

Yingjun WANG, Liang GAO, Jinping QU, Zhaohui XIA, Xiaowei DENG

Isogeometric analysis based on geometric reconstruction models

© The Author(s) 2021. This article is published with open access at link.springer.com and journal.hep.com.cn

Abstract In isogeometric analysis (IGA), the boundary representation of computer-aided design (CAD) and the tensor-product non-uniform rational B-spline structure make the analysis of three-dimensional (3D) problems with irregular geometries difficult. In this paper, an IGA method for complex models is presented by reconstructing analysis-suitable models. The CAD model is represented by boundary polygons or point cloud and is embedded into a regular background grid, and a model reconstruction method is proposed to obtain the level set function of the approximate model, which can be directly used in IGA. Three 3D examples are used to test the proposed method, and the results demonstrate that the proposed method can deal with complex engineering parts reconstructed by boundary polygons or point clouds.

Keywords isogeometric analysis, reconstruction model, level set function, trimmed element, point cloud

Received May 5, 2021; accepted June 9, 2021

Yingjun WANG, Jinping QU
National Engineering Research Center of Novel Equipment for Polymer Processing, The Key Laboratory of Polymer Processing Engineering of the Ministry of Education, Guangdong Provincial Key Laboratory of Technique and Equipment for Macromolecular Advanced Manufacturing, South China University of Technology, Guangzhou 510641, China

Liang GAO
State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Zhaohui XIA
National Enterprise Information Software Engineering Research Center, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Xiaowei DENG (✉)
Department of Civil Engineering, The University of Hong Kong, Hong Kong, China
E-mail: xwdeng@hku.hk

1 Introduction

Isogeometric analysis (IGA), which was first proposed by Hughes et al. [1], uses the same basis functions for geometric and computational models to integrate computer-aided design (CAD) and engineering analysis, is becoming an efficient numerical method. IGA has been successfully employed in the computation of various domains, such as vibration [2,3], beams and shells [4,5], fatigue modeling [6–8], fluid and fluid–structure interaction (FSI) [9–12], and structural optimization [13–17], because of the advantages of IGA, such as low computational cost and high accuracy.

The most important part of IGA is the parameterization of analysis domain with spline models. In practice, representing a complex model by a complete tensor-product non-uniform rational B-spline (NURBS) patch, which impedes IGA to be utilized in engineering models with complex topological structures, is quite difficult. Although the use of multiple patches and multiple point constraints in IGA can deal with a model with complex geometry [18,19], it is very complicated and cannot guarantee the high continuity between patches. Moreover, CAD systems usually use boundary representation (B-rep) as an industry standard to represent geometric objects [20]. Thus, the 3D splines for 3D problems do not exist in CAD systems. To solve the above problems, researchers attempt to find solutions from different aspects and obtain some achievements. Zhu et al. [21] proposed a surface parameterization method using the one-step inverse forming and used Coons surface parameterization to rebuild the CAD models for IGA. Jaxon and Qian [22] presented the IGA on triangulations, where the geometry and physical fields were triangulated by rational triangular Bernstein–Bézier splines, to achieve the automated meshing of complex objects. Martin et al. [23] used discrete volumetric harmonic functions to parameterize a 3D model, and then a single trivariate B-spline was used to fit the parameterized model and made

it suitable for IGA. However, the above methods are all based on discrete models, which cannot avoid meshing techniques. Xu et al. [24] investigated the volume parameterization problem of the multi-block computational domain for IGA, where a constraint optimization problem of minimizing quadratic energy functions was solved to generate the multi-block volume parameterization without self-intersections. Zuo et al. [25] proposed an IGA method for constructive solid geometry (CSG) models, which decomposed a complex model into simple primitive models represented by NURBS and then constructed an analysis-suitable CSG model by Boolean operators, such as union, subtraction, and intersection operators. In addition, the use of other splines that easily represent complex geometries (e.g., T-splines [26,27] and U-splines [28]), is also an alternative for IGA. Although these approaches can alleviate the difficulty of IGA model generation, they have to introduce complicated data structures incompatible to general CAD systems or solve extra problems, thereby imposing extra difficulties and burdens on the preprocessing of IGA.

Another type of method to deal with complex geometric models in IGA is the use of trimming techniques. Kim et al. [29] proposed an IGA for trimmed CAD surfaces, where the trimmed elements were decomposed into triangular or quadrilateral elements with only one trimmed edge, and the integration of trimmed elements was implemented by coordinate transformation. Wang et al. [30] further subdivided the types of trimmed elements, proposed corresponding integration strategies, and implemented the IGA with multiple intersected surfaces. Nagy et al. [31,32] proposed a numerical algorithm to construct efficient quadrature rules for trimmed elements by designing the locations and weights of quadrature points and applied it to isogeometric finite element analysis (FEA) and isogeometric boundary element analysis. Ruess et al. [33] proposed a reduced integration method for trimmed elements where a quadtree-based decomposition was used to locally subdivide trimmed elements, and a recursive bisection approach was used to identify the integration points inside the trimmed elements. Marussig et al. [34] presented extended B-splines as a stable basis for IGA with trimmed elements, in which the B-splines may cause ill-conditioned system matrices substituted by extensions of untrimmed B-splines. Guo et al. [35] proposed an isogeometric shell analysis with trimmed surfaces based on the standard for the exchange of product model data (STEP) format, where mapping and subdivision techniques were used to implement the integration of trimmed elements. More information about trimming techniques in IGA can be found in a recent comprehensive review written by Marussig and Hughes [36], which includes the design, data exchange and computational simulation, and challenges of trimming in IGA.

Apart from the aforementioned approaches, the

embedded domain method that immerses the analysis model into a simple non-boundary-fitted background mesh is also an appropriate candidate for applying IGA to complex models. The finite cell method (FCM) [37,38] embeds the original analysis domain in a geometrically larger domain with simple geometry as the approximate solution domain, and such approximate domain can be easily represented by standard tensor-product NURBS basis functions. In addition, for geometric models represented by voxel models (e.g., the models from medical imaging technologies), the IGA cannot be used because no CAD model is created, and such a problem can be solved by combining the FCM and IGA. For a comprehensive review of FCM and its application in IGA, details can be found in the article of Schillinger and Ruess [39]. Similar to the FCM, another embedded domain method called immersogeometric analysis was proposed by Kamensky et al. [40], which directly analyzes a spline-based structure by immersing it into a non-boundary-fitted analysis mesh. This method was first applied to the FSI simulation of bioprosthetic heart valves and further extended to analyze turbulent flow around complex geometries by a tetrahedral FCM [41]. In embedded domain methods, capturing geometry, which is the core for classifying the quadrature points that determine the accuracy of analysis, is significant. For example, the quadrature points in a trimmed element can be divided into two types: (1) points in the physical domain for numerical integration, and (2) points in the fictitious domain that should be discarded. To improve the performance of integration, a local refinement scheme is used to subdivide the trimmed elements into subcells, and the quadrature point classification will be repeatedly applied to such subcells. For complex CAD models, the point membership classification and a large number of integration points are time-consuming. Furthermore, since embedded domain methods are non-boundary-fitted, special approaches will be used to enforce Dirichlet boundary conditions and visualize analysis results.

Constructing geometric models from draft to solid models in CAD systems is called forward modeling, where all the geometric parameters must be known before the model construction. In many industries, the geometric models are directly obtained from the physical models, which is called reverse modeling (i.e., reverse engineering). For example, the automotive designers usually work on physical prototypes, such as clay or wood models, to achieve a favorable shape in automobile industries [42], and the geometric models of medical models (e.g., organs and bones) are created by reverse modeling based on the digital images from computed tomography, magnetic resonance imaging, and other medical scanning processes [43]. In reverse engineering, the procedure of generating geometric models is called model reconstruction, and such models are called reconstruction models. Model reconstruction can be

divided into two types: (1) surfaced-based methods and (2) feature-based methods [44]. Comparing to the feature-based methods, the surfaced-based methods are more mature and adopted by commercial software, such as CATIA, Geomagic, and Rapidform. The most important part of the surfaced-based methods is the isosurface generation. In 1987, marching cube (MC) algorithm was proposed by Lorensen and Cline, which has become a standard of isosurface generation because of its simplicity and high speed. However, the original MC algorithm may cause ambiguity problems (i.e., different isosurfaces may be obtained from the same voxel data) [45]. To eliminate the ambiguous problems, many new algorithms were proposed by different research groups (e.g., extended MC algorithms [46–48] and marching tetrahedra algorithms [49–51]). Nowadays, reconstruction models can accurately represent the physical models. Hence, the numerical analysis (e.g., FEA and IGA) based on such reconstruction models can also accurately predict the performance of the physical objects. Zander et al. [52] directly used a 3D finite cell toolbox FCMLab to work on a femur defined by voxel model. Verhoosel et al. [53] proposed an image-based adaptive IGA and applied it to the micro-mechanical modeling of trabecular bone, where the goal-adaptive FCM was used to compute elastic properties. Recently, Kudela et al. [54] completed the direct structural analysis of domains defined by oriented point clouds based on the FCM. However, the adaptive FCM will increase the complexity of the analysis, as well as the computational cost.

In this paper, an IGA framework based on reconstruction models is proposed, which can be directly used to analyze complex models with arbitrary geometries, as well as the point cloud data. The outline of this paper is organized as follows: In Section 2, a brief overview of NURBS-based IGA is given; in Section 3, a model reconstruction method suitable for IGA is established; in Section 4, numerical quadrature for elements in detail is presented; in Section 5, numerical examples to verify the efficiency and accuracy of the proposed method are modeled and analyzed; finally, conclusions are summarized in Section 6.

