–167.

Gomez, H., Calo, V.M., Bazilevs, Y., Hughes, T.J.R., 2008. Isogeometric analysis of the Cahn–Hilliard phase-field model. Comput. Methods Appl. Mech. Eng. 197, 4333–4352.

Grasshopper, 2016. http://www.grasshopper3d.com/. Accessed 5 February 2016.

Guo, Y., Ruess, M., 2015. Nitsche's method for a coupling of isogeometric thin shells and blended shell structures. Comput. Methods Appl. Mech. Eng. 284, 881–905.

Hsu, M.C., Akkerman, I., Bazilevs, Y., 2011. High-performance computing of wind turbine aerodynamics using isogeometric analysis. Comput. Fluids 49, 93–100.

Hsu, M.C., Akkerman, I., Bazilevs, Y., 2012. Wind turbine aerodynamics using ALE–VMS: validation and the role of weakly enforced boundary conditions. Comput. Mech. 50, 499–511.

Hsu, M.C., Bazilevs, Y., Calo, V.M., Tezduyar, T.E., Hughes, T.J.R., 2010. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. Comput. Methods Appl. Mech. Eng. 199, 828–840.

Hsu, M.C., Kamensky, D., Bazilevs, Y., Sacks, M.S., Hughes, T.J.R., 2014. Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. Comput. Mech. 54, 1055–1071.

Hsu, M.C., Kamensky, D., Xu, F., Kiendl, J., Wang, C., Wu, M.C.H., Mineroff, J., Reali, A., Bazilevs, Y., Sacks, M.S., 2015a. Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models. Comput. Mech. 55, 1211–1225.

Hsu, M.C., Wang, C., Herrema, A.J., Schillinger, D., Ghoshal, A., Bazilevs, Y., 2015b. An interactive geometry modeling and parametric design platform for isogeometric analysis. Comput. Math. Appl. 70, 1481–1500.

Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y., 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Comput. Methods Appl. Mech. Eng. 194, 4135–4195.

Hughes, T.J.R., Mazzei, L., Jansen, K.E., 2000. Large eddy simulation and the variational multiscale method. Comput. Vis. Sci. 3, 47–59.

Hughes, T.J.R., Mazzei, L., Oberai, A.A., Wray, A., 2001. The multiscale formulation of large eddy simulation: decay of homogeneous isotropic turbulence. Phys. Fluids 13, 505–512.

Hughes, T.J.R., Reali, A., Sangalli, G., 2010. Efficient quadrature for NURBS-based isogeometric analysis. Comput. Methods Appl. Mech. Eng. 199, 301–313.

Hughes, T.J.R., Scovazzi, G., Franca, L.P., 2004. Multiscale and stabilized methods. In: Stein, E., de Borst, R., Hughes, T.J.R. (Eds.), Encyclopedia of Computational Mechanics. Volume 3: Fluids. John Wiley & Sons. Chapter 2.

Jansen, K.E., Whiting, C.H., Hulbert, G.M., 2000. A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. Comput. Methods Appl. Mech. Eng. 190, 305–319.

Kamensky, D., Hsu, M.C., Schillinger, D., Evans, J.A., Aggarwal, A., Bazilevs, Y., Sacks, M.S., Hughes, T.J.R., 2015. An immersogeometric variational framework for fluid–structure interaction: application to bioprosthetic heart valves. Comput. Methods Appl. Mech. Eng. 284, 1005–1053.

Karypis, G., Kumar, V., 1999. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. 20, 359–392.

Kiendl, J., Bletzinger, K.U., Linhard, J., Wüchner, R., 2009. Isogeometric shell analysis with Kirchhoff–Love elements. Comput. Methods Appl. Mech. Eng. 198, 3902–3914.

Kiendl, J., Schmidt, R., Wüchner, R., Bletzinger, K.U., 2014. Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. Comput. Methods Appl. Mech. Eng. 274, 148–167.

Kostas, K.V., Ginnis, A.I., Politis, C.G., Kaklis, P.D., 2015. Ship-hull shape optimization with a T-spline based BEM-isogeometric solver. Comput. Methods Appl. Mech. Eng. 284, 611–622.

Krishnamurthy, A., Khardekar, R., McMains, S., 2009. Optimized GPU evaluation of arbitrary degree NURBS curves and surfaces. Comput. Aided Des. 41, 971–980.

Krishnamurthy, A., McMains, S., Haller, K., 2011. GPU-accelerated minimum distance and clearance queries. IEEE Trans. Vis. Comput. Graph. 17, 729–742.

Lee, Y.K., Lim, C.K., Ghaziaalam, H., Vardhan, H., Eklund, E., 2010. Surface mesh generation for dirty geometries by the Cartesian shrink-wrapping technique. Eng. Comput. 26, 377–390.

Li, X., Scott, M.A., 2014. Analysis-suitable T-splines: characterization, refineability, and approximation. Math. Models Methods Appl. Sci. 24, 1141–1164.

Liu, J., Gómez, H., Evans, J.A., Hughes, T.J.R., Landis, C.M., 2013. Functional entropy variables: a new methodology for deriving thermodynamically consistent algorithms for complex fluids, with particular reference to the isothermal Navier–Stokes–Korteweg equations. J. Comput. Phys. 248, 47–86.

Liu, L., Zhang, Y., Hughes, T.J.R., Scott, M.A., Sederberg, T.W., 2014. Volumetric T-spline construction using Boolean operations. Eng. Comput. 30, 425–439.

Marcum, D.L., Gaither, J.A., 2000. Unstructured grid generation for aerospace applications. In: Salas, M.D., Anderson, W.K. (Eds.), Computational Aerosciences in the 21st Century, vol. 8. Springer, Netherlands, pp. 189–209.