2 Overview of NURBS-based IGA

NURBS [55], which is constructed from B-splines and commonly used in computer graphics for representing curves and surfaces, are used here for the IGA. A knot vector Ξ consists of a sequence of non-decreasing real numbers as

$$\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}, \quad (1)$$

where p is the order of the B-spline, ξ_i ($i = 1, 2, \dots, n+p+1$) is variable for the knot coordinate in parameter space, and n is the number of basis functions. The whole

interval $[\xi_1, \xi_{n+p+1}]$ is a patch, and the knot interval $[\xi_i, \xi_{i+1})$ is a span.

Referred to the Cox-de Boor formula [56], the B-spline basis functions are recursively defined as

$$\begin{cases} B_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi \leq \xi_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \\ B_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} B_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,p-1}(\xi), \end{cases} \quad (2)$$

where $B_{i,p}(\xi)$ is the i th B-spline basis function of p -degree for the coordinate ξ in parameter space.

One important property of the B-spline functions is that they constitute a partition of unity.

$$\sum_{i=1}^n B_{i,p}(\xi) = 1, \quad (3)$$

and another important property is that the continuity between knot spans achieves C^{p-k} where k is the multiplicity of the knots.

NURBS basis functions $N_{i,p}(\xi)$ are obtained from B-splines by introducing a positive weight w_i to the B-spline basis functions as

$$N_{i,p}(\xi) = \frac{B_{i,p}(\xi)w_i}{\sum_{j=1}^n B_{j,p}(\xi)w_j}. \quad (4)$$

Two-dimensional NURBS basis functions from the knot vectors $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ and $H = \{\eta_1, \eta_2, \dots, \eta_{m+q+1}\}$ are constructed as

$$N_{i,p}^{j,q}(\xi, \eta) = N_{i,p}(\xi)N_{j,q}(\eta) = \frac{B_{i,p}(\xi)B_{j,q}(\eta)w_{i,j}}{\sum_{k=1}^n \sum_{l=1}^m B_{k,p}(\xi)B_{l,q}(\eta)w_{k,l}}, \quad (5)$$

where $w_{i,j}$ is the weight value corresponding to the tensor product $B_{i,p}(\xi)B_{j,q}(\eta)$.

In the NURBS-based IGA, the above NUBRS basis functions are used for representing the physical fields as

$$x(\xi, \eta) = N^T(\xi, \eta)\mathbf{x}, \quad (6)$$

where $x(\xi, \eta)$ can be a displacement, force, or other physical values at the position whose coordinates in the parameter space is (ξ, η) , and $N(\xi, \eta)$ and \mathbf{x} are the basis function vector and the physical value vector corresponding to the control points acting on the position (ξ, η) , respectively.

In this work, the NURBS-based IGA is used to solve static linear elasticity problems. The element stiffness matrix \mathbf{K}_e may be written as [14]

$$\mathbf{K}_e = \int_{\Omega_e} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega = \int_{\hat{\Omega}_e} \mathbf{B}^T \mathbf{D} \mathbf{B} |J_1| d\hat{\Omega} = \int_{\bar{\Omega}_e} \mathbf{B}^T \mathbf{D} \mathbf{B} |J_1| |J_2| d\bar{\Omega}, \quad (7)$$

where \mathbf{B} and \mathbf{D} are the strain–displacement matrix and the stress–strain matrix, respectively, and Ω_e , $\hat{\Omega}_e$, and $\bar{\Omega}_e$ are the physical, NURBS parametric, and integration domains of the element, respectively. Jacobian J_1 and J_2

indicate the transformation relation from the NURBS parametric space to the physical space and the integration parametric space to the NURBS parametric space, respectively.

3 Model reconstruction method

In CAD systems, the B-rep is used to represent a geometric object, which means that only boundary information is included in the CAD models (e.g., a 3D model represented by its boundary with only surface information). The B-rep representation is quite different from the computational mesh of finite-element-based computer-aided engineering (CAE), and Hughes et al. [1] illustrated that up to 80% of overall analysis time is devoted to mesh generation in the automotive, aerospace, and ship building industries. Additionally, the 3D splines for 3D IGA cannot be directly obtained from CAD systems due to the B-rep representation.

In this section, we propose a model reconstruction method that can construct an analysis-suitable model of arbitrary geometry for IGA. To reconstruct an analysis-suitable model, the boundary of the original model is first discretized, similar to the boundary discretization of the boundary element method. Subsequently, the model is embedded into a regular domain, which can be represented by complete NURBS basis functions, and the embedded domain is divided into subcells (i.e., the elements for IGA). Fortunately, the triangular mesh is applied in CAD systems for visualization as the boundary discretization mesh to avoid the use of additional mesh generation approaches [57]. Based on the boundary discretization and model division, the elements are classified into different types, and the empty elements are compressed. The following subsections will give the details of the proposed method.

3.1 Elements of embedded domain

In embedded domain methods, such as FCM and

immersogeometric analysis, a simple unfitted structural mesh is used to approximate the solution field to avoid generating boundary conforming meshes. As shown in Fig. 1, the elements of the embedded domain are divided into three types: 1) fictitious elements, 2) solid elements, and 3) trimmed elements. The identification of element types is one of the key issues for embedded domain methods.

The identification of element types is actually a series of inside/outside tests for element positions. One of the simple and efficient approaches to implement such tests is the ray casting algorithm [58,59], which casts rays from a point and count intersections with a closed geometry and has been successfully applied to irregular geometries [40,60]. For 2D models, the inside/outside test can be efficiently implemented by subdividing the closed curve into segments and by applying the ray casting algorithm. However, for 3D models, the boundary surfaces will be subdivided into polygons, and the ray casting algorithm will be more tedious and time-consuming.

Taking a bracket as an example (see Fig. 2), the embedded domain is a regular cuboid. For an element with n nodes ($n = 8$ in Fig. 2), we can define a rule to identify the element type as follows: (i) fictitious element: All the nodes are outside the bracket domain; (ii) solid element: All the nodes are inside the bracket domain; (iii) trimmed element: Some nodes are inside the bracket domain and the other nodes are outside the domain. If the bracket surfaces are subdivided into m polygons (i.e., triangles in Fig. 2), then the ray casting algorithm has to implement $n \times m$ times of point-polygon intersection tests to identify the element type for one element. When the number of elements and the number of polygons are large, it is very time-consuming to implement such element classification.

3.2 A high-efficient element classification method

As mentioned in Section 3.1, it is inefficient to use ray casting algorithm for element classification of large-scale

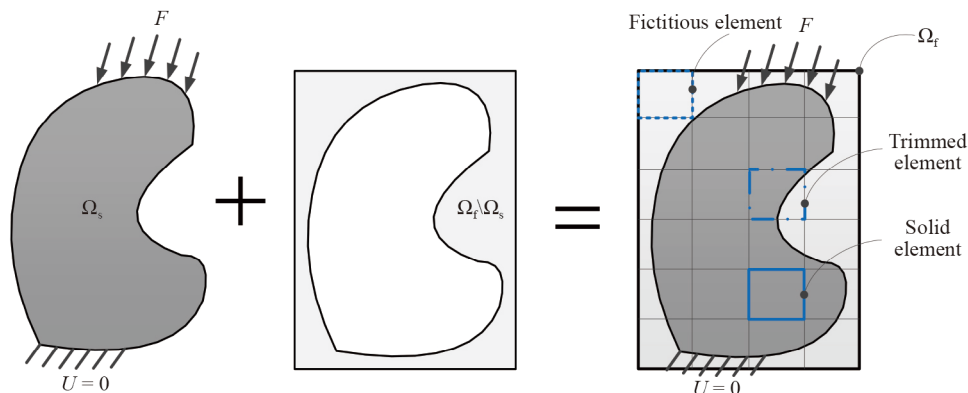


Fig. 1 An illustration of element types in the embedded domain.

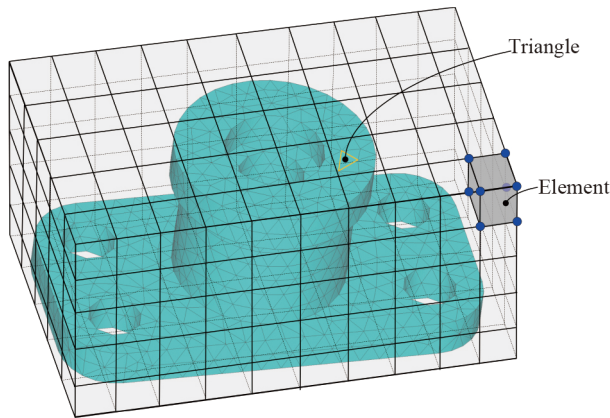


Fig. 2 Embedded domain division of a bracket.

problems. To solve this problem, we will propose a high-efficient element classification method in this section. The element classification method consists of three main steps: 1) embedded domain construction, 2) identification of trimmed elements, and 3) element classification.

3.2.1 CAD surface model

For a CAD model, the main steps of the element classification method are shown in Fig. 3, and a description is given as follows:

(1) A polygon model is first created by dividing the boundary of the CAD model into polygons. In this work, the triangles for the CAD visualization are used as polygons, which can be directly applied to the visualization of results. Based on such boundary polygon model, a bounding box can be generated as the embedded domain, which is divided into regular subdomains (i.e., the elements of the embedded domain). The element size should be larger than the polygon size.

(2) According to the relative position to boundary polygons, the elements of the embedded domain are initially classified into two types: (i) trimmed elements (i.e., the elements cut by the boundary polygons), and (ii) untrimmed elements. As shown in Fig. 4, if any vertex or the center of a polygon is in an element, then the element is defined as a trimmed element; otherwise, it is an untrimmed element.

(3) The minimum distance (MD) from a vertex of a trimmed element to the boundary polygons can be calculated using the algorithm of the MD between a point and a triangle [61], which will be given in Section 3.3. The trimmed element and its adjacent elements are considered an element group. Considering that the element size is larger than the polygon size, we only need to calculate the distances from the vertex to the polygons in the element group and select the minimum one as the MD. The MD can be converted into a sign minimum distance (SMD) by comparing the directions between the vector from the vertex to the polygon plane ν_1 and the

outward vector of the polygon ν_2 . If $\nu_1 \cdot \nu_2 > 0$, then the vertex is in the solid domain, and the sign distance is positive; otherwise, $\nu_1 \cdot \nu_2 < 0$, the vertex is in the fictitious domain, and the sign distance is negative.

After the sign distances are obtained, the elements can be divided into three types by iteratively expanding the signs of the MDs to all vertices of untrimmed elements (Fig. 5). The sign expanding rule is: (i) If the arbitrary vertex of a untrimmed element is in the solid domain (i.e., $SMD > 0$), then all vertices of this element should be in the solid domain, and this element is a solid element; and (ii) if the arbitrary vertex of a untrimmed element is in the fictitious domain (i.e., $SMD < 0$), then all vertices of this element should be in the fictitious domain, and this element is a fictitious element.

3.2.2 Point cloud model

The main steps of the element classification method for point cloud models are given as follows:

(1) Similar to the embedded domain construction of the CAD surface model, a bounding box that contains all the points is generated as the embedded domain, as shown in Fig. 6. The elements are obtained by dividing the embedded domain into regular subdomains. Normals of the point cloud can be directly obtained from the 3D scanner.

(2) The identification of trimmed elements for point cloud models is straightforward. If an element contains at least one point, then this element is a trimmed element, otherwise, it is an untrimmed element.

(3) The MD from a vertex of a trimmed element to the point cloud is obtained by calculating vertex-to-point distance for the points in the trimmed element and its neighbor elements, and then the minimum one is selected as the MD from the vertex to point cloud. Since the outward normals of the points can be directly obtained from general 3D scanners, the SMD is obtained by the vector from the vertex to the nearest point and the outward vector of the point, and the elements can be also divided into three types similar to CAD surface model cases, as that in Section 3.2.1.

3.3 Minimum distance between a point and a triangle in 3D

In Section 3.2, the MD between a point and a triangle plays an important role in element classification. In this study, we used the algorithm proposed in Ref. [61], which is briefly summarized herein. As shown in Fig. 7, a triangle $\triangle ACE$ (vertices are A , C , E) can be parametrically represented by $T(s, t) = A + s\mathbf{l}_1 + t\mathbf{l}_2$ while $(s, t) \in S = \{(s, t) : s \geq 0, t \geq 0, s + t \leq 1\}$, where \mathbf{l}_1 and \mathbf{l}_2 are the vector from A to C and the vector from A to E . The MD between a point P and this triangle is to find the values (\bar{s}, \bar{t}) so that $T(\bar{s}, \bar{t})$ is the triangle point closest to P .

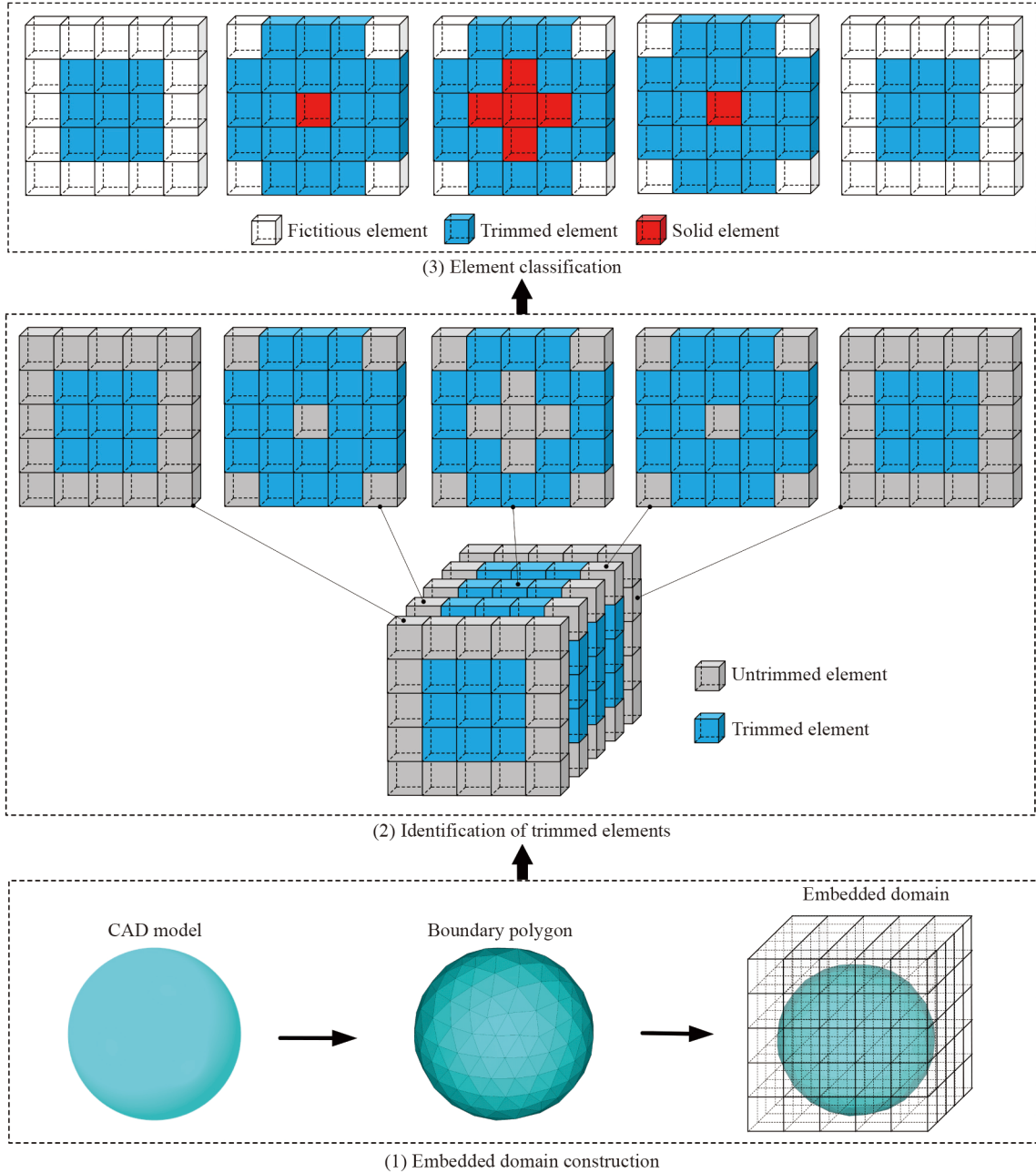


Fig. 3 Flow chart of the high-efficient element classification method.

The distance from P to any point $Q = T(s, t)$ on the triangle ($Dis(s, t)$) is calculated by

$$Dis(s, t) = |Q - P| = \sqrt{as^2 + 2bst + ct^2 + 2ds + 2et + f}, \quad (8)$$

where $a = l_1 \cdot l_1$, $b = l_1 \cdot l_2$, $c = l_2 \cdot l_2$, $d = l_1 \cdot (A - P)$, $e = l_2 \cdot (A - P)$, $f = (A - P) \cdot (A - P)$, respectively.

Since Dis (see Eq. (8)) is a continuously differentiable function, the closest point is either an interior point of domain S where the gradient $\nabla Dis = -(as + bt + d, bs + ct + e) = (0, 0)$ or at a point on the boundary of S . Assuming the closest point locating on the st -plane, the st -plane can be divided into 7 regions by the triangle

domain, as shown in Fig. 8. Assuming $u = be - cd$, $v = bd - ae$, and $w = ac - b^2$, the criteria determining which region contains the closest point are given in Table 1. When the region of the closest point is obtained, the MD is computed in terms of the region, and the detail algorithm can be found in Ref. [61].

3.4 Model reconstruction based on sign minimum distances

Taking the SMDs of vertices as level set function (LSF) values, we reconstruct an approximate model that is implicitly represented by the zero level set isosurface, and

such reconstruction model can be directly used for IGA. When the level set model is obtained, we can construct the explicit geometric model by the marching tetrahedra algorithm [49]. In the marching tetrahedra algorithm, a cube is split into 6 tetrahedrons, as shown in Fig. 9, and a tetrahedron isosurface is constructed for each tetrahedron to reconstruct the model.

For a tetrahedron, if symbol 1/0 indicates that a vertex is inside/outside the boundary polygon, then a binary number of 4 digits can be used to represent different cases

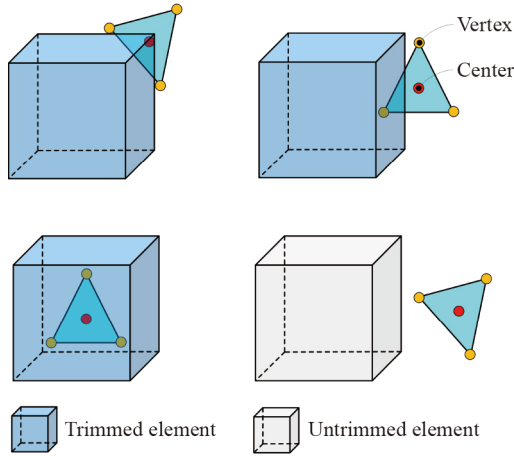


Fig. 4 An illustration of position relation between elements and boundary polygons.

of a tetrahedron cut by a planar facet, as shown in Fig. 10. Since the SMDs of the vertices are known, the symbol 1/0 can be directly obtained. In marching tetrahedra, the planar facet approximate to the zero-distance isosurface is calculated for each tetrahedron. The facet vertices are the points whose MD to the boundary polygon is zero, and these zero-distance points can be obtained by linearly interpolating the SMDs of vertices along each edge as Eq. (9):

$$x_0 = \frac{d_2}{d_2 - d_1} x_1 - \frac{d_1}{d_2 - d_1} x_2, \tag{9}$$

where d_1 and d_2 are the SMDs of the two vertices of an edge, x_1 and x_2 are their corresponding coordinates, and x_0 is the coordinate of the zero-distance point. Figure 11 gives an example of tetrahedron trimming based on SMDs.