Nitsche, J.A., 1970. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. Abh. Math. Semin. Univ. Hamb. 36, 9–15.

Parvizian, J., Düster, A., Rank, E., 2007. Finite cell method: h- and p- extension for embedded domain methods in solid mechanics. Comput. Mech. 41, 122–133.

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. 249–250, 104–115.

Reali, A., Gómez, H., 2015. An isogeometric collocation approach for Bernoulli–Euler beams and Kirchhoff plates. Comput. Methods Appl. Mech. Eng. 284, 623–636.

Requicha, A.A.G., Voelcker, H.B., 1985. Boolean operations in solid modeling: boundary evaluation and merging algorithms. Proc. IEEE 73, 30–44.

Rhinoceros, 2016. http://www.rhino3d.com/. Accessed 5 February 2016.

Rossignac, J.R., Requicha, A.A.G., 1999. Solid modeling. Technical Report GIT-GVU-99-09. Georgia Institute of Technology.

Ruess, M., Schillinger, D., Özcan, A.I., Rank, E., 2014. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. Comput. Methods Appl. Mech. Eng. 269, 46–731.

Saad, Y., Schultz, M., 1986. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. 7, 856–869.

Schillinger, D., Dedè, L., Scott, M.A., Evans, J.A., Borden, M.J., Rank, E., Hughes, T.J.R., 2012a. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. Comput. Methods Appl. Mech. Eng. 249–252, 116–150.

Schillinger, D., Evans, J.A., Reali, A., Scott, M.A., Hughes, T.J.R., 2013. Isogeometric collocation: cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. Comput. Methods Appl. Mech. Eng. 267, 170–232.

Schillinger, D., Hossain, S.J., Hughes, T.J.R., 2014. Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. Comput. Methods Appl. Mech. Eng. 277, 1–45.

Schillinger, D., Ruess, M., 2015. The finite cell method: a review in the context of higher-order structural analysis of CAD and image-based geometric models. Arch. Comput. Methods Eng. 22, 391–455.

Schillinger, D., Ruess, M., Zander, N., Bazilevs, Y., Düster, A., Rank, E., 2012b. Small and large deformation analysis with the p- and B-spline versions of the finite cell method. Comput. Mech. 50 (4), 445–478.

Schmidt, R., Wüchner, R., Bletzinger, K.U., 2012. Isogeometric analysis of trimmed NURBS geometries. Comput. Methods Appl. Mech. Eng. 241–244, 93–111.

Scott, M.A., Borden, M.J., Verhoosel, C.V., Sederberg, T.W., Hughes, T.J.R., 2011. Isogeometric finite element data structures based on Bézier extraction of T-splines. Int. J. Numer. Methods Eng. 88, 126–156.

Scott, M.A., Li, X., Sederberg, T.W., Hughes, T.J.R., 2012. Local refinement of analysis-suitable T-splines. Comput. Methods Appl. Mech. Eng. 213–216, 206–222.

Scott, M.A., Simpson, R.N., Evans, J.A., Lipton, S., Bordas, S.P.A., Hughes, T.J.R., Sederberg, T.W., 2013. Isogeometric boundary element analysis using unstructured T-splines. Comput. Methods Appl. Mech. Eng. 254, 197–221.

Shakib, F., Hughes, T.J.R., Johan, Z., 1989. A multi-element group preconditioned GMRES algorithm for nonsymmetric systems arising in finite element analysis. Comput. Methods Appl. Mech. Eng. 75, 415–456.

Simpson, R.N., Bordas, S.P.A., Trevelyan, J., Rabczuk, T., 2012. A two-dimensional isogeometric boundary element method for elastostatic analysis. Comput. Methods Appl. Mech. Eng. 209–212, 87–100.

Stein, P., Hsu, M.C., Bazilevs, Y., Beucke, K., 2012. Operator- and template-based modeling of solid geometry for isogeometric analysis with application to vertical axis wind turbine simulation. Comput. Methods Appl. Mech. Eng. 213–216, 71–83.

Temizer, İ., Wriggers, P., Hughes, T.J.R., 2012. Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS. Comput. Methods Appl. Mech. Eng. 209–212, 115–128.

Tezduyar, T.E., 1992. Stabilized finite element formulations for incompressible flow computations. Adv. Appl. Mech. 28, 1–44.

Tezduyar, T.E., Osawa, Y., 2000. Finite element stabilization parameters computed from element matrices and vectors. Comput. Methods Appl. Mech. Eng. 190, 411–430.

Varduhn, V., Hsu, M.C., Ruess, M., Schillinger, D., 2016. The tetrahedral finite cell method: higher-order immersogeometric analysis on adaptive non-boundary-fitted meshes. Int. J. Numer. Methods Eng. http://dx.doi.org/10.1002/nme.5207.

Wang, Z.J., Srinivasan, K., 2002. An adaptive Cartesian grid generation method for 'Dirty' geometry. Int. J. Numer. Methods Fluids 39, 703–717.

Wang, W., Zhang, Y., Liu, L., Hughes, T.J.R., 2013. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. Comput. Aided Des. 45, 351–360.

Xu, F., Schillinger, D., Kamensky, D., Varduhn, V., Wang, C., Hsu, M.C., 2016. The tetrahedral finite cell method for fluids: immersogeometric analysis of turbulent flow around complex geometries. Comput. Fluids. http://dx.doi.org/10.1016/j.compfluid.2015.08.027.

Zhang, Y., Bazilevs, Y., Goswami, S., Bajaj, C., Hughes, T.J.R., 2007. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. Comput. Methods Appl. Mech. Eng. 196, 2943–2959.

Zhang, Y., Wang, W., Hughes, T.J.R., 2013. Conformal solid T-spline construction from boundary T-spline representations. Comput. Mech. 51, 1051–1059.