The trimmed tetrahedron in Fig. 11 can be regarded as a reconstruction model constructed by marching tetrahedra algorithm based on the MD. Notably, the explicit geometric model is unnecessary in the IGA method proposed in this paper, but the explicit model is necessary for conventional FEA or manufacturing.

3.5 Element renumbering and reusing approaches

When an embedded domain is constructed, all the elements have to be numbered to complete the element

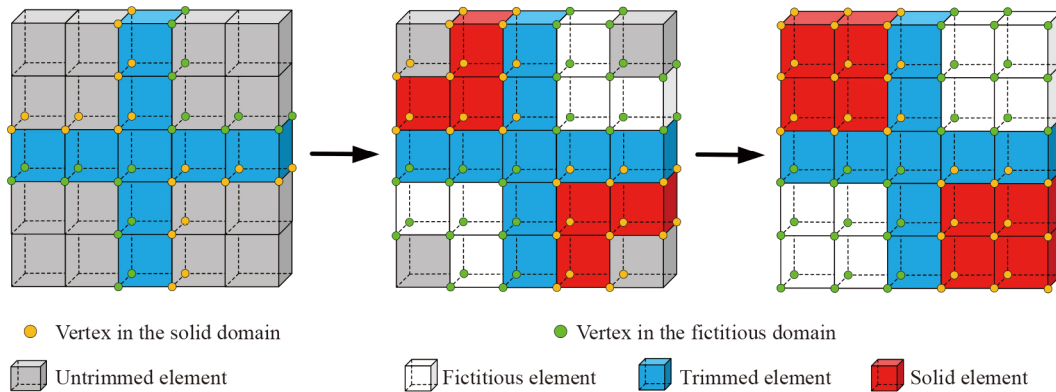


Fig. 5 An illustration of element type classification.

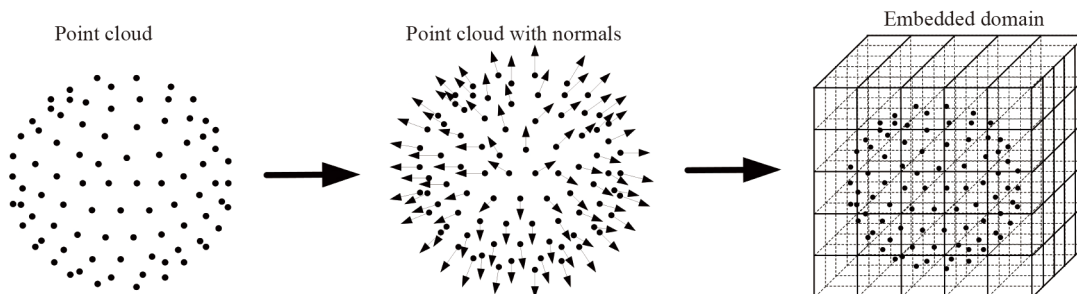


Fig. 6 Flow chart of the high-efficient element classification method.

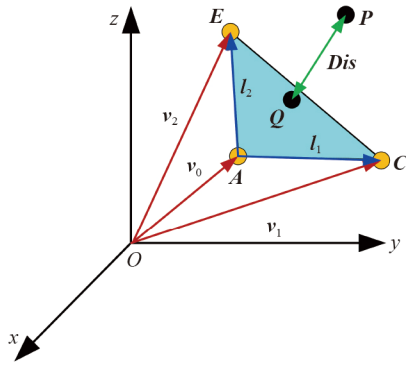


Fig. 7 An illustration of the distance from a point to any point on a triangle.

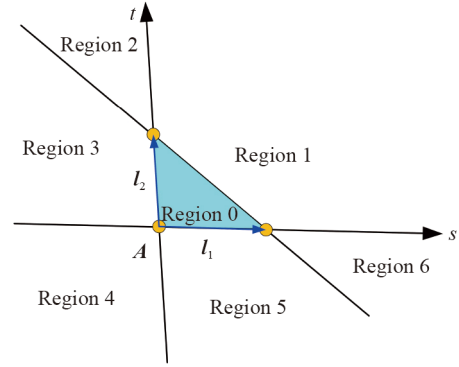


Fig. 8 Partition of the st -plane by the triangle domain.

Table 1 Criteria determining which region contains the closest point

Criteria	Region 0	Region 1	Region 2	Region 3	Region 4	Region 5	Region 6
w	$u + v \leq w$	$u + v > w$	$u + v > w$	$u + v \leq w$	$u + v \leq w$	$u + v \leq w$	$u + v > w$
u	$u \geq 0$	$u \geq 0$	$u < 0$	$u < 0$	$u < 0$	$u \geq 0$	
v	$v \geq 0$	$v \geq 0$		$v \geq 0$	$v < 0$	$v < 0$	$v < 0$

classification, as shown in Fig. 3. However, the fictitious elements that are fully outside the geometric model should be removed in IGA. Thus, the trimmed and solid elements (i.e., the reserved elements) need to be renumbered.

In this work, a simple approach is used to loop over the original elements and renumber the trimmed and solid elements (Fig. 12), and an element renumbering example is given in Fig. 13. In this approach, each trimmed/solid element has two different IDs whose mapping relation is recorded in a list so that the original data (e.g., coordinates of nodes and element connectivity) can be directly reused in IGA based on the mapping relation.

One advantage of the embedded domain method is that the domain can be divided into uniform elements. Therefore, the elements with the same property can be classified into a group, and only one stiffness matrix

needs to be calculated. Considering Fig. 13 as an example, only one stiffness matrix, which can be applied to all the other elements, needs to be solved since all the solid elements have the same geometry and property (Fig. 14). In IGA, apart from the material property, the distribution pattern of control points is also an element

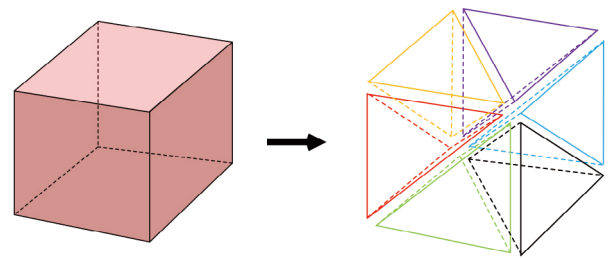


Fig. 9 Tetrahedron split of a cube.

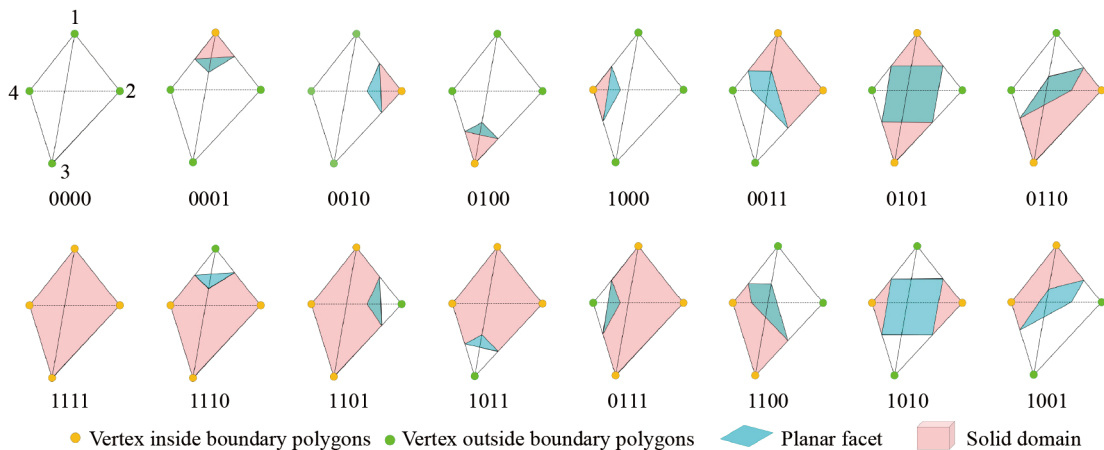


Fig. 10 Different cases of a tetrahedron cut by a planar facet.

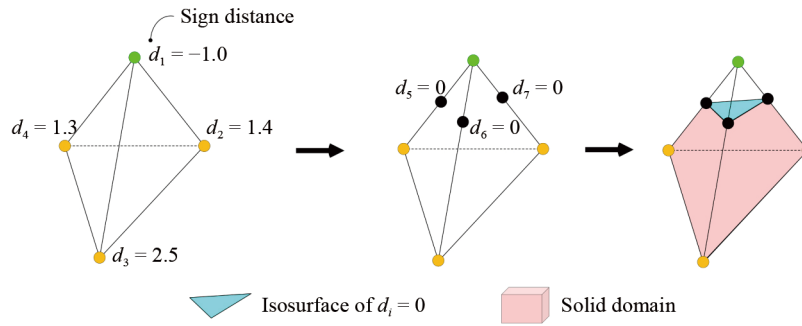


Fig. 11 An example of tetrahedron trimming based on SMDs.

```

numReserved=0; (number of reserved elements)
newEleOID=[]; (a list for recording the reserved elements' original
old IDs)
for each element ei of the embedded domain do
  if ei is a trimmed or solid element then
    numReserved=numReserved+1;
    newEleOID[numReserved]=ei;
  end if
end for
    
```

Fig. 12 Element renumbering algorithm.

property, since the distribution pattern of the control points may be different for the elements that are located at different positions. Figure 15 shows a comparison of computational models (2D problem) between FEA and IGA, where the distribution pattern of nodes is the same

for all elements, but the distribution pattern of control points may be different. Due to the uncertainty of trimming cases, the stiffness matrices of the trimmed elements are still calculated individually. Based on the aforementioned element reusing approach, the computational efficiency of element stiffness matrices will be improved, which is especially useful for large-scale problems.

4 Numerical quadrature for elements

In embedded domain methods, the untrimmed elements are integrated by standard Gauss quadrature method. The trimmed elements are usually integrated by the adaptive quadrature algorithm [39,40]. However, identifying the

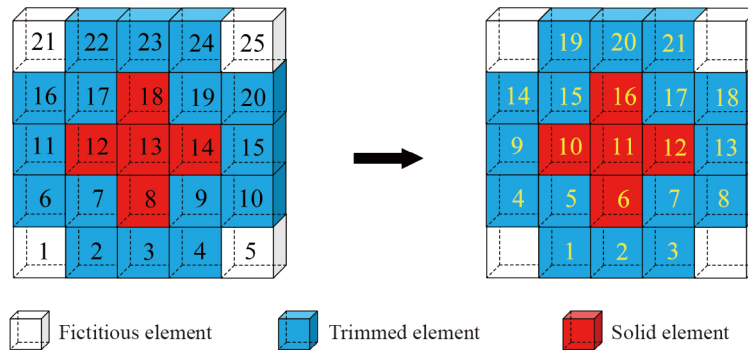


Fig. 13 An illustration of element renumbering.

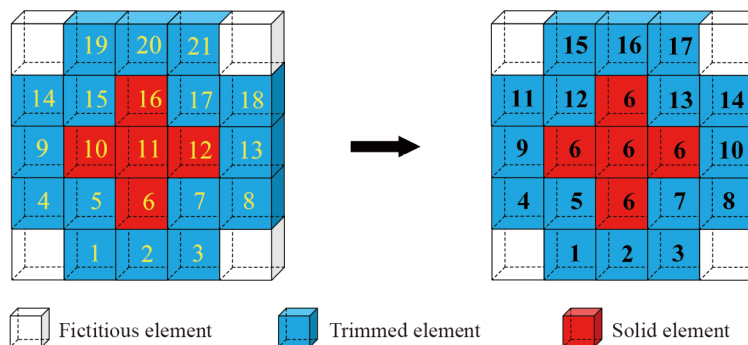


Fig. 14 Stiffness matrix types of the case in Fig. 13.

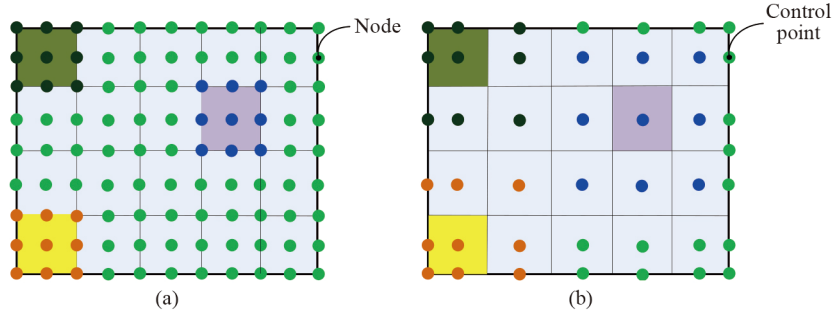


Fig. 15 Comparison of computational models between (a) FEA and (b) IGA.

Gauss points in the solid domain, especially for the complex trimming cases, is very difficult and time-consuming. In this work, the Gauss points in the solid of the trimmed element can be easily identified due to the known LSF values of the element vertices. As such, the trimmed elements can be integrated efficiently.

4.1 Regular element integration

Since the embedded domain is regular, the regular elements can be cubes and the standard Gauss quadrature method can be used to complete the element integration. When the element is a regular cube, the Jacobian determinants $|J_1|$ and $|J_2|$ can be efficiently computed by

$$|J_1| = V_{\Omega_e} / V_{\hat{\Omega}_e}, \quad (10)$$

and

$$|J_2| = V_{\hat{\Omega}_e} / V_{\bar{\Omega}_e}, \quad (11)$$

where V_{Ω_e} , $V_{\hat{\Omega}_e}$, and $V_{\bar{\Omega}_e}$ are the volumes of the element in physical, NURBS parametric, and integration domains, respectively. For 2D problems, the element volume should be replaced by the element area.

4.2 Trimmed element integration

The adaptive quadrature algorithm [39,40] is one of the effective approaches for trimmed element integration, which divides the Gauss points of the embedded trimmed element into two parts: (1) Gauss points in the solid domain, and (2) Gauss points in the fictitious domain. Only the Gauss points in the solid domain will be used in the trimmed element integration. However, the Gauss point classification is very complicated, especially for 3D problems with complex trimming cases, since it is difficult to identify which Gauss point is inside/outside the solid domain if the trimming curves or trimming surfaces are irregular shapes.

When the level set reconstructed model is used, the above problem can be solved by calculating the LSF values at the Gauss points by using a simple interpolation as

$$\phi(\xi, \eta) = \sum_i N_i(\xi, \eta) \phi_i, \quad (12)$$

where $\phi(\xi, \eta)$ is the LSF value of the Gauss point whose

local coordinates are (ξ, η) , and N_i and ϕ_i are the shape function (linear interpolation is used herein and the LSF value of the i th vertex of the regular embedded element, respectively. The above equation can be easily extended to 3D problems as

$$\phi(\xi, \eta, \zeta) = \sum_i N_i(\xi, \eta, \zeta) \phi_i. \quad (13)$$

The Gauss point is in the solid domain if the LSF value of the Gauss point is equal or larger than 0, otherwise, the Gauss point is out of the solid domain (i.e., in the fictitious domain). An illustration of the Gauss point classification is shown in Fig. 16. In this work, two levels of refinement (i.e., the number of levels is 3) and $3 \times 3 \times 3$ Gauss quadrature points in each level are used.

5 Numerical examples

In this section, we use numerical examples, including three CAD models and a point cloud model, to investigate the performance of our IGA method in detail. All examples are run on a desktop: CPU is Intel Xeon E5-2620 2.1GHz, RAM is 64GB, OS is Windows 7 Professional, and the compiler is MATLAB 2014b.

5.1 Cylinder

The first example is a bottom-fixed cylinder subjected to the pressure caused by a heavy box as shown in Fig. 17(a), which can be converted into an analysis model as Fig. 17(b). The boundary polygon model of the cylinder is shown in Fig. 17(c), which can be embedded in a regular domain corresponding to a complete tensor-product NURBS domain. In this model, Young's modulus, E , is 10^5 , the Poisson's ratio, ν , is 0.3, the bottom surface of the cylinder is fixed and the equivalent pressure on the square domain of the top surface is 100.

The cylinder is embedded into a regular domain with $7 \times 7 \times 14$ quadratic 3D NURBS elements. The control points of the top surface are given in Fig. 18, and the pressure loading region is just a part of the surface patch. Imposing the accurate load at control points is not easy since the NURBS control points are not endpoint

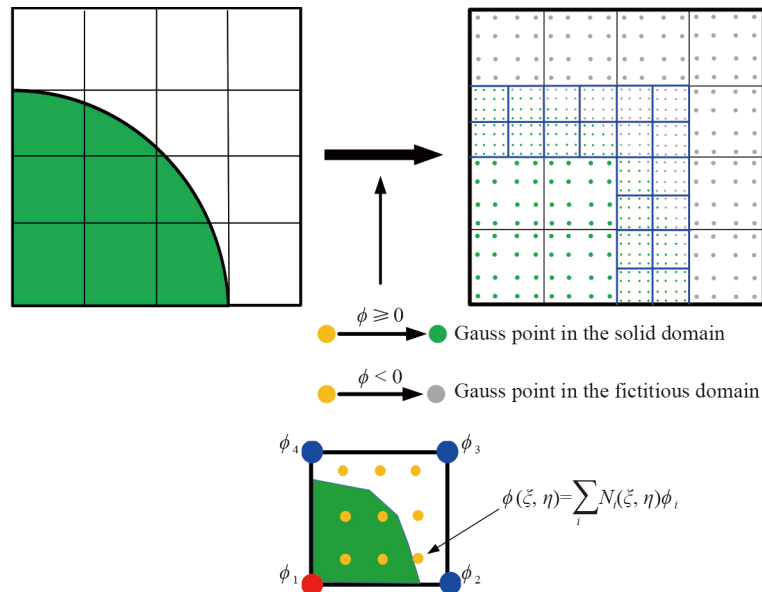


Fig. 16 An illustration of the Gauss point classification.

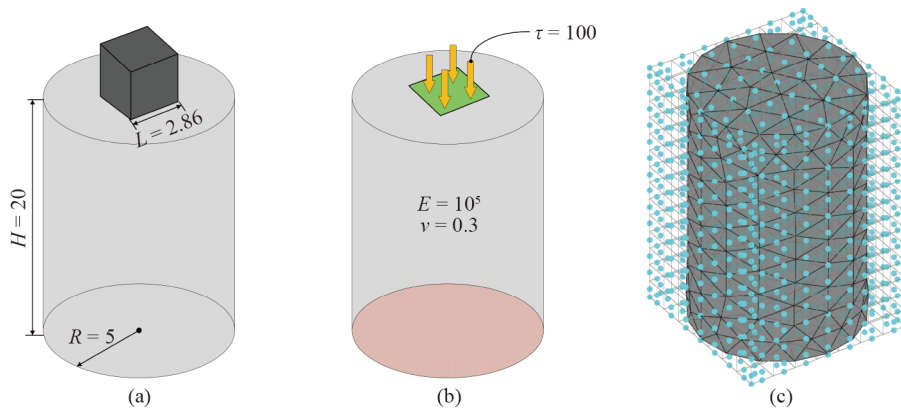


Fig. 17 Cylinder model: (a) physical model, (b) analysis model, and (c) boundary polygon model and its embedded domain.

interpolatory. The load \mathbf{F} to the solid control points that influences the loading region in Fig. 18 can be obtained by the following equation

$$\mathbf{F} = \int_{\Omega} \mathbf{N} \boldsymbol{\tau} d\Omega, \tag{14}$$

where \mathbf{F} is the force vector that should be added at the control points of the loading region, \mathbf{N} is the NURBS shape function vector, which is related to the integration point, and $\boldsymbol{\tau}$ is the traction of the loading region Ω . More accurate loading approaches require more integration points, which can be obtained by the local refinement of the elements.

The displacement and the stress along the pressure direction (i.e., Z direction) are given in Fig. 19. Two finite element reference results from ANSYS (Canonsburg, Pennsylvania, USA) using 1873 (Figs. 19(a) and 19(d)) and 72027 quadratic tetrahedral elements ((Figs. 19(b) and 19(e)) are used to compare with the results of the proposed method ((Figs. 19(c) and

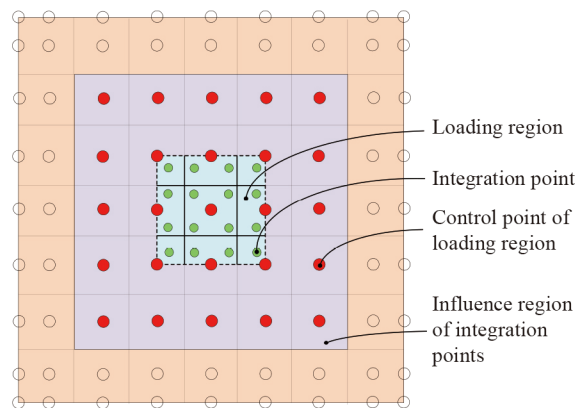


Fig. 18 Local loading illustration of the cylinder.

19(f)). The displacement results are consistent for all the three models, but the large displacement region around the loading region is larger in the IGA result. The reason for this is that the forces are imposed at the control points

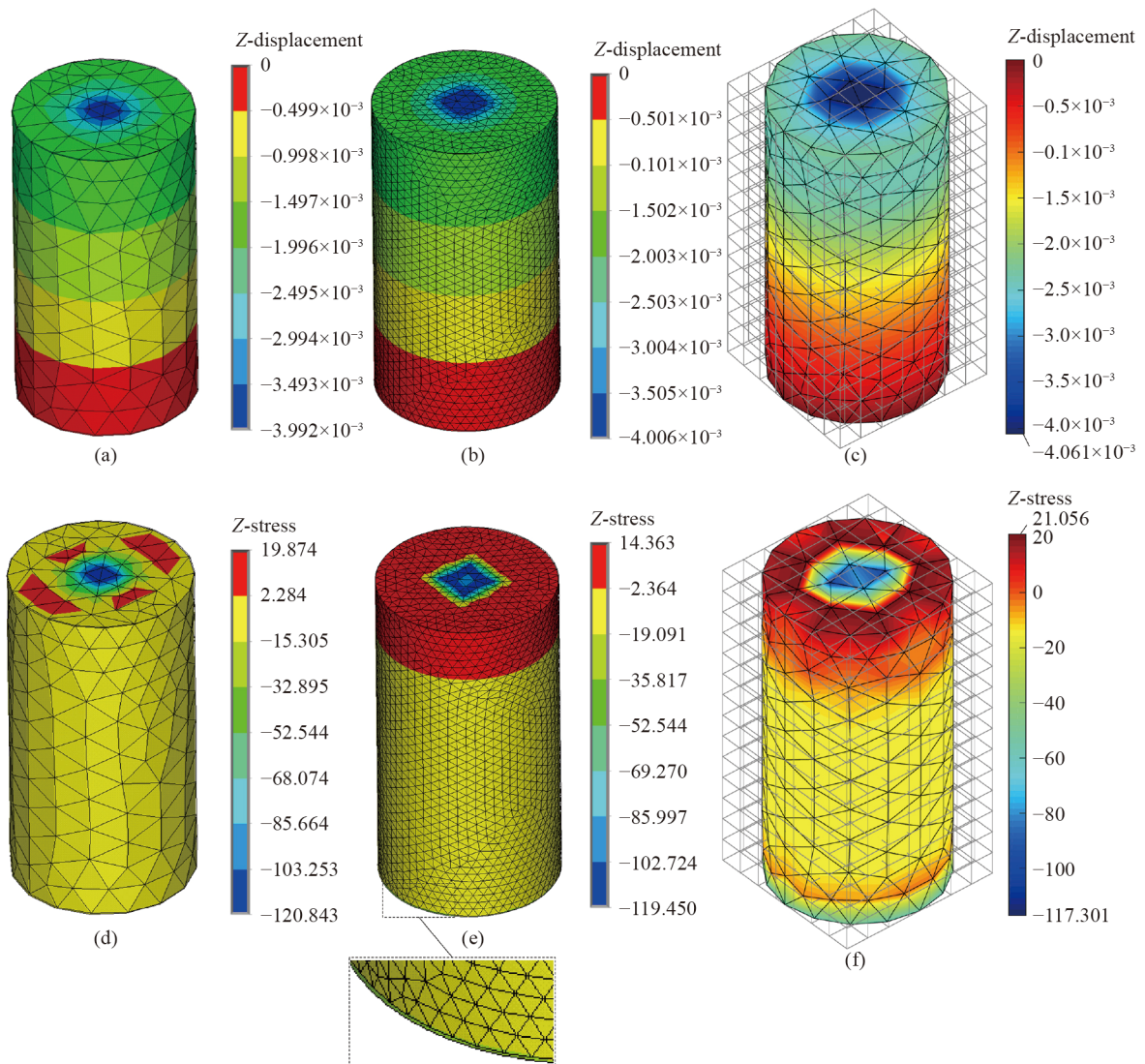


Fig. 19 Z-direction results of the cylinder model: (a) Z-displacement of FEA with 1873 elements, (b) Z-displacement of FEA with 72027 elements, (c) Z-displacement of IGA with 686 elements, (d) Z-stress of FEA with 1873 elements, (e) Z-stress of FEA with 72027 elements, and (f) Z-stress of IGA with 686 elements.

(solid control points in Fig. 18) which influence a larger region than the loading region of FEA. Comparing to the FEA with 1873 elements whose element size is similar to that of IGA, the stress distribution of IGA is closer to the FEA with 72027 elements which is regarded as the reference standard, especially for the stress concentration around the edge of the bottom surface. The above demonstrates the validity and accuracy of the proposed IGA method.

5.2 Automotive part

An automotive part subjected to tension loading is shown in Fig. 20, where the bottom is fixed; and the tension pressure is 10^4 , and Young's modulus and the Poisson's ratio are 10^9 and 0.3, respectively.

The automotive part is embedded into a regular domain

with $16 \times 32 \times 8 = 4096$ quadratic 3D NURBS elements. Figs. 21(a) and 21(b) show the total deformation results of the FEA of ANSYS using 40143 quadratic tetrahedral elements and the proposed IGA, respectively. The two results are almost the same, that is, both results demonstrate the high accuracy of the proposed method for real engineering parts. Notably, the boundary triangular facets can be directly obtained from the CAD system, which usually uses triangular facets for visualization [57]. Thus, an additional meshing tool is not necessary.

5.3 Point cloud of a gear shaft

Different from the abovementioned examples whose reconstruction models are constructed based on boundary polygons, this gear shaft example will show how to

directly implement the IGA based on point cloud data. The flowchart of the IGA in this paper is given in Fig. 22 which consists of the following main steps:

(1) The 3D scanner JTscan-DS (Jeatech, Guangzhou, China) is used to scan the physical model of the gear shaft. The 360-degree auto scan mode is used to scan the model, and the oriented point cloud can be directly obtained by using the JTscan-DS software. Therefore, the reconstruction model can be constructed by applying the approach proposed in Section 3.

(2) A total of 140959 points can be observed in the point cloud, and the point cloud is embedded into a regular domain with $16 \times 32 \times 16 = 8192$ quadratic 3D NURBS elements. The analysis model is constructed by

fixing the bottom and adding forces at the 24 control points (the force for each control point is 100) of a gear tooth as shown in Fig. 22(b), and Young’s modulus and the Poisson’s ratio are 105 and 0.3, respectively.

(3) After the IGA, the deformation of the control points is obtained, and the deformation of each point of the point cloud can be also obtained by the NURBS interpolation, which is shown in Fig. 22(c). The maximum deformation occurs at the force loading position while the minimum deformation occurs on the fixed bottom face, which is reasonable for the gear shaft model subjected to the boundary condition as Fig. 22(c). This example demonstrates that our method can be directly used to analyze point clouds.

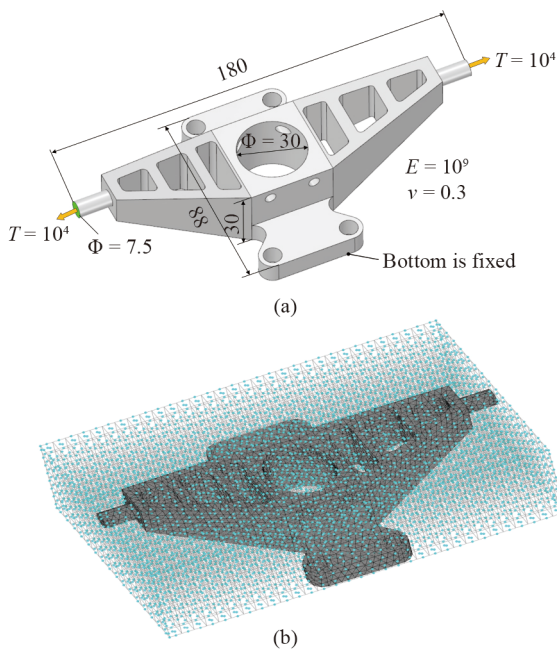


Fig. 20 Automotive part model: (a) analysis model and (b) boundary polygon model and its embedded domain.

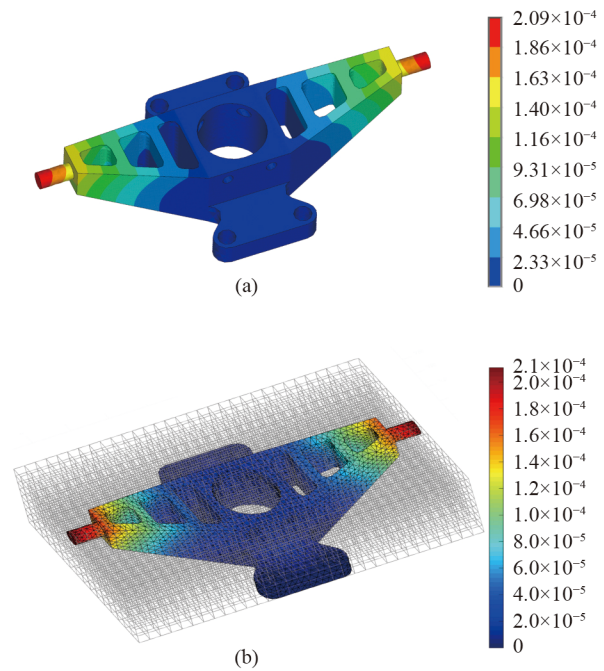


Fig. 21 Total deformation results of the automotive part model: (a) FEA result and (b) IGA result.

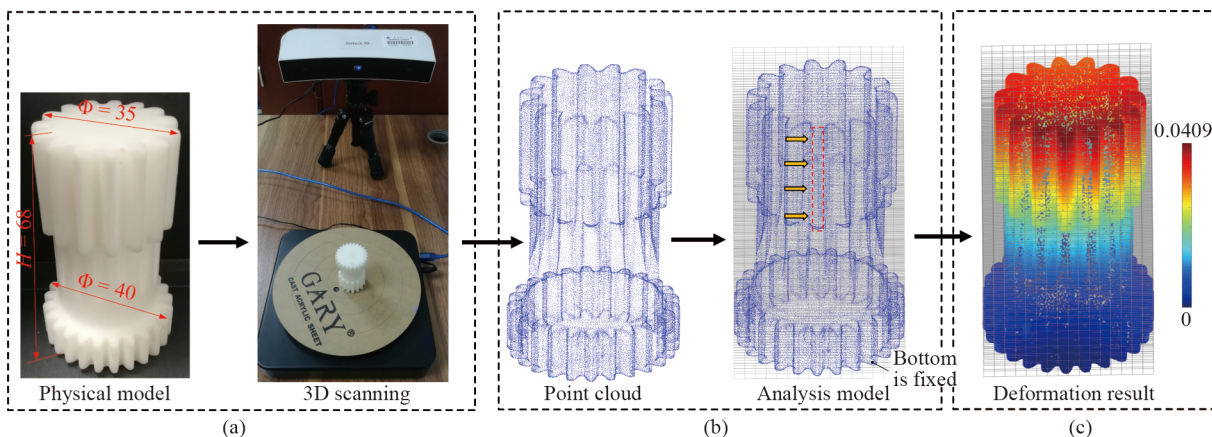


Fig. 22 Flowchart of the proposed IGA method for point cloud model: (a) physical model scanning by a 3D scanner, (b) analysis model construction by embedding the point cloud into a regular domain, and (c) deformation result of the proposed IGA.

6 Conclusions

This paper has proposed a new IGA method based on reconstruction models. In this method, a CAD model represented by boundary polygons or point cloud is embedded into a regular domain with NURBS mesh, and the LSF is obtained by a model reconstruction method that efficiently calculates the minimum sign distances between the vertices of NURBS mesh and the boundary polygons/points to approximately describe the CAD model. The Gauss points in trimmed elements can be directly obtained by evaluating the LSF values at the Gauss points, so the integration of trimmed elements can be easily implemented by the adaptive quadrature algorithm. Three numerical examples, including CAD models represented by boundary polygons and point cloud, are used to test the proposed method, and the results demonstrate its feasibility and validity, especially for the direct analysis of point cloud data.

In the future, a better method to impose Dirichlet boundary conditions will be explored to solve the low-fidelity loading problem, and a more accurate and efficient quadrature method (e.g., the optimal quadrature approach [31,32]) will be tested to replace the adaptive quadrature algorithm. Furthermore, the proposed method will be expanded to structure optimization [62,63] and other engineering problems.

Nomenclature

Variables

B	Strain–displacement matrix
$B_{i,p}(\xi)$	i th B-spline basis function of p -degree
C^{p-k}	Continuity between knot spans
d_1, d_2	SMDs of the two vertices of an edge
D	Stress–strain matrix
Dis	Distance from a point to a point on a triangle
E	Young's modulus
F	Force vector
J	Jacobian matrix
J_1, J_2	Transformation relation from the NURBS parametric space to the physical space and the integration parametric space to the NURBS parametric space, respectively
k	Multiplicity of the knots
K_e	Element stiffness matrix
m	Number of polygons
n	Number of basis functions
N	Shape function vector
N_i	Shape function

$N_{i,p}(\xi)$	i th NURBS basis function of p -degree
p	Order of the B-spline
v_1	Vector from the vertex to the polygon plane
v_2	Outward vector of the polygon
$V_{\Omega_e}, V_{\hat{\Omega}_e},$ and $V_{\bar{\Omega}_e}$	Volumes of the element in physical, NURBS parametric, and integration domains, respectively
w_i	Positive weight
$w_{i,j}$	Weight value corresponding to the tensor product $B_{i,p}(\xi)B_{j,q}(\eta)$
x	Physical value (e.g., coordinate, displacement, and force)
\mathbf{x}	Physical value vector
ν	Poisson's ratio
τ	Traction of the loading region Ω
ξ, η	Variables for the coordinates in parameter space
Ξ	Knot vector
ϕ	Level set function value
ϕ_i	LSF value of the i th vertex of the regular embedded element
$\Omega_e, \hat{\Omega}_e,$ and $\bar{\Omega}_e$	Physical, NURBS parametric, and integration domains of the element, respectively

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
B-rep	Boundary representation
CAD	Computer-aided design
CAE	Computer-aided engineering
CSG	Constructive solid geometry
FCM	Finite cell method
FEA	Finite element analysis
FSI	Fluid–structure interaction
IGA	Isogeometric analysis
LSF	Level set function
MC	Marching cube
MD	Minimum distance
NURBS	Non-uniform rational B-spline
SMD	Sign minimum distance
STEP	Standard for the exchange of product model data

Acknowledgements This work has been supported by the National Key R&D Program of China (Grant No. 2020YFB1708300), the National Natural Science Foundation of China (Grant Nos. 52075184 and 51705158), and the Research Grants Council via Early Career Scheme, Hong Kong, China (RGC Ref. No. 27209817). These supports are gratefully acknowledged.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution, and reproduction in any medium or format as long as appropriate credit is given to the original author(s) and source, a link to

the Creative Commons license is provided, and the changes made are indicated.

The images or other third-party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Hughes T J R, Cottrell J A, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 2005, 194(39–41): 4135–4195
- Cottrell J A, Reali A, Bazilevs Y, et al. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 2006, 195(41–43): 5257–5296
- Weeger O, Wever U, Simeon B. Isogeometric analysis of nonlinear Euler–Bernoulli beam vibrations. *Nonlinear Dynamics*, 2013, 72(4): 813–835
- Benson D J, Bazilevs Y, Hsu M C, et al. A large deformation, rotation-free, isogeometric shell. *Computer Methods in Applied Mechanics and Engineering*, 2011, 200(13–16): 1367–1378
- Deng X, Korobenko A, Yan J, et al. Isogeometric analysis of continuum damage in rotation-free composite shells. *Computer Methods in Applied Mechanics and Engineering*, 2015, 284: 349–372
- Bazilevs Y, Deng X, Korobenko A, et al. Isogeometric fatigue damage prediction in large-scale composite structures driven by dynamic sensor data. *Journal of Applied Mechanics*, 2015, 82(9): 091008
- Bazilevs Y, Korobenko A, Deng X, et al. Fluid-structure interaction modeling for fatigue-damage prediction in full-scale wind-turbine blades. *Journal of Applied Mechanics*, 2016, 83(6): 061010
- Peng X, Atroshchenko E, Kerfriden P, et al. Isogeometric boundary element methods for three dimensional static fracture and fatigue crack growth. *Computer Methods in Applied Mechanics and Engineering*, 2017, 316: 151–185
- Bazilevs Y, Calo V M, Hughes T J R, et al. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 2008, 43(1): 3–37
- Hsu M C, Akkerman I, Bazilevs Y. High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Computers & Fluids*, 2011, 49(1): 93–100
- Yan J, Deng X, Korobenko A, et al. Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines. *Computers & Fluids*, 2017, 158: 157–166
- Bazilevs Y, Yan J, Deng X, et al. Computer modeling of wind turbines: 2. Free-surface FSI and fatigue-damage. *Archives of Computational Methods in Engineering*, 2019, 26(4): 1101–1115
- Wang Y, Benson D J. Isogeometric analysis for parameterized LSM-based structural topology optimization. *Computational Mechanics*, 2016, 57(1): 19–35
- Wang Y, Xu H, Pasini D. Multiscale isogeometric topology optimization for lattice materials. *Computer Methods in Applied Mechanics and Engineering*, 2017, 316: 568–585
- Wang Z P, Poh L H, Dirrenberger J, et al. Isogeometric shape optimization of smoothed petal auxetic structures via computational periodic homogenization. *Computer Methods in Applied Mechanics and Engineering*, 2017, 323: 250–271
- Xie X, Wang S, Xu M, et al. A new isogeometric topology optimization using moving morphable components based on R-functions and collocation schemes. *Computer Methods in Applied Mechanics and Engineering*, 2018, 339: 61–90
- Deng X W, Wu N, Yang K, et al. Integrated design framework of next-generation 85-m wind turbine blade: modelling, aeroelasticity and optimization. *Composites Part B, Engineering*, 2019, 159: 53–61
- Wang Z P, Poh L H. Optimal form and size characterization of planar isotropic petal-shaped auxetics with tunable effective properties using IGA. *Composite Structures*, 2018, 201: 486–502
- Wang Z P, Poh L H, Zhu Y, et al. Systematic design of tetra-petals auxetic structures with stiffness constraint. *Materials & Design*, 2019, 170: 107669
- Babic B, Nesic N, Miljkovic Z. A review of automated feature recognition with rule-based pattern recognition. *Computers in Industry*, 2008, 59(4): 321–337
- Zhu X F, Hu P, Ma Z D, et al. A new surface parameterization method based on one-step inverse forming for isogeometric analysis-suited geometry. *International Journal of Advanced Manufacturing Technology*, 2013, 65(9–12): 1215–1227
- Jaxon N, Qian X. Isogeometric analysis on triangulations. *Computer-Aided Design*, 2014, 46: 45–57
- Martin T, Cohen E, Kirby R M. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Computer-Aided Geometric Design*, 2009, 26(6): 648–664
- Xu G, Mourrain B, Duvigneau R, et al. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. *Computer-Aided Design*, 2013, 45(2): 395–404
- Zuo B Q, Huang Z D, Wang Y W, et al. Isogeometric analysis for CSG models. *Computer Methods in Applied Mechanics and Engineering*, 2015, 285: 102–124
- Bazilevs Y, Calo V M, Cottrell J A, et al. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 2010, 199(5–8): 229–263
- Scott M A, Li X, Sederberg T W, et al. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 2012, 213–216: 206–222
- Whetten C, Sederberg M, Scott M. Isogeometric analysis using the *IGA_INCLUDE_BEZIER keyword in LS-DYNA. 2019
- Kim H J, Seo Y D, Youn S K. Isogeometric analysis for trimmed CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 2009, 198(37–40): 2982–2995
- Wang Y W, Huang Z D, Zheng Y, et al. Isogeometric analysis for compound B-spline surfaces. *Computer Methods in Applied Mechanics and Engineering*, 2013, 261–262: 1–15
- Nagy A P, Benson D J. On the numerical integration of trimmed

- isogeometric elements. *Computer Methods in Applied Mechanics and Engineering*, 2015, 284: 165–185
32. Wang Y, Benson D J, Nagy A P. A multi-patch nonsingular isogeometric boundary element method using trimmed elements. *Computational Mechanics*, 2015, 56(1): 173–191
 33. Ruess M, Schillinger D, Bazilevs Y, et al. 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*, 2013, 95(10): 811–846
 34. Marussig B, Zechner J, Beer G, et al. Stable isogeometric analysis of trimmed geometries. *Computer Methods in Applied Mechanics and Engineering*, 2017, 316: 497–521
 35. Guo Y, Heller J, Hughes T J R, et al. Variationally consistent isogeometric analysis of trimmed thin shells at finite deformations, based on the STEP exchange format. *Computer Methods in Applied Mechanics and Engineering*, 2018, 336: 39–79
 36. Marussig B, Hughes T J. A review of trimming in isogeometric analysis: challenges, data exchange and simulation aspects. *Archives of Computational Methods in Engineering*, 2018, 25(4): 1059–1127
 37. Parvizian J, Düster A, Rank E. Finite cell method. *Computational Mechanics*, 2007, 41(1): 121–133
 38. Düster A, Parvizian J, Yang Z, et al. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 2008, 197(45–48): 3768–3782
 39. Schillinger D, Ruess M. 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*, 2015, 22(3): 391–455
 40. Kamensky D, Hsu M C, Schillinger D, et al. An immersogeometric variational framework for fluid–structure interaction: application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 2015, 284: 1005–1053
 41. Xu F, Schillinger D, Kamensky D, et al. The tetrahedral finite cell method for fluids: immersogeometric analysis of turbulent flow around complex geometries. *Computers & Fluids*, 2016, 141: 135–154
 42. Huang J, Menq C H. Automatic CAD model reconstruction from multiple point clouds for reverse engineering. *Journal of Computing and Information Science in Engineering*, 2002, 2(3): 160–170
 43. Manmadhachary A, Kumar R, Krishnanand L. Improve the accuracy, surface smoothing and material adaption in STL file for RP medical models. *Journal of Manufacturing Processes*, 2016, 21: 46–55
 44. Wang J, Gu D, Yu Z, et al. A framework for 3D model reconstruction in reverse engineering. *Computers & Industrial Engineering*, 2012, 63(4): 1189–1200
 45. Rajon D A, Bolch W E. Marching cube algorithm: review and trilinear interpolation adaptation for image-based dosimetric models. *Computerized Medical Imaging and Graphics*, 2003, 27(5): 411–435
 46. Montani C, Scateni R, Scopigno R. A modified look-up table for implicit disambiguation of marching cubes. *The Visual Computer*, 1994, 10(6): 353–355
 47. Newman T S, Yi H. A survey of the marching cubes algorithm. *Computers & Graphics*, 2006, 30(5): 854–879
 48. Chang M, Oh J W, Chang D S, et al. Interactive marching cubes algorithm for intraoral scanners. *International Journal of Advanced Manufacturing Technology*, 2017, 89(5–8): 2053–2062
 49. Doi A, Koide A. An efficient method of triangulating equi-valued surfaces by using tetrahedral cells. *IEICE Transactions on Information and Systems*, 1991, 74(1): 214–224
 50. Treece G M, Prager R W, Gee A H. Regularised marching tetrahedra: Improved iso-surface extraction. *Computers & Graphics*, 1999, 23(4): 583–598
 51. Guo D, Li C, Wu L, et al. Improved marching tetrahedra algorithm based on hierarchical signed distance field and multi-scale depth map fusion for 3D reconstruction. *Journal of Visual Communication and Image Representation*, 2017, 48: 491–501
 52. Zander N, Bog T, Elhaddad M, et al. FCMLab: a finite cell research toolbox for MATLAB. *Advances in Engineering Software*, 2014, 74: 49–63
 53. Verhoosel C V, Van Zwieten G J, Van Rietbergen B, et al. Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone. *Computer Methods in Applied Mechanics and Engineering*, 2015, 284: 138–164
 54. Kudela L, Kollmannsberger S, Almac U, et al. Direct structural analysis of domains defined by point clouds. *Computer Methods in Applied Mechanics and Engineering*, 2020, 358: 112581
 55. Piegl L, Tiller W. *The NURBS Book*. New York: Springer, 1997
 56. de Boor C. On calculating with B-splines. *Journal of Approximation Theory*, 1972, 6(1): 50–62
 57. Xia Z, Wang Q, Wang Y, et al. A CAD/CAE incorporate software framework using a unified representation architecture. *Advances in Engineering Software*, 2015, 87: 68–85
 58. Requicha A A G, Voelcker H B. Boolean operations in solid modeling: boundary evaluation and merging algorithms. *Proceedings of the IEEE*, 1985, 73(1): 30–44
 59. Hales T C. The Jordan curve theorem, formally and informally. *American Mathematical Monthly*, 2007, 114(10): 882–894
 60. Wang Y, Benson D J. Geometrically constrained isogeometric parameterized level-set based topology optimization via trimmed elements. *Frontiers of Mechanical Engineering*, 2016, 11(4): 328–343
 61. Eberly D. Distance between point and triangle in 3D. *Geometric Tools*, 2008
 62. Wang Y, Liao Z, Ye M, et al. An efficient isogeometric topology optimization using multilevel mesh, MGCG and local-update strategy. *Advances in Engineering Software*, 2020, 139: 102733
 63. Zhu B, Zhang X, Zhang H, et al. Design of compliant mechanisms using continuum topology optimization: a review. *Mechanism and Machine Theory*, 2020, 143: 103622